Performance Comparison of Embeddings and Keyword Selection Methods in Enterprise Document

Putri Cristin [1]*, Brenda Natalia [2]*, Joseph Clio Limantara[3*], Sarwosri[4*#]
* Teknik Informatika, Institut Teknologi Sepuluh Nopember
putri.cristin@gmail.com [1], brendanatalia1299@gmail.com [2], josephlimantara17@gmail.com [3], sarwosri@its.ac.id [4#]
The symbol # indicates the corresponding authors.

Article Info

Article history:

Received 2025-06-26
Revised 2025-07-26
Accepted 2025-07-31

Keyword:

KeyBERT ,
Embedding Models,
Enterprise Documents,
Keyword Extraction,
Keyword Selection Method.

ABSTRACT

Keyword extraction is widely used in domains such as social media and e-commerce, but its application for enterprise document retrieval remains limited. Most organizations still depend on structured systems or rule-based approaches for indexing, which often lack semantic understanding and scalability. While several techniques like TextRank and RAKE have been explored, few studies assess their effectiveness on operational document retrieval in institutional settings, revealing a research gap. This study investigates the use of KeyBERT to extract keywords from university documents, including SOPs, manuals, and guidelines. KeyBERT leverages transformer-based embeddings to generate semantically relevant keywords and is chosen for its ease of use, model flexibility, and no need for labeled data. Additionally, it supports diversification strategies such as Maximum Marginal Relevance (MMR) and MaxSum to reduce redundancy and enhance keyword variety. We evaluate six embedding models combined with three keyword selection methods: Cosine similarity, MMR, and MaxSum. The best F1 score of 0.78 is achieved using Cosine with the *paraphrase-MiniLM-L3-v2* model, along with an average extraction time of 184.02 seconds. These findings highlight the effectiveness of combining lightweight embeddings with strategic keyword selection for enterprise-scale document indexing.

I. Introduction

In recent years, keyword extraction has gained significant attention as a crucial task in information retrieval (IR), particularly for applications in social media, e-commerce, and academic research. The ability to automatically extract relevant keywords from textual content enables enhanced search functionalities and supports content organization. Within the realm of document information retrieval in corporate settings, keyword extraction holds a promising potential to revolutionize the way organizations manage and access their operational documents, such as Standard Operating Procedures (SOPs), guidelines, policies, and forms.

KeyBERT has emerged as one of the most widely adopted tools for keyword extraction, largely due to its compatibility with a range of transformer-based embedding models such as BERT, RoBERTa, and DistilBERT. A comparative study by Lestari et al. [1] examined the influence of different embedding models on the effectiveness of KeyBERT, while Nadim et al. [2] evaluated various unsupervised keyword extraction tools and found that KeyBERT, when combined with post-processing techniques like Maximum Marginal Relevance (MMR) and MaxSum, consistently outperformed other approaches across multiple scenarios.

Although previous research has compared keyword extraction methods, KeyBERT stands out not only for its robust performance but also for its extensibility. Its seamless integration with diverse embedding models and its support for tunable diversification strategies such as MMR and MaxSum make it a highly adaptable tool with significant potential for further enhancement.

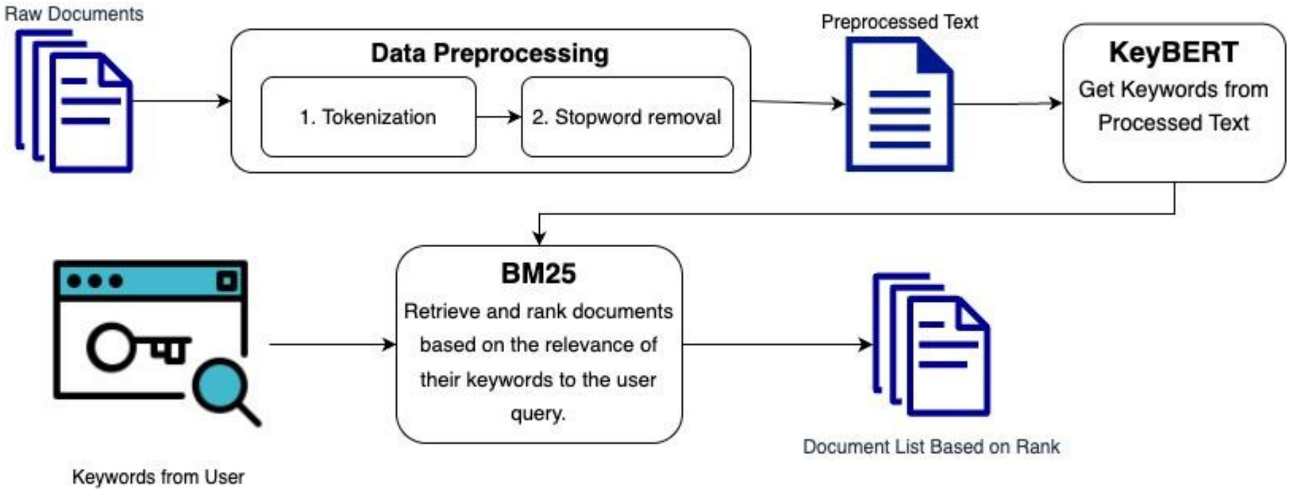

Raw Documents
Data Preprocessing
1. Tokenization
2. Stopword removal
Preprocessed Text
KeyBERT
Get Keywords from Processed Text
BM25
Retrieve and rank documents based on the relevance of their keywords to the user query.
Keywords from User
Document List Based on Rank

Figure 1. Methodology

Building upon these insights, the present study investigates the combined impact of embedding model selection and keyword diversification techniques—specifically MMR and MaxSum—on enhancing keyword quality for document retrieval tasks.

Currently, some organizations have adopted information system or rule based system to facilitate search retrieval and labelling [3]. Yahya et al. [4] presented a review of keyword extraction methods such as TextRank, underscoring the growing diversity in algorithmic approaches.

These traditional and semi-automated [5] approaches still face inherent limitations in scalability and efficiency—particularly as the volume of unstructured documents continues to grow rapidly. Similar concerns around manual processes and system scalability have been highlighted in other technical domains as well [5], [6]. Automated keyword extraction offers an opportunity to reduce reliance on manual processes while improving both retrieval speed and relevance.

This study focuses on applying and evaluating KeyBERT-based keyword extraction to improve document retrieval in enterprise environments, where research on such applications is still limited. Specifically, this study aims to compare the performance of different embedding models on enterprise document datasets and to maximize the use of keyword selection methods such as Maximum Marginal Relevance (MMR) and MaxSum. Through this approach, we aim to identify the most effective combination of embedding models and keyword selection techniques for optimizing keyword-based document indexing.

The dataset used in this study consists of real-world operational documents from a university setting, including SOPs, manuals, policies, and guidelines. Its diversity makes it a suitable testbed for evaluating the practical utility of automated keyword extraction methods in organizational environments.

This paper is organized into three main sections. The Methodology section outlines the dataset detail, KeyBERT steps, embedding models, keyword selection, and evaluation setup. The Results and Discussion section presents the experimental findings and key insights. Finally, the Conclusion summarizes the study and proposes directions for future research.

II. Metodology

In this study, we propose a methodology for evaluating keyword extraction methods using KeyBERT, combined with various embedding models and keyword selection strategies, in the context of document retrieval. The steps outlined in Figure *1* describe the data preprocessing, keyword extraction, document retrieval, and evaluation process.

A. *Dataset*

The dataset used in this research consists of 1,000 PDF documents, with a total size of approximately 1.49 GB. These documents represent a wide range of enterprise operational materials, including Standard Operating Procedures (SOPs), guidelines, work instructions, policies, and other administrative resources. The documents vary significantly in length, with character counts ranging from 673 to 1,069,680 characters, and an average length of approximately 143,934 characters per document. This high variation reflects the diversity and complexity of the content typically found in institutional documents.

The documents were collected from multiple internal sources within the university, namely:

- administrative office archives
- student services unit
- several operational departments, such as Human Capital, Quality Assurance, Marketing, Academic Affairs, and other related units involved in institutional governance and service delivery
- a compilation of documents from various faculties and departments

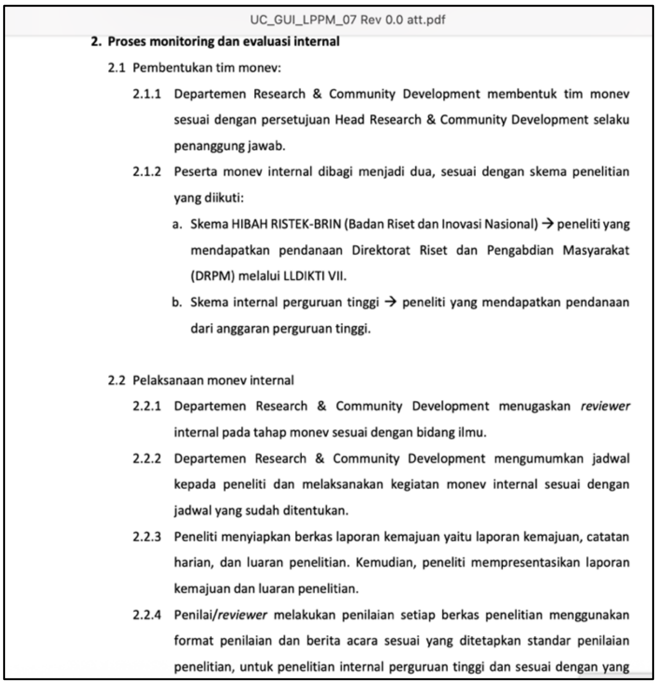
Each document varies in length, with page counts ranging from 2 to 184 pages, reflecting the diversity in scope and complexity of the content.

The data collection was carried out in collaboration with the university to ensure its legality and relevance to the research context. The data collection process involved gathering documents published within the period from 2022 to 2025 to ensure that the data used is up-to-date and relevant to the current context. This period selection aims to accurately represent the dynamics of information in the academic environment.

The features extracted from the dataset are <*File Name*>, <*Document No.*>, <*Document Title*>, and <*Document Content*> which detailed in TABLE I. The example of dataset is displayed in Figure 2. All text from document was extracted from real PDF Files with Python using PyPDF2 libraries.

TABLE I
DATASET FEATURE

Feature	Data Type
File Name	String
Document No.	String
Document Title	String
Document Content	String



UC_GUI_LPPM_07 Rev 0.0 att.pdf

2. Proses monitoring dan evaluasi internal

2.1 Pembentukan tim monev:

2.1.1 Departemen Research & Community Development membentuk tim monev sesuai dengan persetujuan Head Research & Community Development selaku penanggung jawab.

2.1.2 Peserta monev internal dibagi menjadi dua, sesuai dengan skema penelitian yang diikuti:

a. Skema HIBAH RISTEK-BRIN (Badan Riset dan Inovasi Nasional) → peneliti yang mendapatkan pendanaan Direktorat Riset dan Pengabdian Masyarakat (DRPM) melalui LLDIKTI VII.

b. Skema internal perguruan tinggi → peneliti yang mendapatkan pendanaan dari anggaran perguruan tinggi.

2.2 Pelaksanaan monev internal

2.2.1 Departemen Research & Community Development menugaskan *reviewer* internal pada tahap monev sesuai dengan bidang ilmu.

2.2.2 Departemen Research & Community Development mengumumkan jadwal kepada peneliti dan melaksanakan kegiatan monev internal sesuai dengan jadwal yang sudah ditentukan.

2.2.3 Peneliti menyiapkan berkas laporan kemajuan yaitu laporan kemajuan, catatan harian, dan luaran penelitian. Kemudian, peneliti mempresentasikan laporan kemajuan dan luaran penelitian.

2.2.4 Penilai/*reviewer* melakukan penilaian setiap berkas penelitian menggunakan format penilaian dan berita acara sesuai yang ditetapkan standar penilaian penelitian, untuk penelitian internal perguruan tinggi dan sesuai dengan yang

Figure 2. Dataset Preview

B. *Preprocessing*

Before applying the keyword extraction techniques, the raw text data undergoes several preprocessing steps to ensure the text is in an optimal format for analysis. The preprocessing includes the following:

1) Stopword Removal: Commonly used words such as "the," "and" and "is," which do not contribute significant meaning to the content, are removed from the text. This ensures that only meaningful words are considered during the keyword extraction process.
2) Cleaning: The documents from the operational company often contain various formatting elements, such as headers, footers, signatures, and names on the validation pages.

These preprocessing steps help to clean the text and reduce noise, making the keyword extraction process more accurate and efficient. Typically, tokenization and stemming are performed during the extraction process. However, in this Indonesian-language dataset, stemming can sometimes significantly alter the meaning of words. For example, the words "jabatan" (position), "pejabat" (official), and "jabat tangan" (handshake) have different meanings despite sharing the same root word "jabat." Considering this, we chose not to apply stemming to allow the embedding models to better understand the text semantically and extract keywords with the correct meaning.

C. *Keyword Extraction Using KeyBERT*

KeyBERT is a keyword extraction technique that leverages pre-trained BERT embeddings to generate semantically relevant keywords from a document [6]. The KeyBERT model enables users to extract keywords or key phrases from a given text by embedding sentences or documents into high-dimensional vector representations using BERT[7]. KeyBERT is widely adopted in various domains due to its ease of use and its ability to produce human-interpretable results without requiring large, annotated datasets. Its implementation is publicly available and actively maintained through the official GitHub repository, which provides extensive examples and supports various transformer models.

Importantly, KeyBERT accommodates a wide range of embedding models, including both BERT-based and more recent architectures such as BGE (BAAI General Embedding), offering flexibility in semantic representation. It also supports keyword diversification methods like Maximum Marginal Relevance (MMR) and MaxSum, allowing for greater variation and coverage in extracted keywords.

KeyBERT has been applied as a model for extracting keywords from both medical and non-medical service guidelines [7], [8]. Kelebercová et al.[9] used KeyBERT to analyze news related to COVID-19 in order to distinguish between fake and true news.

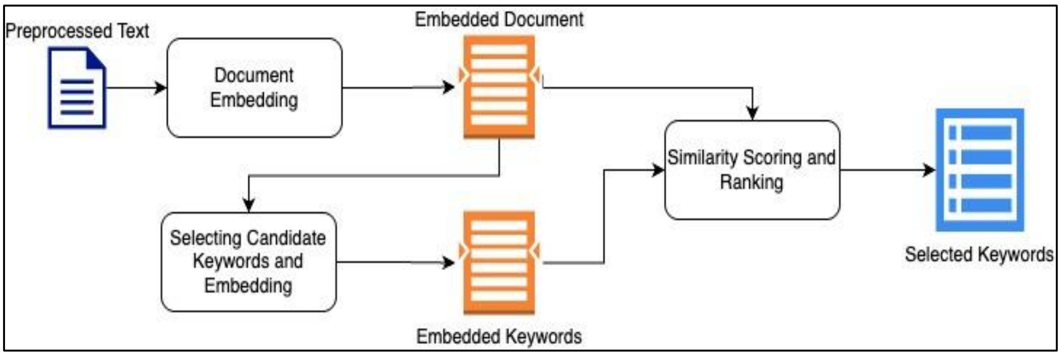


Figure 3. KeyBERT Steps

KeyBERT, as shown in Figure 3, has three steps: (1) document embedding, (2) candidate keywords selection and embedding, then (3) similarity scoring and ranking.

2) *Document Embedding:* The input document is first embedded into a dense vector using a pre-trained embedding model. This vector captures the semantic meaning of the entire document in a high-dimensional space. We evaluated six sentences embedding models, each differing in architecture, training objectives, and embedding capabilities.

Sentence embeddings are fixed-length dense vector representations of text that aim to capture the semantic meaning of sentences. They are fundamental in various natural language processing (NLP) applications such as semantic search, text clustering, and keyword extraction, particularly in embedding-based methods like KeyBERT framework.

The selected models vary in terms of architectural design, pretraining objectives, and embedding dimensionality, as summarized in TABLE II.

TABLE II
EMBEDDING MODEL FOR KEYBERT

Embedding Model Name	Architecture Model
all-MiniLM-L6-v2	MiniLM (based on BERT)
all-MiniLM-L12-v2	MiniLM (based on BERT)
paraphrase-MiniLM-L3-v2	MiniLM (based on BERT)
paraphrase-MiniLM-L6-v2	MiniLM (based on BERT)
msmarco-distilbert-base-v2	DistilBERT
BAAI/bge-base-en-v1.5	BGE

The *all-MiniLM-L6-v2* model is a lightweight, general-purpose sentence embedding model developed by the SentenceTransformers framework [10]. It comprises six transformer layers and approximately 22 million parameters. This model is trained using a contrastive learning objective on diverse sentence pairs, optimizing for semantic textual similarity tasks. Despite its relatively small size, it demonstrates strong performance across a variety of benchmarks and is widely adopted in scenarios where computational efficiency is critical. Its low latency makes it particularly suitable for real-time applications such as clustering or similarity-based keyword extraction.

The *all-MiniLM-L12-v2* is a deeper variant of the *MiniLM* series, consisting of 12 transformer layers. It extends the representational capacity of the L6 version by introducing a larger architecture (approximately 33 million parameters), enabling it to capture more complex semantic relationships. Like its predecessor, it is trained using a contrastive learning setup and is optimized for general-purpose sentence similarity tasks. This model is better suited for applications requiring more nuanced understanding, such as semantic clustering in large corpora or content recommendation systems, where embedding quality outweighs computational constraints.

The *paraphrase-MiniLM-L3-v2* model is an extremely compact sentence embedding model, containing only three transformer layers and roughly 14 million parameters. It is specifically fine-tuned on paraphrase detection tasks, which focus on identifying semantically equivalent or near-duplicate sentences. Although its embedding quality is relatively modest compared to larger models, its rapid inference speed makes it ideal for scenarios where processing time is a limiting factor, such as edge computing or large-scale, low-latency retrieval systems.

Building on the L3 variant, the *paraphrase-MiniLM-L6-v2* introduces a deeper architecture with six transformer layers. This increase in depth enhances the model's ability to generate more informative semantic representations, improving its performance on paraphrase identification and other sentence-level similarity tasks. It maintains a favourable balance between speed and accuracy, making it an effective option for real-time paraphrase mining or search ranking systems that require moderately high semantic resolution.

The *msmarco-distilbert-base-v2* model is a specialized embedding model trained for dense passage retrieval, particularly within question answering and information retrieval contexts. It is based on the *DistilBERT* architecture [11] and fine-tuned on the *MS MARCO* dataset [12], which consists of large-scale query-passage pairs. Unlike symmetric models used for general sentence similarity, this model is optimized asymmetrically—queries and passages are embedded using different objectives—to enhance relevance ranking in retrieval systems. As such, it is highly effective for search applications but less suited for clustering or general semantic similarity tasks.

The *BAAI/bge-base-en-v1.5* model is a recent state-of-the-art embedding model developed by the Beijing Academy of Artificial Intelligence. It belongs to the Bilingual Generation Embedding (BGE) family and is optimized specifically for English. With approximately 110 million parameters, it is significantly larger than the *MiniLM* and *DistilBERT*-based models. It supports both standard sentence embedding and instruction-tuned formats. BGE-base-v1.5 has demonstrated strong performance in multiple semantic retrieval benchmarks and is well-suited for tasks involving high-precision semantic understanding, such as zero-shot search or multi-document summarization.

By evaluating embedding models with diverse architectures, dimensional complexities, and training objectives, this study seeks to assess their capability in extracting semantically rich and contextually relevant keywords from textual documents.

3) *Candidate Keywords Selection and Embedding:* Candidate keywords are selected from the document, typically using simple n-gram extraction (e.g., unigrams, bigrams, trigrams). Each candidate is then individually embedded into a vector using the same embedding model used for the document.

4) *Similarity Scoring and Ranking:* The similarity between each candidate's embedding and the document embedding is computed, typically using cosine similarity as the default similarity measurement method. Candidates are then ranked based on their similarity scores, and the top-N most similar phrases are selected as the final keywords. This ensures that the extracted keywords are semantically close to the overall content of the document.

To further enhance the informativeness and diversity of extracted keywords, KeyBERT provides support for alternative selection strategies beyond its default cosine similarity-based ranking. These include Maximal Marginal Relevance (MMR) and MaxSum Similarity, which aim to address the issue of keyword redundancy by incorporating diversity-aware mechanisms during selection [13].

The default approach in KeyBERT ranks candidate keywords based solely on their cosine similarity to the document embedding [14]. In (1) given a document embedding $\vec{\mathrm{d}}$ and $\vec{\mathrm{k_i}}$ and a candidate keyword embedding the relevance score is calculated as:

$$\cos\left(\vec{\mathrm{d}},\vec{\mathrm{k_i}}\right)=\frac{\vec{\mathrm{d}}\cdot\vec{\mathrm{k_i}}}{\left|\vec{\mathrm{d}}\right|\cdot\left|\vec{\mathrm{k_i}}\right|} \tag{1}$$

While this method effectively identifies highly relevant terms, it does not explicitly account for redundancy among selected keywords, which may lead to overlapping or semantically similar terms dominating the result set.

Maximal Marginal Relevance (MMR) [15], [16] is a strategy designed to balance relevance and diversity in information retrieval. In the context of keyword extraction, MMR selects keywords that are not only similar to the document but also dissimilar to each other. The selection criterion for a keyword $\mathrm{k_i}$ at each step is defined as:

$$\mathrm{MMR}(\mathrm{k_i})=\lambda\cdot\cos\left(\vec{\mathrm{d}},\vec{\mathrm{k_i}}\right)-(1-\lambda)\cdot\max_{\mathrm{k_j}\in S}\cos\left(\vec{\mathrm{k_i}},\vec{\mathrm{k_j}}\right) \tag{2}$$

Where in (2) :

- $\lambda\in[0,1]$ is a tunable parameter controlling the trade-off between relevance and diversity.
- S is the set of already selected keywords.
- The first term promotes relevance to the document, while the second term penalizes similarity to already selected terms.

MMR is particularly useful in scenarios where the risk of redundancy is high, ensuring that the final set of keywords provides a more representative and diverse summary of the document's content.

MaxSum Similarity offers an alternative approach that focuses on global informativeness and diversity rather than sequential selection. Instead of iteratively selecting terms, MaxSum identifies a subset of N keywords from a pool of top M candidates that collectively maximize the pairwise distance (or minimize similarity) among them, while still being relevant to the document [17].

The algorithm involves two steps:

a) Relevance filtering: Select the top M candidates based on cosine similarity to the document.

b) Diversity maximization: From this subset, choose N candidates whose pairwise cosine similarities are minimized, i.e.,

$$MaxSum=argmin_{S\subseteq C,|S|=N}\sum_{(k_\mathrm{i},k_\mathrm{j})\in S,i\neq j}cos(k_\mathrm{i},k_\mathrm{j}) \tag{3}$$

Where in (3), C is the candidate pool of size M and S is the selected set of N keywords.

This method favors sets of keywords that are semantically distinct from one another, thus improving coverage of the document's content without relying on parameter tuning like MMR.

The primary distinction between these methods and the default cosine similarity ranking lies in the consideration of inter-keyword relationships. While cosine similarity ranks keywords solely by their relevance to the document, MMR and MaxSum explicitly reduce redundancy, making them more suitable for extractive tasks that require semantic diversity. In this research, we incorporate both MMR and MaxSum methods to improve the informativeness and coverage of the extracted keywords.

D. Document Retrieval with BM25

To simulate a real-world document retrieval system, we employ Best Matching 25 (BM25), one of the most successful text retrieval algorithms. BM25 is widely recognized for its effectiveness in various applications, including web search and document ranking[18]. Previous studies also explored hierarchical retrieval approaches to improve access to legal and institutional documents [19]. It is based on the probabilistic retrieval model, specifically the Okapi BM25 variant. The retrieval process consists of the following steps:

1) Keyword Input: The user enters a single keyword to search for relevant documents. In this study, we selected five keywords for evaluation: *pedoman (guideline), jabatan (position), mahasiswa (student), admission, and laboratorium*. The first three keywords are Indonesian terms, marked in italics to indicate their foreign origin. The last two keywords, admission and laboratorium, are retained in their original form because they are commonly used terms within the university context, even in English-language communications. These keywords were chosen because they are generally relevant to the content of the document dataset and reflect commonly searched topics in a university setting.

2) Document Ranking: BM25 ranks the documents by comparing the user's keyword against the document keywords generated by the KeyBERT process, calculating a relevance score for each document.

3) Result Display: Documents are presented to the user in order of their BM25 scores, with the most relevant appearing first.

In this study, BM25 was applied to a sample of 250 documents out of a total of 1,000 documents that had previously been processed using KeyBERT. This sample was verified by the university's Quality Assurance department, which is responsible for the official management and validation of institutional documents. The verification ensures that the sample represents high-quality and authoritative content suitable for evaluation.

Only 250 documents were included in the sample due to limited resources, which made it infeasible to manually verify the entire dataset. Therefore, a representative subset was selected to balance accuracy and feasibility within the constraints of the study.

E. Evaluation

The experiments were performed on a document dataset using six different pre-trained sentence embedding models. Each model was tested under three different keyword selection method: the default cosine similarity, Maximal Marginal Relevance (MMR), and MaxSum. The effectiveness of the keyword extraction and retrieval system is evaluated using several metrics:

1) *Execution Time of Keyword Extraction*: We measure the time taken by the keyword extraction process using KeyBERT and various embedding models. This is critical to assess the scalability and efficiency of the system, particularly in large-scale document retrieval scenarios.

2) *Ground Truth*: a collection of true labels or annotations that serves as the reference standard against which model predictions are evaluated. In this study, the ground truth was created with the assistance of the university's Quality Assurance department.

Due to resource constraints, the team selected a verified sample of 250 documents from the total corpus of 1,000 documents. These documents were carefully reviewed to ensure the reliability of the evaluation results.

The labeling process was based on five representative keywords that had been pre-selected in Section D.1. For each document in the sample, the Quality Assurance team manually determined which of the five keywords were relevant. As a result, a single document could be assigned to multiple labels, reflecting its association with more than one key topic.

This ground truth dataset provides a benchmark for evaluating the performance of the keyword extraction methods, particularly in terms of semantic relevance and coverage. The same set of labels is also used in the BM25-based evaluation described in Section D.

3) *Precision*: Measures the proportion of retrieved documents that are relevant.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{4}$$

Where in (4), *TP* is the number of relevant documents correctly retrieved, and *FP* is the number of non-relevant documents incorrectly retrieved.

4) *Recall*: Measures the proportion of relevant documents that were retrieved.

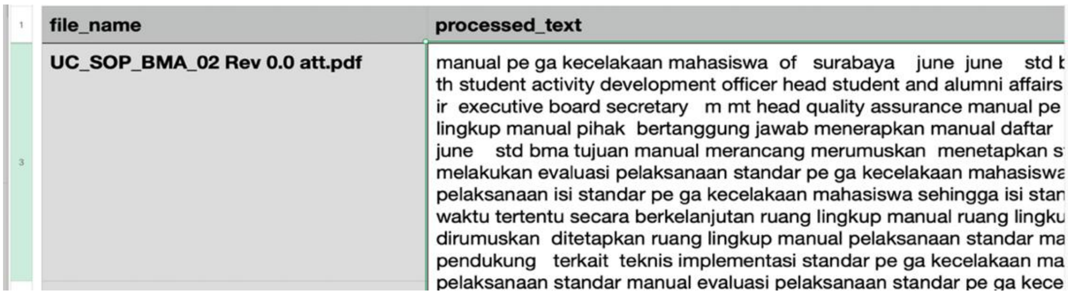
$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{5}$$

where in (5), *FN* is the number of relevant documents that were not retrieved

5) *F1 Score*: The harmonic mean of precision and recall used to balance both metrics into a single performance measure. The formula is defined as (6):

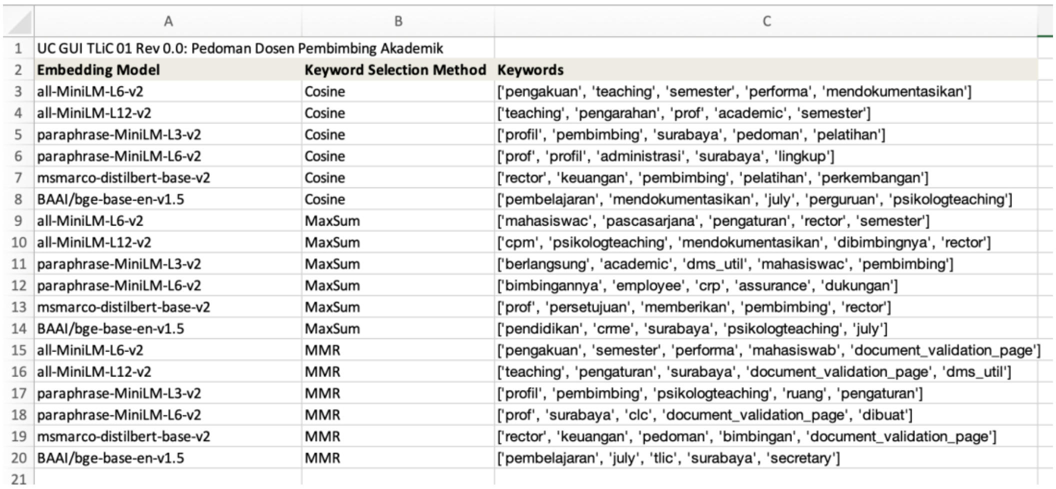
$$\text{F1 Score} = 2\ \text{x} \frac{\text{Precision x Recall}}{\text{Precision} + \text{Recall}} \tag{6}$$

In this study, the F1 Score is calculated using the micro-average approach, which aggregates the total true positives, false positives, and false negatives across all labels before computing precision, recall, and F1.



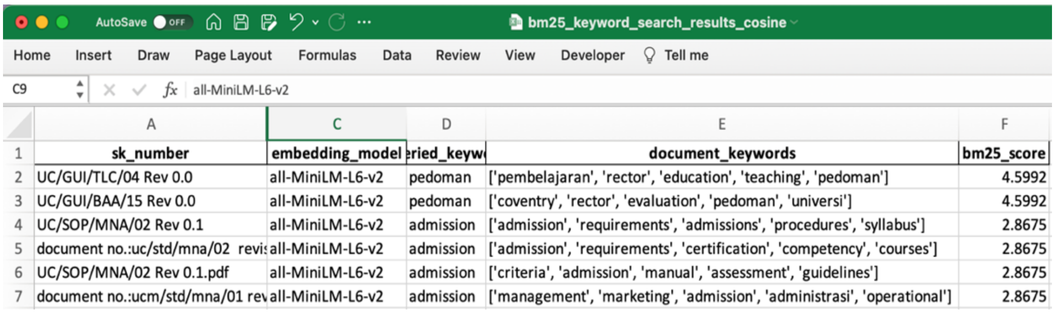
1	file_name	processed_text
3	UC_SOP_BMA_02 Rev 0.0 att.pdf	manual pe ga kecelakaan mahasiswa of surabaya june june std b th student activity development officer head student and alumni affairs ir executive board secretary m mt head quality assurance manual pe lingkup manual pihak bertanggung jawab menerapkan manual daftar june std bma tujuan manual merancang merumuskan menetapkan s melakukan evaluasi pelaksanaan standar pe ga kecelakaan mahasiswa pelaksanaan isi standar pe ga kecelakaan mahasiswa sehingga isi stan waktu tertentu secara berkelanjutan ruang lingkup manual ruang lingku dirumuskan ditetapkan ruang lingkup manual pelaksanaan standar ma pendukung terkait teknis implementasi standar pe ga kecelakaan ma pelaksanaan standar manual evaluasi pelaksanaan standar pe ga kece

Figure 4.Preprocessed Text



	A	B	C
1	UC GUI TLiC 01 Rev 0.0: Pedoman Dosen Pembimbing Akademik		
2	**Embedding Model**	**Keyword Selection Method**	**Keywords**
3	all-MiniLM-L6-v2	Cosine	['pengakuan', 'teaching', 'semester', 'performa', 'mendokumentasikan']
4	all-MiniLM-L12-v2	Cosine	['teaching', 'pengarahan', 'prof', 'academic', 'semester']
5	paraphrase-MiniLM-L3-v2	Cosine	['profil', 'pembimbing', 'surabaya', 'pedoman', 'pelatihan']
6	paraphrase-MiniLM-L6-v2	Cosine	['prof', 'profil', 'administrasi', 'surabaya', 'lingkup']
7	msmarco-distilbert-base-v2	Cosine	['rector', 'keuangan', 'pembimbing', 'pelatihan', 'perkembangan']
8	BAAI/bge-base-en-v1.5	Cosine	['pembelajaran', 'mendokumentasikan', 'july', 'perguruan', 'psikologteaching']
9	all-MiniLM-L6-v2	MaxSum	['mahasiswac', 'pascasarjana', 'pengaturan', 'rector', 'semester']
10	all-MiniLM-L12-v2	MaxSum	['cpm', 'psikologteaching', 'mendokumentasikan', 'dibimbingnya', 'rector']
11	paraphrase-MiniLM-L3-v2	MaxSum	['berlangsung', 'academic', 'dms_util', 'mahasiswac', 'pembimbing']
12	paraphrase-MiniLM-L6-v2	MaxSum	['bimbingannya', 'employee', 'crp', 'assurance', 'dukungan']
13	msmarco-distilbert-base-v2	MaxSum	['prof', 'persetujuan', 'memberikan', 'pembimbing', 'rector']
14	BAAI/bge-base-en-v1.5	MaxSum	['pendidikan', 'crme', 'surabaya', 'psikologteaching', 'july']
15	all-MiniLM-L6-v2	MMR	['pengakuan', 'semester', 'performa', 'mahasiswab', 'document_validation_page']
16	all-MiniLM-L12-v2	MMR	['teaching', 'pengaturan', 'surabaya', 'document_validation_page', 'dms_util']
17	paraphrase-MiniLM-L3-v2	MMR	['profil', 'pembimbing', 'psikologteaching', 'ruang', 'pengaturan']
18	paraphrase-MiniLM-L6-v2	MMR	['prof', 'surabaya', 'clc', 'document_validation_page', 'dibuat']
19	msmarco-distilbert-base-v2	MMR	['rector', 'keuangan', 'pedoman', 'bimbingan', 'document_validation_page']
20	BAAI/bge-base-en-v1.5	MMR	['pembelajaran', 'july', 'tlic', 'surabaya', 'secretary']
21			

Figure 5. Visualization Keyword Extracted from Document



AutoSave OFF
bm25_keyword_search_results_cosine
Home Insert Draw Page Layout Formulas Data Review View Developer Tell me
C9
fx

all-MiniLM-L6-v2

	A	C	D	E	F
1	sk_number	embedding_model	eried_keyw	document_keywords	bm25_score
2	UC/GUI/TLC/04 Rev 0.0	all-MiniLM-L6-v2	pedoman	['pembelajaran', 'rector', 'education', 'teaching', 'pedoman']	4.5992
3	UC/GUI/BAA/15 Rev 0.0	all-MiniLM-L6-v2	pedoman	['coventry', 'rector', 'evaluation', 'pedoman', 'universi']	4.5992
4	UC/SOP/MNA/02 Rev 0.1	all-MiniLM-L6-v2	admission	['admission', 'requirements', 'admissions', 'procedures', 'syllabus']	2.8675
5	document no.:uc/std/mna/02 revi	all-MiniLM-L6-v2	admission	['admission', 'requirements', 'certification', 'competency', 'courses']	2.8675
6	UC/SOP/MNA/02 Rev 0.1.pdf	all-MiniLM-L6-v2	admission	['criteria', 'admission', 'manual', 'assessment', 'guidelines']	2.8675
7	document no.:ucm/std/mna/01 rev	all-MiniLM-L6-v2	admission	['management', 'marketing', 'admission', 'administrasi', 'operational']	2.8675

Figure 6. Result from BM25

This approach is particularly suitable for multi-label classification tasks with imbalanced label distribution, as it reflects the overall performance across the entire dataset rather than treating each label equally. By using these evaluation metrics along with the ground truth dataset, we are able to assess the overall performance of the keyword extraction tool, the effectiveness of different embedding models, and the impact of MMR and MaxSum methods in improving keyword extraction quality.

III. Result and Discussion

A. *Experimental Setup*

The experiments were conducted on the platform with following system specifications: Python version: 3.11.12, CPU architecture: x86_64, RAM: 12.67 GB, GPU: NVIDIA Tesla T4 with 14.74 GB of VRAM. The following Python libraries and versions were used throughout the experiments: PyTorch (v2.6.0+cu124), Transformers (v4.51.3), Sentence-Transformers (v3.4.1), KeyBERT (v0.9.0)

B. *Result and Discussion*

In this study, the experimental pipeline comprises three core stages: data preprocessing, semantic embedding, and document retrieval via BM25 ranking. Each of these stages contributes sequentially to the production of interpretable results and supports the overall goal of extracting meaningful keywords and retrieving relevant documents.

First, during the data preprocessing stage, the input dataset undergoes tokenization and stopword resulting in raw text documents Figure 4. This step is crucial to ensure consistency and remove irrelevant noise prior to embedding.

Next, the embedding stage employs sentence-level transformer models to generate dense vector representations for each processed document. From these embeddings, a list of top-ranked keywords is extracted using similarity-based ranking methods such as cosine similarity, Maximal Marginal Relevance (MMR), or MaxSum Similarity. These keywords asshown in Figure 5, reflect the core semantic content of each document and serve as the primary output of the keyword extraction task.

We conducted a small-scale limited qualitative review by comparing extracted keywords from multiple embedding models and selection methods with the content of a selected ground truth document in Figure 5, "*UC GUI TLiC 01 Rev 0.0: Pedoman Dosen Pembimbing Akademik*". Overall, the extracted keywords were topically aligned with the document's content, including terms like *pembimbing, pedoman, mahasiswa, and administrasi*. However, some outputs revealed less relevant or noisy terms, such as *psikologteaching, dms_util, and document_validation_page*. These artifacts appear to be due to tokenization issues, or uncommon compound words not filtered during preprocessing. Despite that, we observe there is not repetitive keywords within the same output set. Selection methods like MaxSum and MMR contributed positively to reducing redundancy, although they occasionally included less semantically meaningful terms.

Finally, in the retrieval stage, the BM25 algorithm is applied to rank documents based on their lexical relevance to a given query or keyword set. The outcome of this phase is a ranked list of documents, where higher-ranking entries are considered more relevant according to weighting schemes inherent to BM25. The result can be seen in Figure 6.

This sequential flow—from raw text to semantic representation and finally to ranked document retrieval—enables a structured evaluation of both keyword quality and retrieval performance.

We evaluate this experiment measuring execution time, precision, recall, and f1-score are calculated to evaluate performance of KeyBERT.

1) *Execution Time*: Total execution time (in seconds) presented in *TABLE III.* Among all models, *paraphrase-MiniLM-L3-v2* demonstrated the fastest execution time across all methods, with **111.29** seconds for cosine similarity, **146.83** seconds for MMR, and **271.76** seconds for MaxSum. In contrast, *BAAI/bge-base-en-v1.5* exhibited the longest processing time, particularly under the MaxSum methods (452.62 seconds), which is consistent with its larger model size and higher computational demands.

We can observe that the default cosine similarity method yields the shortest execution times with average **184.02** seconds for all models. Introducing MMR increases execution time moderately, while MaxSum consistently results in the highest execution times. This is expected, as both MMR and MaxSum involve additional computations for optimizing keyword diversity and coverage. This is not fully in line with experiment conducted by Nadim et al. [2], which KeyBERT with Maxsum resulted with longest execution time 7.2472 seconds for SemEval2017 dataset. While KeyBERT with cosine take 0.9553 seconds and KeyBERT with MMR take 0.9541seconds in the same dataset.

Moreover, comparing across models, the all-MiniLM series (L6-v2 and L12-v2) showed a balanced trade-off between performance and efficiency. While all-MiniLM-L12-v2 performed slower than L6-v2, it remained faster than larger models like BAAI/bge-base-en-v1.5.

These findings suggest that lighter models, particularly MiniLM-based variants, are more suitable for large-scale or resource-constrained environments, especially when combined with simpler keyword selection method like cosine similarity.

TABLE III
EXECUTION TIME USING DEFAULT COSINE SIMILARITY

Embedding Model	Execution time (s)		
	Cosine Similarity	**MMR**	**MaxSum**
all-MiniLM-L6-v2	**155.88 s**	197.32 s	324.06 s
all-MiniLM-L12-v2	**233.93 s**	265.70 s	387.08 s
paraphrase-MiniLM-L3-v2	**111.29 s**	146.83 s	271.76 s
paraphrase-MiniLM-L6-v2	**146.51 s**	189.52 s	315.11 s
msmarco-distilbert-base-v2	**165.34 s**	196.92 s	331.25 s
BAAI/bge-base-en-v1.5	**291.14 s**	317.62 s	452.62 s
Average Execution Time	**184.02 s**	218.99 s	339.45 s

2) *Precision, Recall, and F1 Score*: Result of precision, recall and F1 Score from combination embedding models and keyword selection method presented in Table IV. The experimental results demonstrate that keyword extraction performance varies significantly depending on the combination of embedding models and keyword selection method.

The performance comparison across different sentence embedding models and retrieval strategies (Cosine, MMR, MaxSum) reveals distinct trends in precision, recall, and F1-score.

TABLE IV
EVALUATION RESULT EACH EMBEDDING MODEL AND KEYWORD SELECTION METHOD

Keyword Selection Method	Embedding Model	Precision	Recall	F1 Score
Cosine	BAAI/bge-base-en-v1.5	0.78	0.60	0.68
	all-MiniLM-L12-v2	0.70	0.70	0.70
	all-MiniLM-L6-v2	0.55	0.55	0.55
	msmarco-distilbert-base-v2	0.86	0.68	0.76
	paraphrase-MiniLM-L3-v2	**0.79**	**0.76**	**0.78**
	paraphrase-MiniLM-L6-v2	0.71	0.65	0.68
MMR	BAAI/bge-base-en-v1.5	0.83	**0.70**	**0.76**
	all-MiniLM-L12-v2	0.54	0.54	0.54
	all-MiniLM-L6-v2	0.63	0.56	0.59
	msmarco-distilbert-base-v2	**0.87**	0.65	0.74
	paraphrase-MiniLM-L3-v2	0.72	0.62	0.67
	paraphrase-MiniLM-L6-v2	0.75	0.66	0.70
Maxsum	BAAI/bge-base-en-v1.5	**0.90**	0.68	**0.77**
	all-MiniLM-L12-v2	0.64	**0.70**	0.67
	all-MiniLM-L6-v2	0.70	0.65	0.67
	msmarco-distilbert-base-v2	0.79	0.66	0.72
	paraphrase-MiniLM-L3-v2	0.78	0.65	0.70
	paraphrase-MiniLM-L6-v2	0.81	0.68	0.74

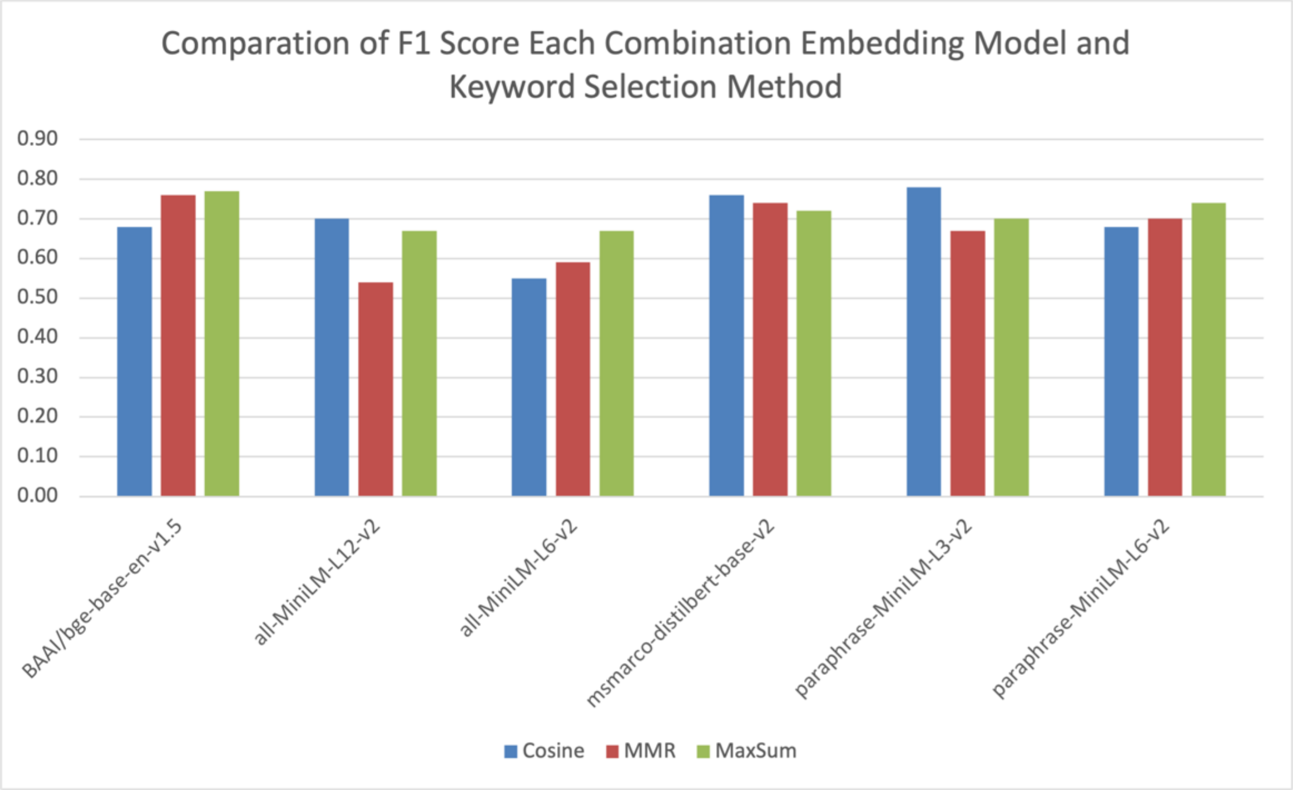


Figure 7. Comparation of F1 Score Each Combination Embedding Model and Keyword Selection Method

TABLE V.
TRADE OFF F1 SCORE AND EXECUTION TIME

Method	Embedding Model	Execution Time (s)	F1 Score
Cosine	all-MiniLM-L12-v2	233.93	0.70
Cosine	all-MiniLM-L6-v2	155.88	0.55
Cosine	BAAI/bge-base-en-v1.5	291.14	0.68
Cosine	msmarco-distilbert-base-v2	165.34	0.76
Cosine	paraphrase-MiniLM-L3-v2	111.29	0.78
Cosine	paraphrase-MiniLM-L6-v2	146.51	0.68
MaxSum	all-MiniLM-L12-v2	387.08	0.67
MaxSum	all-MiniLM-L6-v2	324.06	0.67
MaxSum	BAAI/bge-base-en-v1.5	452.62	0.77
MaxSum	msmarco-distilbert-base-v2	331.25	0.72
MaxSum	paraphrase-MiniLM-L3-v2	271.76	0.70
MaxSum	paraphrase-MiniLM-L6-v2	315.11	0.74
MMR	all-MiniLM-L12-v2	265.70	0.54
MMR	all-MiniLM-L6-v2	197.32	0.59
MMR	BAAI/bge-base-en-v1.5	317.62	0.76
MMR	msmarco-distilbert-base-v2	196.92	0.74
MMR	paraphrase-MiniLM-L3-v2	146.83	0.67
MMR	paraphrase-MiniLM-L6-v2	189.52	0.70

Among all configurations, the combination of MaxSum with *BAAI/bge-base-en-v1.5* yields the highest precision (**0.90**). This suggests its strong capability to identify the most relevant content, although the recall remains moderate at **0.68**, indicating some trade-off in coverage.

In terms of overall F1 score, which reflects a balanced performance between precision and recall, the best-performing setup is Cosine with paraphrase-MiniLM-L3-v2, achieving an F1 score of **0.78**. It is closely followed by MaxSum with *BAAI/bge-base-en-v1.5* (**0.77**) and MMR with the same model (**0.76**), underscoring the consistent strength of the BAAI model across methods.

The *paraphrase-MiniLM-L3-v2* model performs well likely due to its fine-tuning for semantic similarity and paraphrase detection, using datasets such as STS and SNLI. This allows it to capture nuanced meanings between text pairs, aligning particularly well with the Cosine similarity method, which relies on tight semantic distance in vector space.

In addition, *paraphrase-MiniLM-L3-v2* benefits from a compact and efficient architecture, making it less prone to overfitting and more generalizable across varied inputs. Unlike larger models, it creates dense but focused embeddings, enabling it to maintain high precision (**0.79**) while also achieving strong recall (**0.76**), which contributes to its top F1 score among all configurations.

This result is also not fully in line with Nadim et al.[2], where the experiment resulted best F1 Score in MMR and followed by Maxsum and Cosine in Semval2017 dataset. From our analysis in can be caused by different complexity and structure of our dataset compared by Semval2017.

TABLE VI
F1 SCORE GROUPED BY KEYWORD SELECTION METHOD

Keyword Selection Method	Precision	Recall	F1
Cosine	0.73	**0.77**	**0.75**
MMR	0.73	0.73	0.73
Maxsum	**0.76**	0.75	**0.75**

Then, in the evaluation by keyword selection method in Table VI, Cosine and MaxSum both achieve the highest overall F1 Score (**0.75**), indicating similarly strong performance in balancing precision and recall. Cosine yields the highest recall (**0.77**), suggesting it captures more true relevant keywords, albeit with a slight compromise in precision. Meanwhile, MMR presents a perfectly balanced precision and recall (both **0.73**), but its overall F1 Score remains marginally lower.

However, since the average F1 scores across the three methods are relatively close, this suggests that the dominant factor influencing retrieval performance is not the keyword selection method itself, but rather how well each embedding model aligns with a given method. In other words, the synergy between model architecture and retrieval strategy plays a more critical role than the method in isolation.

To further support the performance analysis, we evaluate the trade-off between F1 Score and execution time of each model-method combination. This analysis provides practical insights, especially for real-time or resource-constrained applications where execution efficiency is a priority alongside retrieval accuracy. Table V below summarizes the F1 scores and corresponding execution times (in seconds) for all evaluated configurations.

Upon analyzing the relationship between execution time and F1 score, no strong or consistent linear correlation emerges across all model-method configurations. In theory, one might expect higher-performing models to require more computational time; however, the data shows that execution time does not always predict performance.

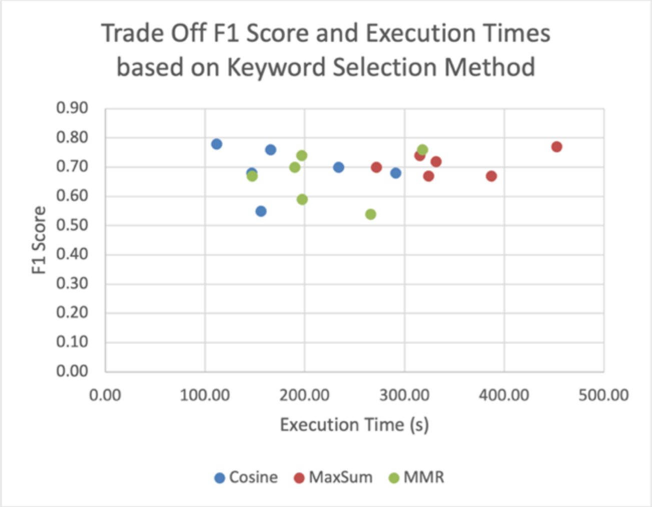


Figure 8. Trade Off F1 Score and Execution Times based on Keyword Selection Method

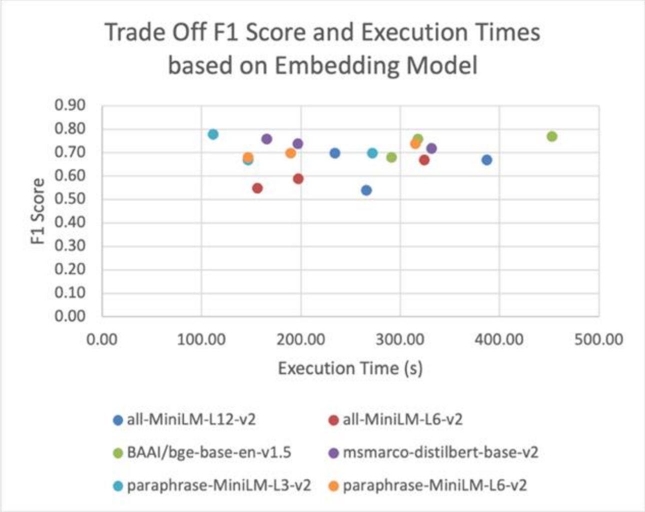


Figure 9. Trade Off F1 Score and Execution Times based on Embedding Model

For example in Figure 8 and Figure 9, *paraphrase-MiniLM-L3-v2* with Cosine achieves the highest F1 score (0.78) with the lowest execution time (111.29s), outperforming more computationally expensive setups like *BAAI/bge-base-en-v1.5* with MaxSum (F1: 0.77, Time: 452.62s). Similarly, some configurations with high execution time, such as MaxSum with *all-MiniLM-L12-v2*, do not yield competitive F1 scores. These findings indicate that efficiency and effectiveness are not necessarily proportional, and selecting the optimal setup requires evaluating both dimensions independently.

IV. CONCLUSION

This study presents a novel contribution by applying and evaluating the performance of KeyBERT for enterprise document retrieval—a domain where automated keyword extraction is still underutilized. The research focuses not only on using various sentence embedding models but also on comparing multiple keyword selection strategies (Cosine, MMR, MaxSum). By combining these two axes of experimentation, this work provides a comprehensive understanding of how embedding model characteristics interact with selection strategies, offering new insights into the optimization of document indexing in operational environments.

The experimental results indicate that the combination of *paraphrase-MiniLM-L3-v2* with the Cosine similarity method achieves the best F1 score (**0.78**), highlighting its suitability for lightweight, efficient retrieval systems. On the other hand, *BAAI/bge-base-en-v1.5* with MaxSum produces the highest precision, making it ideal for use cases requiring high retrieval accuracy. Across all models and methods, Cosine and MaxSum yield the highest average F1 scores (**0.75**), reflecting a balanced trade-off between relevance and diversity in keyword extraction. These findings suggest that the synergy between model architecture and selection technique plays a more significant role than the method alone.

For future work, several directions can be explored to further enhance system performance. First, implementing document chunking during preprocessing may improve the semantic precision of extracted keywords, especially for long documents. Additionally, integrating this keyword extraction framework into enterprise-level document management systems could improve real-world usability by offering seamless keyword-based search and indexing functionalities. Such developments could help institutional users better navigate and retrieve complex document collections efficiently. The findings are also aligned with recent developments such as ColPali, which emphasizes efficient document retrieval through customized keyword strategies [20].

REFERENCES

[1] B. Issa, M. B. Jasser, H. N. Chua, and M. Hamzah, "A Comparative Study on Embedding Models for Keyword Extraction Using KeyBERT Method," in *2023 IEEE 13th International Conference on System Engineering and Technology (ICSET)*, Shah Alam, Malaysia: IEEE, Oct. 2023, pp. 40–45. doi: 10.1109/ICSET59111.2023.10295108.

[2] M. Nadim, D. Akopian, and A. Matamoros, "A Comparative Assessment of Unsupervised Keyword Extraction Tools," *IEEE Access*, vol. 11, pp. 144778–144798, 2023, doi: 10.1109/ACCESS.2023.3344032.

[3] Y. Bi, T. Anderson, and S. McClean, "Rule Generation Based on Rough Set Theory for Text classification," in *Research and Development in Intelligent Systems XVII*, M. Bramer, A. Preece, and F. Coenen, Eds., London: Springer London, 2001, pp. 157–170. doi: 10.1007/978-1-4471-0269-4_12.

[4] M. Yahya, D. Eleyan, and A. Eleyan, "A Systematic Literature Review Of Automatic Keyword Extraction Algorithms: Textrank And," . *Vol.*, no. 20, 2021.

[5] R. Keeling *et al.*, "Empirical Comparisons of CNN with Other Learning Algorithms for Text Classification in Legal Document Review," in *2019 IEEE International Conference on Big Data (Big Data)*, Los Angeles, CA, USA: IEEE, Dec. 2019, pp. 2038–2042. doi: 10.1109/BigData47090.2019.9006248.

[6] M. Grootendorst, *MaartenGr/KeyBERT: BibTeX*. (Jan. 25, 2021). Zenodo. doi: 10.5281/ZENODO.4461265.

[7] Z. H. Amur, Y. K. Hooi, G. M. Soomro, H. Bhanbhro, S. Karyem, and N. Sohu, "Unlocking the Potential of Keyword Extraction: The Need for Access to High-Quality Datasets," *Appl. Sci.*, vol. 13, no. 12, p. 7228, Jun. 2023, doi: 10.3390/app13127228.

[8] J.-S. Lee and J. Hsiang, "Patent classification by fine-tuning BERT language model," *World Pat. Inf.*, vol. 61, p. 101965, Jun. 2020, doi: 10.1016/j.wpi.2020.101965.

[9] L. Kelebercová and M. Munk, "Search queries related to COVID-19 based on keyword extraction," *Procedia Comput. Sci.*, vol. 207, pp. 2618–2627, 2022, doi: 10.1016/j.procs.2022.09.320.

[10] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," 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)*, Hong Kong, China: Association for Computational Linguistics, 2019, pp. 3980–3990. doi: 10.18653/v1/D19-1410.

[11] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," 2019, *arXiv*. doi: 10.48550/ARXIV.1910.01108.

[12] P. Bajaj *et al.*, "MS MARCO: A Human Generated MAchine Reading COmprehension Dataset," 2016, *arXiv*. doi: 10.48550/ARXIV.1611.09268.

[13] M. A. A. Fattah and R. Meiyanti, "Comparison Of Maximal Marginal Relevance (MMR) And Textrank Automatic Text Summarization Methods In Journals," vol. 2, 2024.

[14] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*, 1st ed. Cambridge University Press, 2008. doi: 10.1017/CBO9780511809071.

[15] J. Carbonell and J. Goldstein, "The use of MMR, diversity-based reranking for reordering documents and producing summaries," in *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, Melbourne Australia: ACM, Aug. 1998, pp. 335–336. doi: 10.1145/290941.291025.

[16] A. M. A. Zeyad and A. Biradar, "Abstractive Multi-Document Summarization: Exploiting Maximal Marginal Relevance and Pretrained Models," in *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, Delhi, India: IEEE, Jul. 2023, pp. 1–5. doi: 10.1109/ICCCNT56998.2023.10307351.

[17] C. Yoo and H. Lee, "Improving Abstractive Dialogue Summarization Using Keyword Extraction," *Appl. Sci.*, vol. 13, no. 17, p. 9771, Aug. 2023, doi: 10.3390/app13179771.

[18] S. Dhokane, C. Deshmukh, A. Bollabattin, S. Karande, B. Karangale, and P. S. Varade, "BM25 Implementation For Information Retrieval: Candidate Shortlister For Recruitment Process," in *2024 Intelligent Systems and Machine Learning Conference (ISML)*, Hyderabad, India: IEEE, May 2024, pp. 722–727. doi: 10.1109/ISML60050.2024.11007378.

[19] Y. Chen, Y. Guo, Y. Xie, and Z. Mi, "Legal and Regulation Retrieval System Based on Hierarchical Retrieval," in *2021 International Conference on Communications, Computing, Cybersecurity, and Informatics (CCCI)*, Beijing, China: IEEE, Oct. 2021, pp. 1–5. doi: 10.1109/CCCI52664.2021.9583204.

[20] M. Faysse *et al.*, "ColPali: Efficient Document Retrieval With," 2025.