



ระบบแนะนำอัตโนมัติสำหรับการหางานโดยใช้เทคนิคการประมวลผล
ภาษาธรรมชาติ

โดย

นาย ภัทรพล ทองยอดแก้ว

โครงการพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

วิทยาศาสตร์บัณฑิต

สาขาวิชาวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยธรรมศาสตร์

ปีการศึกษา 2567

ลิขสิทธิ์ของมหาวิทยาลัยธรรมศาสตร์

ระบบแนะนำอัตโนมัติสำหรับการหางานโดยใช้เทคนิคการประมวลผล
ภาษาธรรมชาติ

โดย

นาย ภัทรพล ทองยอดแก้ว

โครงงานพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

วิทยาศาสตร์บัณฑิต

สาขาวิชาวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยธรรมศาสตร์

ปีการศึกษา 2567

ลิขสิทธิ์ของมหาวิทยาลัยธรรมศาสตร์

Recommendation System for Job Finding using Natural Language
Processing Techniques

BY

MISTER PATTARAPOL TONGYODKAEW

A FINAL-YEAR PROJECT REPORT SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF BACHELOR OF SCIENCE

COMPUTER SCIENCE

FACULTY OF SCIENCE AND TECHNOLOGY

THAMMASAT UNIVERSITY

ACADEMIC YEAR 2024

COPYRIGHT OF THAMMASAT UNIVERSITY

มหาวิทยาลัยธรรมศาสตร์
คณะวิทยาศาสตร์และเทคโนโลยี

รายงานโครงการพิเศษ

ของ

นาย ภัทรพล ทองยอดแก้ว

เรื่อง

ระบบแนะนำอัตโนมัติสำหรับการหางานโดยใช้เทคนิคการประมวลผลภาษาธรรมชาติ

ได้รับการตรวจสอบและอนุมัติ ให้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร


หลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์

เมื่อ วันที่ 30 พฤษภาคม พ.ศ. 2568

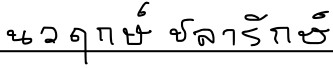
อาจารย์ที่ปรึกษา


(ผศ. ดร. ปกป้อง ส่องเมือง)

กรรมการสอบโครงการพิเศษ


(ผศ. ดร. วิลาวรรณ รักผางวงศ์)

กรรมการสอบโครงการพิเศษ


(อ. ดร. นวฤกษ์ ชลารักษ์)

มหาวิทยาลัยธรรมศาสตร์
คณะวิทยาศาสตร์และเทคโนโลยี

รายงานโครงการพิเศษ

ของ

นาย ภัทรพล ทองยอดแก้ว

เรื่อง


ระบบแนะนำอัตโนมัติสำหรับการหางานโดยใช้เทคนิคการประมวลผลภาษาธรรมชาติ

ได้รับการตรวจสอบและอนุมัติ ให้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
หลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์
เมื่อ วันที่ 30 พฤษภาคม พ.ศ. 2568

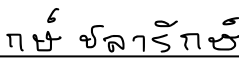
อาจารย์ที่ปรึกษา


(ผศ. ดร.ป๋อง ส่องเมือง)

กรรมการสอบโครงการพิเศษ


(ผศ. ดร.วิลาวรรณ รักผางค์)

กรรมการสอบโครงการพิเศษ


(อ. ดร.นวฤทธิ์ ชลารักษ์)

หัวข้อโครงการพิเศษ

ระบบแนะนำอัตโนมัติสำหรับการหางานโดยใช้เทคนิค

การประมวลผลภาษาธรรมชาติ

ชื่อผู้เขียน

นาย ภัทรพล ทองยอดแก้ว

ชื่อปริญญา

วิทยาศาสตร์บัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์

สาขาวิชา/คณะ/มหาวิทยาลัย

สาขาวิชาวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์และเทคโนโลยี

มหาวิทยาลัยธรรมศาสตร์

อาจารย์ที่ปรึกษาโครงการพิเศษ

ผศ. ดร. ปกป้อง ส่องเมือง

ปีการศึกษา

2567

บทคัดย่อ

ในปัจจุบัน ตลาดแรงงานมีความต้องการบุคลากรที่มีทักษะเฉพาะทางเพิ่มสูงขึ้นและเปลี่ยนแปลงได้อย่างตลอดเวลา งานวิจัยฉบับนี้นำเสนอระบบแนะนำรายวิชาแบบเฉพาะบุคคลที่อาศัยความคล้ายเชิงความหมายระหว่างคำอธิบายตำแหน่งงานและคำอธิบายรายวิชาเพื่อคัดเลือกคอร์สที่สอดคล้องกับความต้องการของผู้เรียน หลังจากกรองรายวิชาที่ผู้เรียนลงทะเบียนแล้ว ระบบจะดึงข้อมูลประกาศรับสมัครงานจาก JobsDB และบัญชีรายวิชามาแปรเป็นเวกเตอร์ข้อความด้วยเทคนิค embedding สี่แบบ ได้แก่ TF-IDF, Doc2Vec, และ Sentence-BERT จากนั้นคำนวณความคล้ายด้วย cosine similarity เพื่อสร้างลำดับคำแนะนำรายวิชา ซึ่งจะมีการยืนยันความแม่นยำด้วยการประเมินโดยใช้เทคนิค Normalized Discounted Cumulative Gain การผสานข้อมูลจากหลายแหล่งเข้ากับโมเดล NLP นี้จึงเป็นแนวทางที่มีศักยภาพในการปิดช่องว่างระหว่างพื้นฐานทางการศึกษาของผู้เรียนและความต้องการของตลาดแรงงานจริงในปัจจุบัน

คำสำคัญ: ประมวลผลภาษาทางธรรมชาติ, ความคล้ายคลึงโคไซน์, ระบบแนะนำการหางาน, ระบบแนะนำวิชาเรียน, ดัชนีผลตอบรับเชิงสะสมแบบลดทอนที่ปรับให้เป็นมาตรฐาน

Thesis Title	Recommendation System for Job Finding using Natural Language Processing Techniques
Author	Pattarapol Tongyodkaew
Degree	Bachelor of Science
Major Field/Faculty/University	Computer Science Faculty of Science and Technology Thammasat University
Project Advisor	Asst.Prof. Dr. Pokpong Songmuang
Academic Years	2024

ABSTRACT

In the present day, the demand for professionals with specialized skills is increasing. However, identifying the gap between knowledge gained from university education and the skills required by the job market remains a challenge. This paper presents a personalized course-recommendation system that leverages semantic similarity between job descriptions and course descriptions to suggest the most relevant coursework for learners. After filtering out courses a user has already completed, the system ingests web-scraped job postings from JobsDB and official university course catalogs, then transforms all text into vector representations using four embedding techniques, TF-IDF, Doc2Vec, and Sentence-BERT. Cosine similarity between a target job profile and each candidate course generates a ranked list of recommendations. The system's performance is quantitatively assessed through Normalized Discounted Cumulative Gain (NDCG) ranking based on relevance judgments from domain-specific undergraduate students, enabling direct comparison of each embedding strategy. By combining multi-modal data sources with NLP models and rigorous NDCG evaluation, this framework offers a promising solution for closing the gap between learners' academic backgrounds and real-world job demands.

Keywords: Natural Language Processing, Cosine similarity, Job recommendation system, Course recommendation system, Normalized Discounted Cumulative Gain

ACKNOWLEDGEMENTS

I would like to express my gratitude to Asst.Prof. Dr. Pokpong Songmuang, my advisor, for their invaluable guidance, many constructive feedback, expertise, and mentorship, they have been crucial in shaping this work. Your inspiring encouragement has kept me going and motivated me to strive for the better.

I am also grateful to my peers, who provided helpful insights. I would also like to extend my most sincere thanks to Mr. Surasit Uypatchawong for generously assisting me with data provision and for helping me so much during the course of this work. Their deep expertise and unending support have been immensely valuable.

Special thanks go to Thammasat University for providing the resources and support necessary to carry out this research.

Finally, I would like to thank my family and Yok for their words of encouragement throughout this journey.

Pattarapol Tongyodkaew

CONTENTS

	Page
ABSTRACT IN THAI	(1)
ABSTRACT IN ENGLISH	(2)
ACKNOWLEDGEMENTS	(5)
CONTENTS	(6)
LIST OF TABLES	(8)
LIST OF FIGURES	(9)
LIST OF ABBREVIATIONS	(10)
CHAPTER	
1 Introduction	11
1.1 Project Significance	11
1.2 Research Objectives	12
1.3 Scope	12
1.4 Benefits of the project	13
2 Literature reviews	14
2.1 Job Recommendation system	14
2.2 Word Embeddings	16
2.2.1 TF-IDF: A Simple and Foundational Technique	16
2.2.2 Doc2Vec: A Contextual Extension of Word2Vec	16
2.2.3 Sentence-BERT: Sentence-Level Semantic Understanding	17
2.3 Evaluation Metric: Normalized Discounted Cumulative Gain (NDCG)	18
2.3.1 Discounted Cumulative Gain (DCG)	18

	(7)
2.3.2 Ideal DCG (IDCG)	19
2.3.3 Normalized DCG (NDCG)	19
3 Research Methodology	20
3.1 System Overview	20
3.2 Data Sources & Collection	21
3.3 Embedding models/Vector similarities	22
3.3.1 TF-IDF + SVD Embeddings	22
3.3.2 Doc2Vec Embeddings	23
3.3.3 Sentence-BERT embedding	23
3.4 Tools & technologies	23
3.5 Evaluation	23
4 Results and discussion	25
4.1 Overview of Experiments	25
4.2 Quantitative Results	25
4.2.1 Per-Domain NDCG@10	25
4.2.2 Overall NDCG@10	25
4.2.3 Summary of Quantitative Findings	26
4.3 Runtime Efficiency Evaluation	26
5 Conclusion Limitation and Future Work	27
Reference List	29

LIST OF TABLES

	Page
Table 2.1 Feature of reviewed systems compared to proposed system	15
Table 4.1 Table of Per-Domain NDCG@10 Results	25
Table 4.2 Table of Overall NDCG@10 Results	26
Table 4.3 Table of all Runtime Results	27

LIST OF FIGURES

	Page
Figure 2.1 TF-IDF Formula	16
Figure 2.2 SBERT Siamese Architecture	18
Figure 2.3 DCG, IDCG, and NDCG Formula based on NDCG@3	19
Figure 3.1 Proposed System's Work flow	21

Abbreviations

Abbreviations

Definition

NLP

Natural Language Processing

SBERT

Sentence-BERT

TF-IDF

Term Frequency-Inverse Document
Frequency

NDCG

Normalized Discounted Cumulative Gain

Chapter 1

Introduction

1.1 Project Significance

In today's labor market, the job landscape in Thailand is increasingly shaped by technology, flexibility, and diversity. Structural challenges, intensified by the digital revolution, demand urgent adaptation from workers, businesses, and the government to remain globally competitive [1]. As highlighted by the National Economic and Social Development Council (NESDC) in the third quarter of 2024, long-term unemployment in Thailand rose by 16.2% compared to the previous year, with 81,000 individuals jobless for over a year [2]. Notably, a significant portion of the unemployed are university graduates, underscoring a potential mismatch between educational qualifications and market demands [2]. This dynamic era requires individuals to align their skills with evolving industry needs. This involves not only acquiring technical and soft skills but also embracing lifelong learning to stay competitive. Despite the growing demand for specialized knowledge and skills, many job seekers struggle to find job opportunities that match their qualifications while identifying areas for improvement remains a challenge for both individuals and educational institutions.

To address this issue, this project proposes a recommendation system that bridges the gap between job seekers' skills and market requirements. The system analyzes user's completed courses, compares them with employer-specified qualifications in job descriptions of their desired job, and calculates the similarity between the two. By identifying skill gaps, the system can recommend relevant university courses to help users enhance their qualifications. This approach empowers individuals to upskill effectively, increasing their chances of securing jobs aligned with their aspirations and market demands.

To quantitatively assess recommendation quality, we employ the Normalized Discounted Cumulative Gain (NDCG) metric, which accounts for both relevance judgments and their positions in the ranked list. We collected these

judgments from bachelor-level students in each target discipline—computer science, law, and healthcare—and then computed NDCG on their rated rankings to compare how well each embedding technique surfaced the most valuable up-skilling opportunities.

1.2 Research Objectives

This project aims to develop an NLP-based, automated job-course recommendation system that identifies gaps between users' academic histories and their desired job qualifications, then suggests the most relevant university courses. To meet this goal, we will:

1. Design and implement a recommendation engine that matches learners completed courses with employer-specified qualifications in target job descriptions to generate a ranked list of up-skilling suggestions.
2. Compare the performance of multiple embedding techniques (TF-IDF, Doc2Vec, Sentence-BERT) and similarity measures (e.g., cosine similarity) to identify the combination that yields the most accurate course rankings.
3. Evaluate recommendation quality via a human-judgment study we recruited bachelor students in computer science, law, and healthcare—to compute NDCG scores and directly compare embedding strategies.

1.3 Scope

1. Embedding techniques evaluated: TF-IDF, Doc2Vec, and Sentence-BERT
2. Similarity measure: Cosine similarity only
3. Job domains: Computer Science, Healthcare, and Law.
4. Course data source: Sakon Nakhon Rajabhat University's official catalog.

5. Language: Thai text for both job descriptions and course information. With some technical vocab still in English.
6. Evaluation metric: NDCG@10 based on field specific bachelor-level student relevance ratings

1.4 Benefits of the Project

1. Quickly identifies how well a user's completed coursework aligns with target job requirements and pinpoints missing competencies.
2. Guides users to the exact university courses they need to bridge identified gaps and improve employability.
3. Compares TF-IDF, Doc2Vec, and Sentence-BERT to reveal which embedding approach best supports job-course matching.

Chapter 2

Literature Reviews

This chapter reviews the foundations of our work in two parts. First, we examine leading job-recommendation platforms—JobsDB, JobTopGun, and LinkedIn—evaluating their methodologies, strengths, and limitations, and highlighting common gaps such as reliance on simple keyword matching and the lack of skill-gap analysis or personalized up-skilling suggestions. Second, we introduce the core NLP techniques used in this study—TF-IDF, Doc2Vec, and Sentence-BERT—outlining how each method works and why they are well-suited for improving semantic similarity and text representation in the context of job–course matching.

2.1 Job Recommendation systems

During the research phase, several existing job recommendation systems were examined, including JobsDB, JobTopGun, and LinkedIn, to understand their functionality and limitations compared to the proposed project.

JobsDB enables keyword, location, and industry-based job searches via string matching and basic filters. While effective for broad browsing, it offers no mechanism to analyze a user’s existing qualifications or identify which skills are missing.

JobTopGun extends filter options like company, salary range, job type, but its recommendation engine remains strictly keyword driven. Like JobsDB, it does not perform semantic matching of user profiles against job requirements, nor does it suggest upskilling resources.

LinkedIn leverages the skills section in a user’s profile to surface relevant job postings and “people-like-you” suggestions. However, its matching remains largely profile-keyword based. It does not analyze academic histories in depth, compute similarity scores between completed courses and job descriptions, or recommend specific courses to close skill gaps

Feature	JobsDB	JobTopGun	LinkedIn	Proposed System
Keyword-based job search	✓	✓	✓	X
Location and industry filters	✓	✓	✓	X
Semantic analysis of qualifications	X	X	X	✓
Identification of skill gaps	X	X	X	✓
Personalized course recommendations	X	X	X	✓
Calculation of similarity scores	X	X	X	✓
Advanced NLP for job-user matching	X	X	X	✓

Table 2.1 Feature of the systems compared to proposed system

All three platforms excel at surface-level job matching but share two key limitations. Absence of semantic analysis. They rely on exact-term matching rather than embedding-based similarity. And they have no tailored learning recommendations. They do not pinpoint missing skills or recommend concrete upskilling paths.

In contrast, our system applies NLP techniques (Doc2Vec and Sentence-BERT) to generate dense vector representations of both job descriptions and users completed courses. By computing cosine similarity between these embeddings, we can accurately quantify skill alignment and produce a ranked list of personalized course recommendations thereby bridging the divide between academic preparation and industry needs.

2.2 Word embeddings

Word embeddings map words or phrases into dense, continuous vectors in a high-dimensional space so that semantically similar items lie close together. Unlike sparse, count-based methods, embeddings capture both syntactic and semantic relationships, enabling downstream tasks, like recommendation or similarity search, to reason about meaning rather than just keyword overlap. Over time, embedding techniques have evolved from term-weighting schemes (e.g., TF-IDF) to neural approaches such as Doc2Vec, and most recently to context-aware transformer models like Sentence-BERT, each step bringing richer, more nuanced text representations.

2.2.1 TF-IDF: A Simple and Foundational Technique

Term Frequency-Inverse Document Frequency (TF-IDF) is one of the earliest and most straightforward approaches for representing textual data. It measures the importance of a word in a document relative to a collection of documents by assigning weights based on the frequency of occurrence and the rarity of the term in the corpus [3]. Its core idea is to assign each term t in a document d a weight that reflects both how common the term is in that document (Term Frequency) and how rare it is across the entire corpus D (Inverse Document Frequency).

$$TF(t, d) = \frac{\text{number of times } t \text{ appears in } d}{\text{total number of terms in } d}$$

$$IDF(t) = \log \frac{N}{1 + df}$$

$$TF - IDF(t, d) = TF(t, d) * IDF(t)$$

Figure 2.1 TF-IDF Formula

2.2.2 Doc2Vec: A Contextual Extension of Word2Vec

To address the limitations of Word2Vec, created by Google [4], Doc2Vec extends the idea to generate embeddings for entire documents, not just

words. This approach, based on Paragraph Vectors [5], captures the contextual meaning of sentences or paragraphs, making it more suitable for tasks like comparing job descriptions with user's taken courses. Doc2Vec effectively considers the broader context of text, aligning well with the requirements of this project.

2.2.3 Sentence-BERT: Sentence-Level Semantic Understanding

Sentence-BERT (SBERT) adapts the BERT architecture into a Siamese (or triplet) network to produce semantically meaningful, fixed-size embeddings for sentences or short texts [6]. Rather than using BERT's [CLS] token directly, SBERT fine-tunes two identical BERT encoders in parallel so that the cosine similarity between their pooled outputs reflects true semantic relatedness. It is also a multilingual model, meaning it has been trained on more than 50 languages, including Thai. Its ability to process and understand sentence-level semantics makes it a powerful choice for tasks that require contextual alignment, such as matching user profiles to job descriptions. The Siamese Architecture works as follows (see in figure 2.2).

1. Sentence Inputs (blue boxes): Two sentences (e.g., course text and job description).
2. BERT Encoders (white boxes): Identical BERT models with shared weights transform each sentence into contextual token representations.
3. Pooling (green boxes): Token outputs are pooled (mean/max) into fixed-size vectors h_1 and h_2 .

4. Cosine Similarity (arrow & label): The two vectors are compared via cosine similarity to yield a semantic similarity score.

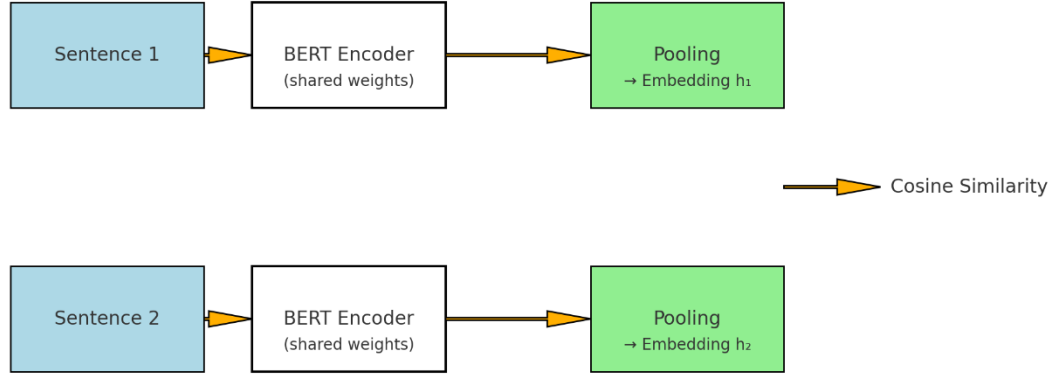


Figure 2.2 SBERT Siamese Architecture

2.3 Evaluation Metric: Normalized Discounted Cumulative Gain (NDCG)

To objectively measure how well our system ranks up-skilling suggestions, we adopt Normalized Discounted Cumulative Gain (NDCG), a standard metric in information retrieval and recommender-system research that rewards placing highly relevant items near the top of the list. Importantly, the theoretical properties of NDCG—such as its convergence behavior and its ability to consistently distinguish better ranking functions—have been rigorously analyzed by Wang et al. (2013), which gives us confidence in using NDCG as a reliable evaluation metric [7]. There is 3 important formula in calculating the NDCG score. All NDCG formulas can be seen in Figure 2.3 on page 9.

2.3.1 Discounted Cumulative Gain (DCG)

$$DCG_p = \sum_{i=1}^p \frac{2^{r_i} - 1}{\log_2(i + 1)}$$

DCG_p measures the total “gain” of the top p items in a ranked list by summing each item’s graded relevance r_i while diminishing its contribution by a

logarithmic factor based on its position. In other words, highly relevant courses near the top contribute more, and relevance further down the list is increasingly discounted.

2.3.2 Ideal DCG (IDCG)

$$IDCG_p = \sum_{i=1}^p \frac{2^{r_i^*} - 1}{\log_2(i + 1)}$$

$IDCG_p$ is the maximum possible DCG for those same p items, it's computed by taking the same relevance scores, sorting them in descending order (so the best items occupy the highest ranks), and then applying the **DCG** formula. **IDCG** represents the “perfect” ranking benchmark.

2.3.3 Normalized DCG (NDCG)

$$NDCG_p = \frac{DCG_p}{IDCG_p}$$

$NDCG_p$ normalizes DCG_p by dividing it by $IDCG_p$, yielding a score between 0 and 1. An **NDCG** of 1 means your ranking exactly matches the ideal ordering, while lower values indicate the degree to which your system's ordering deviates from perfection.

$$DCG@3_x = \frac{0}{\log_2(1+1)} + \frac{0}{\log_2(2+1)} + \frac{1}{\log_2(3+1)} = 1.88685$$

$$IDCG@3_x = \frac{1}{\log_2(1+1)} + \frac{1}{\log_2(2+1)} + \frac{1}{\log_2(3+1)} = 2.13093$$

$$NDCG@3_x = \frac{DCG_x}{IDCG_x} = \frac{0.5}{2.13093} \approx 0.23464$$

Figure 2.3 DCG, IDCG, and NDCG Formula based on @3

Chapter 3

Research Methodology

3.1 System Overview

The core functionality of the proposed job recommendation system is based on a comparison between the users' academic histories, input as name of the courses they've taken, and the job requirements specified in job descriptions. The system diagram can be seen in figure 3.1 with the explanation as follows

1. **Input Data:**

Learner profile: Desired job title and list of completed course names.

Course catalog: Full set of university course descriptions.

Job postings: Collected job descriptions for the target role.

2. **Embedding:** Transform each cleaned document (course description or job description) into a fixed-length embedding vector using one of the evaluated techniques (TF-IDF / Doc2Vec / SBERT).
3. **Candidate Course:** Remove any course whose name appears in the learner's completed list so that recommendations only include new learning opportunities.
4. **Similarity Computation:** Cosine similarity are used to compute the cosine similarity between the job-description embedding and each remaining course embedding, yielding a relevance score for every candidate course.
5. **Course Recommendation:** Sort candidate courses by descending similarity score. Present the top-10 highest-scoring courses as personalized upskilling suggestions to the learner.

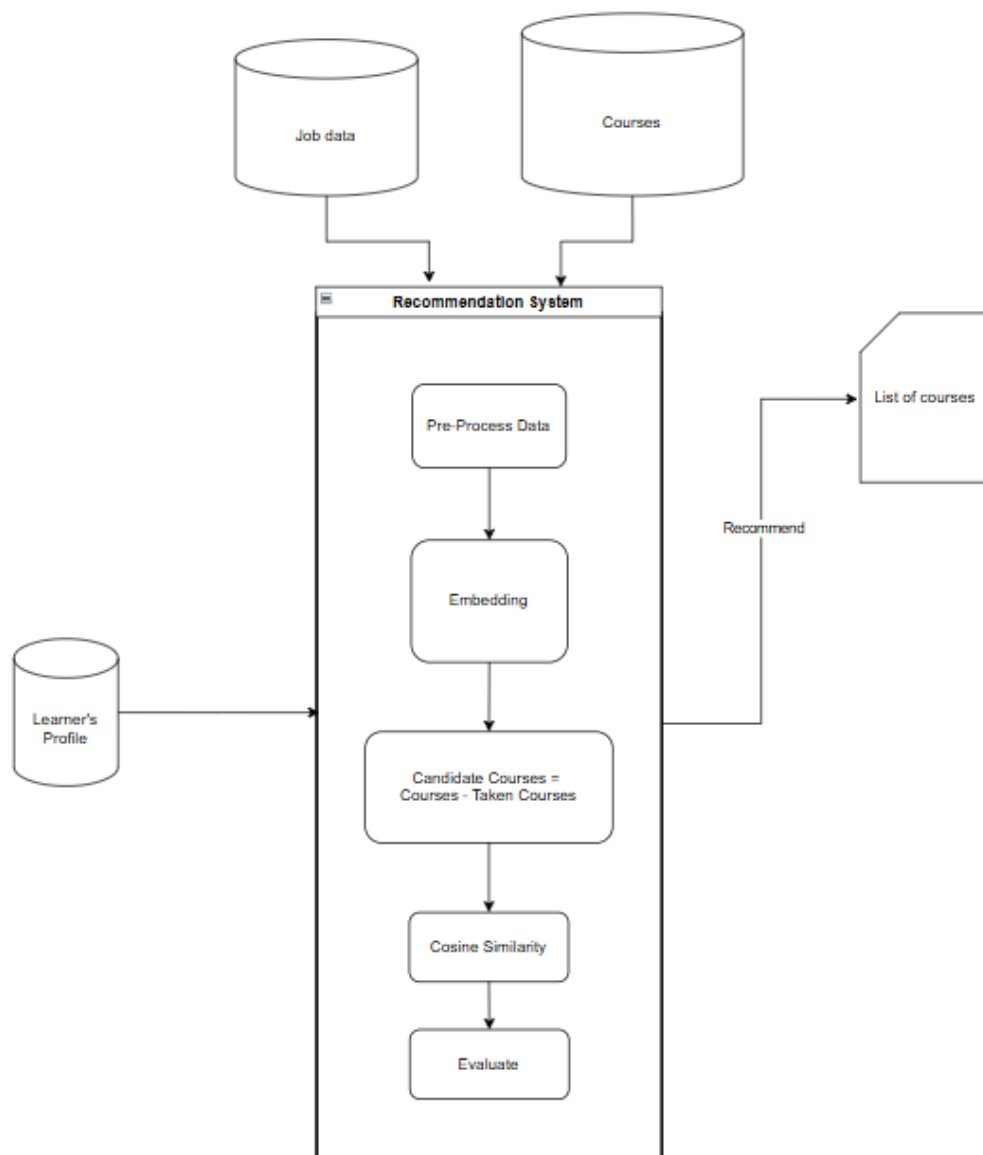


Fig 3.1 System Overview Diagram

3.2 Data Sources & Collection

3.2.1 Job Postings

- **Source:** JobsDB and JobTopGun (February–March 2025).
- **Collection method:** Automated web scraping using Python (Selenium for dynamic pages; BeautifulSoup for HTML parsing).

- **Initial volume:** about 300 raw postings across Computer Science, Healthcare, and Law domains.
- **Consolidation into aggregated jobs:**
 1. Grouped postings by identical job title.
 2. For each group, merged the “core” sentences from all descriptions to form one composite job profile.
- **Final dataset:** 48 aggregated job descriptions ($\approx 15\text{--}16$ per domain).
- **Cleaning steps:**
 1. **Language check:** Ensure descriptions are predominantly Thai
 2. **HTML/markup removal:** Strip all tags, scripts, and inline CSS.
 3. **Link deletion:** Remove URLs and email addresses.
 4. **Whitespace normalization:** Collapse repeated spaces/newlines into single spaces.

3.2.2 Course Catalog

- **Source:** Official course listings from Sakon Nakhon Rajabhat University.
- **Initial volume:** 3,000 courses across all faculties.
- **Sampling for experiment:** 144 courses (≈ 48 per domain) were chosen to mirror the job domains, focusing on core and elective offerings most relevant to each field.
- **Cleaning steps:** HTML/markup removal and link deletion as above.

3.3 Embedding models/Vector similarities

The embedding models used in this project are TF-IDF, Doc2Vec, and Sentence-Bert. Embeddings from these models are then calculated to find vector similarities using Cosine Similarity.

3.3.1 TF-IDF + SVD Embeddings:

In our pipeline, we turn each job and course description into a 300-dimensional dense vector by first computing TF-IDF weights over a shared vocabulary, then applying truncated SVD to capture latent topics.

3.3.2 Doc2Vec Embeddings:

Because there isn't a publicly available Thai Doc2Vec, we train our own PV-DM (Paragraph Vector–Distributed Memory) model on the combined corpus of job and course texts. This lets the model learn Thai-specific semantics directly from our data.

A Gensim Doc2Vec is initialized with a 300-dimensional vector size to match our other embeddings, a context window of 5, min_count=1 to include all terms, 4 worker threads, and 40 training epochs. After building its vocabulary from the tagged documents, the model is trained over all job and course texts.

3.3.3 Sentence-BERT embedding:

For deep, context-aware representations of Thai job and course texts, we employ the multilingual MiniLM SBERT model (paraphrase-multilingual-MiniLM-L12-v2), which natively supports Thai and embedded English keywords.

We load the pre-trained SBERT checkpoint onto GPU (if available) and enforce a 512-token maximum to match the model's capacity

We encode all documents in batches of 32, truncating to 512 tokens and converting outputs to NumPy arrays for efficiency:

3.4 Tools & technologies

The scraping of jobs data are done on Google Colab with Python language. The model experiment and recommendation system are written in Python on Visual Studio Code.

3.5 Evaluation

We assessed the quality of our course-recommendation rankings using Normalized Discounted Cumulative Gain at cutoff 10 (NDCG@10), based on human relevance judgments. The evaluation procedure was as follows:

Participants: 30 bachelor-level students (10 each from Computer Science, Law, and Healthcare).

Judgment task: Each participant received a top-10 course recommendations (from three aggregated job profiles) generated by each embedding method. They rated each recommended course on a 4-point scale: 0 = Not relevant, 1 = Slightly relevant, 2 = Relevant, 3 = Highly relevant.

Then we averaged NDCG@10 across all participants and job profiles for each embedding method (TF-IDF+SVD, Doc2Vec, and SBERT).

Chapter 4

Results and discussion

4.1 Overview of Experiments

We created three composite job profiles, one each for Computer Science, Law, and Healthcare and generated three top-10 course lists (TF-IDF+SVD, Doc2Vec, SBERT) per profile. For each domain:

- 10 in-field students rated the three recommendation lists for their field’s job profile on a 0–3 relevance scale.
- We computed NDCG@10 per student per method. This yields 10 NDCG scores for each embedding in each domain.

4.2 Quantitative Results

4.2.1 Per-Domain NDCG@10

Domain	TF-IDF+SVD	Doc2Vec	SBERT
Law	0.926	0.871	0.943
Computer Science	0.809	0.913	0.938
Healthcare	0.892	0.866	0.910

Table 4.1 Table of Per-Domain NDCG@10 Results

Sentence-BERT leads in all three fields, with its strongest advantage in Computer Science (0.938 vs. 0.809 for TF-IDF). TF-IDF+SVD surprisingly performs nearly on par with SBERT in Law (0.926 vs. 0.943) and Healthcare (0.892 vs. 0.910), this might be that keyword importance alone captures much of the domain terminology there. Doc2Vec ranked worse than SBERT and TF-IDF in Law and Healthcare, but outperforms TF-IDF in CS, indicating its ability with technical content.

4.2.2 Overall NDCG@10

Averaged across all 30 student ratings (10 per field), Sentence-BERT achieves the highest overall NDCG@10 (0.930), outperforming Doc2Vec (0.883) by

0.047 and TF-IDF+SVD (0.875) by 0.055. The consistent strength of SBERT highlights the value of deep contextual embeddings on both Thai with some English vocabs when aligning course descriptions with job requirements. Doc2Vec provides a strong middle ground, offering a lightweight yet semantically richer alternative to raw TF-IDF, which still hold its own in less technical domains.

Technique	Average NDCG@10
Sentence-BERT	0.930
Doc2Vec	0.883
TF-IDF+SVD	0.875

Table 4.2 Table of Overall NDCG@10 Results

4.2.3 Summary of Quantitative Findings

Taken together, these results clearly display that Sentence-BERT delivers the strongest ranking quality across all three domains and altogether, with the overall NDCG@10 of 0.930. Its deep, contextual embeddings give it a consistent edge, particularly in the technical Computer Science field where it outperforms TF-IDF+SVD by over 0.13.

However, **TF-IDF+SVD** proves surprisingly competitive in Law (0.926) and Healthcare (0.892), suggesting that straightforward keyword weighting capability of TF-IDF still captures much of the essential terminology in less jargon-dense domains. **Doc2Vec** occupies the middle ground, outperforming TF-IDF in Computer Science but falling slightly behind SBERT overall.

4.3 Runtime Efficiency Evaluation

To evaluate the runtime efficiency of each semantic similarity model used in the recommendation system—TF-IDF, Doc2Vec, and SBERT—this study measured and compared the average embedding time and similarity computation time across ten runs for each model. Table 4.3 summarizes the average times and their

corresponding standard deviations, which indicate the variability in performance across runs.

Technique	Avg Embedding Time (s)	Standard Deviation ($\pm s$)	Avg Similarity Time (s)	Standard Deviation ($\pm s$)	Total Runtime (s)
TF-IDF	0.209	± 0.036	0.051	± 0.008	0.260
Doc2Vec	1.221	± 0.023	0.054	± 0.008	1.275
SBERT	3.978	± 0.048	0.064	± 0.005	4.042

Table 4.3 Table of all Runtime Results

TF-IDF demonstrated the fastest processing overall, with an average embedding time of 0.209 seconds and similarity computation time of 0.051 seconds, resulting in a total average time of 0.260 seconds. Doc2Vec followed with a significantly higher average embedding time of 1.221 seconds and a similar similarity computation time of 0.054 seconds, totaling approximately 1.275 seconds per run. SBERT, while offering state-of-the-art semantic understanding, had the highest runtime with an average embedding time of 3.978 seconds and similarity computation time of 0.064 seconds, leading to a total average time of 4.042 seconds.

The standard deviation values across all models were relatively low, indicating consistent performance between runs. Notably, SBERT's higher embedding time reflects the computational cost associated with deep contextual language models. These findings suggest a trade-off between runtime efficiency and semantic accuracy, which is discussed further in the model performance section.

Chapter 5

Conclusion, Limitation and Future Work

This project addresses the real-world issue of skill mismatches in the labor market by delivering a personalized, data-driven recommendation system that guides learners from self-assessment to action. The system aligns completed university coursework with job requirements using three distinct semantic embedding strategies, TF-IDF with SVD, Doc2Vec, and Sentence-BERT, and ranks job-course relevance using NDCG@10, based on domain-expert student judgments.

The results demonstrate that Sentence-BERT consistently achieved the highest ranking performance across all three tested fields. It scored 0.943 in Law, 0.938 in Computer Science, and 0.910 in Healthcare, with an overall NDCG@10 score of 0.930. These findings highlight the strength of deep contextual embeddings in capturing nuanced relationships within complex job descriptions.

Doc2Vec provided a reasonable middle ground, achieving an overall score of 0.883. However, it underperformed in domains such as Law and Healthcare, where capturing specific domain terminology and co-occurrence patterns was more critical. TF-IDF + SVD, with an overall score of 0.875, remained surprisingly competitive, particularly in Law and Healthcare. Its performance suggests that in fields dominated by a stable set of key terms, traditional statistical embeddings may still offer meaningful relevance.

In terms of runtime efficiency, TF-IDF was the fastest, followed by Doc2Vec and then SBERT, which was the slowest due to its complex neural architecture. These results suggest a trade-off between semantic accuracy and computational efficiency that stakeholders must consider when deploying such systems in practice. These are some of the limitations I have found throughout the making of this system.

1. **Incomplete data coverage:** Both our course catalog (144 sampled courses) and the aggregated job descriptions may not fully represent all

topics or qualifications required by certain roles, which can lead to gaps in the recommendations.

2. **Generic job postings:** Some scraped job descriptions list only the required major or degree rather than detailed qualifications, limiting the depth of semantic matching.

3. **Company-specific requirements:** In fields like Law, requirements vary by company type (e.g., banking vs. insurance), yet many postings do not specify these distinctions.

4. **Single role per domain:** We evaluated only one aggregated job profile in each field, which may not generalize across all job titles

5. **Static snapshot:** Job postings and course offerings evolve; our system does not dynamically update its dataset.

These are some of the future works that could be beneficial to this system:

1. **Hybrid and learning-to-rank approaches:** Combine embedding-based cosine scores with metadata features (e.g. course level, credit hours, prerequisite structure) in a learned ranking model for finer-grained recommendations.

2. **Explore deeper similarity analyses:** Investigate semantic overlap between different job descriptions or qualifications to improve the candidate-filtering and aggregation steps.

3. **Fine-tune embedding models:** Perform supervised or unsupervised fine-tuning of SBERT (and potentially Doc2Vec) on an expanded, in-domain Thai corpus to better capture specialized terminology.

4. **Dynamic requirement updates:** Automate periodic re-scraping and re-processing of job postings so that the system continually adapts to evolving market demands and emerging skills.

Reference list

- [1] Rhee, C. (2024, February 6). *Challenges awaiting the Thai labor market in the digital age*. SIAM NEWS NETWORK PTE. LTD. <https://www.thailand-business-news.com/education/125276-challenges-awaiting-the-thai-labor-market-in-the-digital-age>
- [2] Turner, R. (2024, November 26). *Thai graduates face high unemployment despite stable job market*. Thaiger. <https://thethaiger.com/news/national/thai-graduates-face-high-unemployment-despite-stable-job-market>
- [3] Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5), 513–523.
- [4] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. In *arXiv:1301.3781*. <https://doi.org/10.48550/arXiv.1301.3781>
- [5] Le, Q., & Mikolov, T. (2014). Distributed representations of sentences and documents. *Proceedings of the 31st International Conference on Machine Learning (ICML)*, 1188–1196. <https://arxiv.org/abs/1405.4053>
- [6] Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 3982–3992. <https://doi.org/10.48550/arXiv.1908.10084>
- [7] Wang, Y., Wang, L., Li, Y., He, D., & Liu, T.-Y. (2013). A theoretical analysis of NDCG type ranking measures. *Annual Conference Computational Learning Theory*, 30, 25–54. <https://proceedings.mlr.press/v30/Wang13.html>