ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA TRÍ TUỆ NHÂN TẠO

BÁO CÁO ĐỒ ÁN MÔN HỌC

ĐỀ TÀI

GỢI Ý CÂU HỎI PHỎNG VẤN THÔNG MINH CHO NHÀ TUYỂN DỤNG DỰA TRÊN HỒ SƠ ỨNG VIÊN

Môn học: CS116.E31 – Lập trình Python cho máy học

Giảng viên hướng dẫn: Nguyễn Hữu Quyền

Thực hiện bởi nhóm 14, bao gồm:

1. Hàng Xương Hoàn 25410053
2. Nguyễn Trung Kiên 25410160
3. Hoàng Xuân Việt 25410072
4. Huỳnh Hoàng Hảo 25410047
5. Võ Hoàng Lộc 25730003

TP. HỒ CHÍ MINH, 08/2025

MỤC LỤC

DANH SÁCH HÌNH, BẢNG

1. Giới thiệu

1.1. Đặt vấn đề

Trong bối cảnh thị trường lao động ngày càng cạnh tranh, các doanh nghiệp thường nhận được số lượng lớn hồ sơ xin việc (CV) cho mỗi vị trí tuyển dụng. Tuy nhiên, phần lớn các CV này tồn tại ở dạng **phi cấu trúc**, đa dạng về cách trình bày, ngôn ngữ, thậm chí có lỗi chính tả hoặc thiếu chuẩn hóa. Việc phân tích thủ công không chỉ tốn nhiều thời gian, chi phí mà còn dễ dẫn đến sai sót, chủ quan từ phía nhà tuyển dụng.

Mặt khác, quy trình tuyển dụng hiện đại không chỉ dừng lại ở việc đánh giá kỹ năng cứng, mà còn cần xem xét sự phù hợp về kinh nghiệm, trình độ học vấn, khả năng phát triển lâu dài và thậm chí là mức độ hòa nhập văn hóa doanh nghiệp. Trong khi đó, nguồn ứng viên tiềm năng có thể đến từ nhiều quốc gia với các ngôn ngữ khác nhau, càng làm tăng độ phức tạp cho quy trình lọc và đánh giá hồ sơ.

Chính vì vậy, nhu cầu về một **hệ thống phân tích CV tự động, thông minh** ngày càng trở nên cấp thiết. Hệ thống này cần có khả năng xử lý dữ liệu ngôn ngữ tự nhiên, đánh giá sự phù hợp giữa CV và mô tả công việc (Job Description – JD), đồng thời hỗ trợ nhà tuyển dụng trong các bước tiếp theo của quy trình.

1.2. Mục tiêu

Đồ án này hướng tới xây dựng một **hệ thống phân tích CV tự động ứng dụng Trí tuệ Nhân tạo (AI)**, trong đó kết hợp nhiều kỹ thuật khác nhau như **Xử lý Ngôn ngữ Tự nhiên (NLP), Biểu diễn ngữ nghĩa (Semantic Embedding), Học máy (Machine Learning), và Sinh câu hỏi ngữ nghĩa (Semantic Question Generation)**.

Các mục tiêu cụ thể của hệ thống bao gồm:

- **Tự động hóa xử lý CV phi cấu trúc:** Sử dụng kỹ thuật NLP (tokenization, stopword removal, NER) để trích xuất thông tin quan trọng từ hồ sơ.
- **Đo lường mức độ phù hợp CV–JD:** Áp dụng TF-IDF, Cosine Similarity và Sentence-BERT để so sánh về mặt ngữ nghĩa.
- **Phân loại ứng viên:** Ứng dụng mô hình ML (Logistic Regression, Random Forest) nhằm dự đoán mức độ phù hợp của ứng viên so với vị trí tuyển dụng.
- **Hỗ trợ phỏng vấn:** Sinh câu hỏi phỏng vấn cá nhân hóa dựa trên kỹ năng và kinh nghiệm của ứng viên.
- **Đề xuất lộ trình phát triển:** Phát hiện khoảng trống kỹ năng so với yêu cầu công việc và gợi ý hướng bổ sung.

Thông qua việc tích hợp các kỹ thuật AI này, hệ thống kỳ vọng sẽ:

- Nâng cao **tốc độ và độ chính xác** trong sàng lọc hồ sơ.
- Giảm thiểu sai sót do yếu tố chủ quan.
- Cung cấp **công cụ hỗ trợ ra quyết định** cho nhà tuyển dụng.
- Đồng thời mang lại **trải nghiệm tốt hơn** cho ứng viên trong suốt quá trình tuyển dụng.

2. Cơ sở lý thuyết:

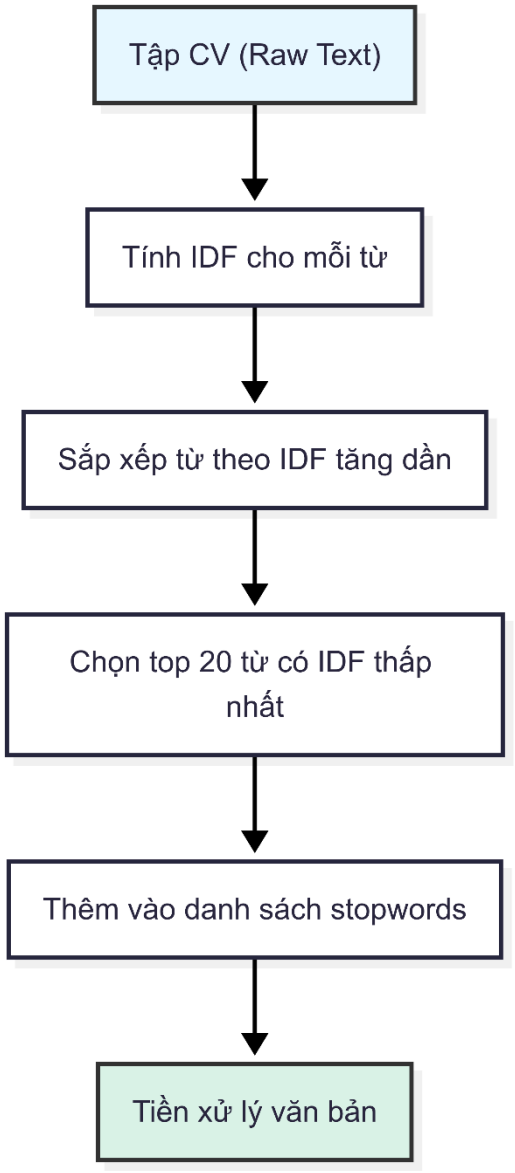
Hệ thống phân tích CV tự động được xây dựng dựa trên nhiều kỹ thuật thuộc lĩnh vực **Xử lý Ngôn ngữ Tự nhiên (NLP)** và **Học Máy (ML)**. Mỗi kỹ thuật giữ một vai trò riêng, từ xử lý dữ liệu đầu vào, đánh giá mức độ phù hợp, đến hỗ trợ nhà tuyển dụng trong giai đoạn phỏng vấn. Dưới đây là các thành phần lý thuyết cốt lõi.

2.1. Xử lý ngôn ngữ tự nhiên (Natural Language Processing – NLP)

Xử lý ngôn ngữ tự nhiên là bước nền tảng để làm sạch và chuẩn hóa dữ liệu văn bản từ CV và JD, nhằm hỗ trợ các tác vụ phân tích sâu hơn. Một số kỹ thuật cơ bản:

- **Tokenization (Tách từ):** Phân chia văn bản thành các đơn vị nhỏ (token) như từ hoặc cụm từ, giúp máy tính dễ dàng xử lý.
- **Stopword Removal (Loại bỏ từ dừng):** Loại bỏ các từ phổ biến nhưng ít giá trị phân tích như “and”, “is”, “for”, nhằm tập trung vào nội dung quan trọng.
- **Named Entity Recognition (NER):** Nhận diện và phân loại các thực thể quan trọng trong CV (ví dụ: kỹ năng, tổ chức, số năm kinh nghiệm, học vị).
- **Skill Extraction**: So khớp từ khóa từ danh sách kỹ năng (SKILL_LIST) và dataset skill2vec.
- **Transfer Learning cho mở rộng kỹ năng:** kỹ thuật transfer learning để mở rộng danh sách kỹ năng từ dữ liệu không nhãn

Ngoài ra, hệ thống còn áp dụng **cơ chế tự động phát hiện stopwords** dựa trên phân bố IDF (Inverse Document Frequency) của tập dữ liệu CV. Cụ thể, các từ xuất hiện quá phổ biến trong toàn bộ tập dữ liệu (có giá trị IDF thấp) sẽ được bổ sung vào danh sách stopwords. Điều này giúp loại bỏ các từ ít có giá trị phân biệt (ví dụ: 'worked', 'developed', 'using') và tăng độ chính xác khi trích xuất các kỹ năng chuyên môn đặc thù. Phương pháp này đặc biệt hữu ích khi xử lý các CV từ nhiều ngành nghề khác nhau.

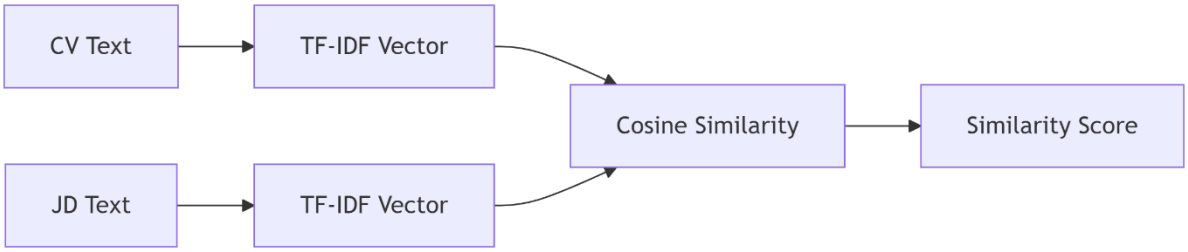


Hình 1 : Cơ chế tự động phát hiện stopwords

2.2. Độ tương đồng ngữ nghĩa dựa trên TF-IDF và Cosine Similarity

Để đo lường mức độ phù hợp giữa CV và JD, cần tính toán độ tương đồng giữa hai văn bản. Một phương pháp phổ biến là kết hợp **TF-IDF (Term Frequency – Inverse Document Frequency)** và **Cosine Similarity**:

- **TF-IDF:** Biểu diễn văn bản thành vector trọng số, phản ánh tầm quan trọng của từ trong toàn bộ tập dữ liệu.
- **Cosine Similarity:** Đo lường độ gần gũi giữa hai vector bằng cosin của góc giữa chúng, cho giá trị trong khoảng [0,1], càng gần 1 thì độ tương đồng càng cao.



Hình 2 : Minh họa TF-IDF

Phương pháp này được triển khai trong hàm `tfidf_cosine_similarity()`

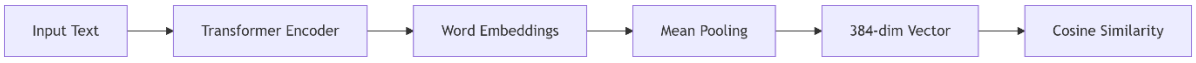
2.3. Sentence Embedding (Sentence-BERT)

Trong nhiều trường hợp, TF-IDF chưa nắm bắt đầy đủ ngữ nghĩa ngữ cảnh. Do đó, cần sử dụng **sentence embedding**:

- **Mô hình all-MiniLM-L6-v2:** Chuyển đổi câu thành vector 384 chiều, đo độ tương đồng ngữ nghĩa bằng cosine similarity**.**
- **Ứng dụng:** So sánh semantic similarity giữa CV và JD, gợi ý câu hỏi phỏng vấn.

Về mặt kỹ thuật, Sentence-BERT (**SBERT**) tối ưu hóa kiến trúc BERT gốc thông qua **phép đo cosine triplet loss**. Cơ chế này cho phép so sánh trực tiếp độ tương đồng giữa các câu mà không cần tính toán pairwise tốn kém. Công thức tính toán:

$$L=\max(0,\cos(\text{anchor},\text{negative})-\cos(\text{anchor},\text{positive})+\epsilon)$$



Hình 3 : Kiến trúc Sentence-BERT

Trong hệ thống, chúng tôi sử dụng **all-MiniLM-L6-v2** - phiên bản tối ưu cho tốc độ xử lý với độ chính xác cao (79.7% trên benchmark STS). Mô hình này có kiến trúc 6 lớp ẩn (hidden layers) và tạo embedding 384 chiều, cân bằng giữa hiệu năng và độ chính xác.

2.4. Phân loại bằng Machine Learning

Sau khi trích xuất đặc trưng (feature engineering), bài toán phân tích CV có thể được coi là một bài toán phân loại nhị phân (phù hợp/không phù hợp). Các mô hình sử dụng:

- **Logistic Regression:** Mô hình tuyến tính đơn giản, dễ giải thích, thích hợp cho dữ liệu có quan hệ tuyến tính.

- **Random Forest:** Mô hình ensemble dựa trên nhiều cây quyết định, mạnh trong việc xử lý dữ liệu phức tạp và quan hệ phi tuyến.
- **MLP Classifier :** Mạng neural 1 lớp ẩn (10 neurons) để học biểu diễn phức tạp
- **Đánh giá mô hình:** Sử dụng các chỉ số như Precision, Recall, F1-Score và AUC-ROC để đảm bảo độ tin cậy.

2.5. Semantic Question Generation

Một ứng dụng nâng cao của NLP là sinh câu hỏi phỏng vấn cá nhân hóa cho từng ứng viên.

- **Semantic Search:** Hệ thống tìm kiếm câu hỏi có sẵn trong dataset (200 câu từ Kaggle) có embedding gần nhất với CV, sử dụng mô hình `paraphrase-multilingual-MiniLM-L12-v2`
- **Ý tưởng:** Hệ thống tạo ra câu hỏi dựa trên kỹ năng, kinh nghiệm được trích xuất từ CV.
- **Phương pháp Semantic Generation:** Dựa trên các mô hình ngôn ngữ hiện đại để sinh câu hỏi mang tính tự nhiên, đa dạng, và không bị trùng lặp ngữ nghĩa.
- **Ví dụ:** Với kỹ năng "Python", hệ thống có thể sinh ra câu hỏi như *"Bạn có thể mô tả dự án phức tạp nhất mà bạn đã triển khai bằng Python không?"*.

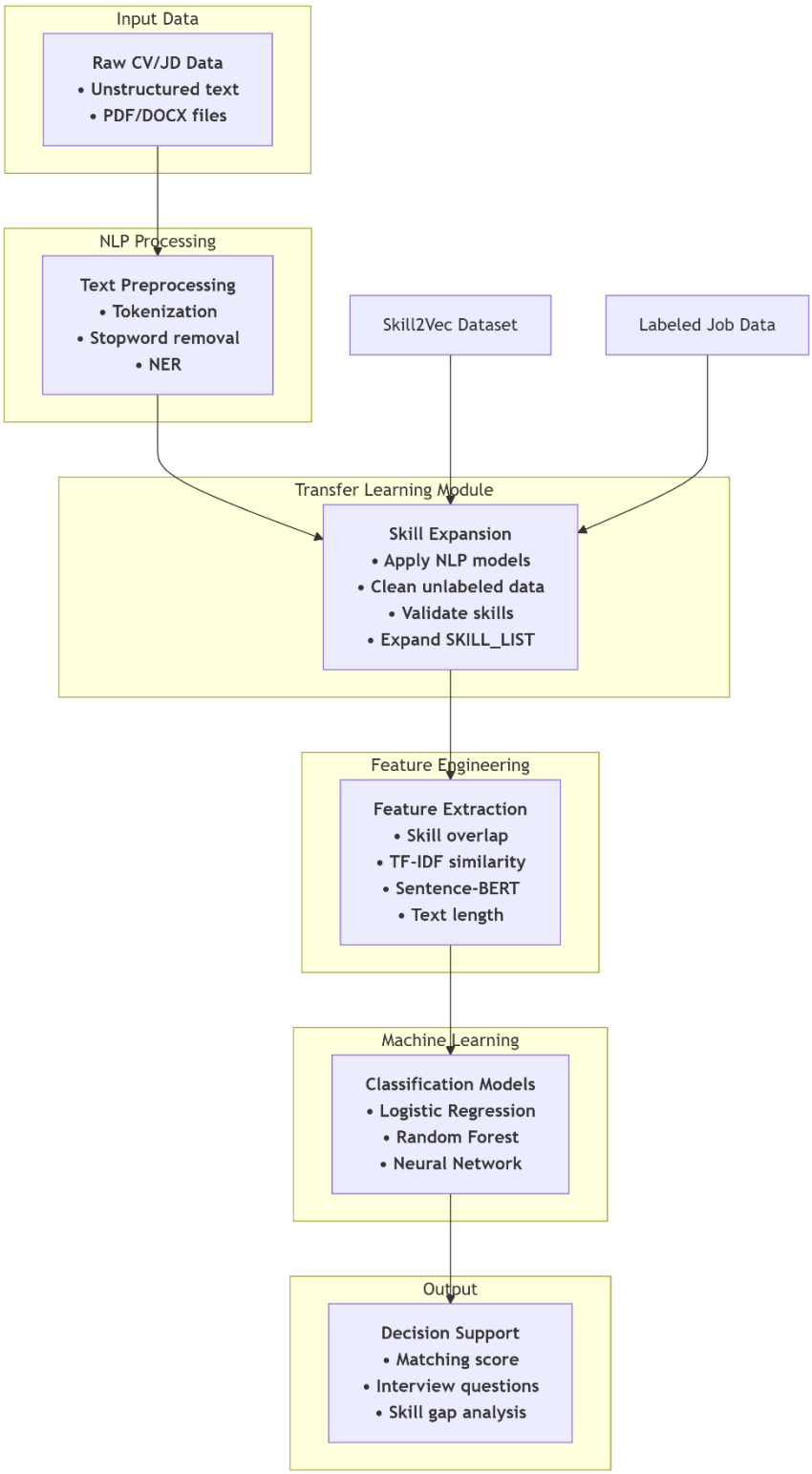
Cơ chế semantic search được thực hiện qua hàm `print_matched_questions()`

3. Thiết kế hệ thống

3.1. Quy trình tổng thể

Hệ thống phân tích CV tự động được thiết kế theo một **pipeline nhiều tầng**, từ bước tiền xử lý văn bản đến phân loại ứng viên và tạo câu hỏi phỏng vấn. Toàn bộ quy trình gồm các giai đoạn chính:

1. **Tiền xử lý dữ liệu (NLP):** Làm sạch và chuẩn hóa văn bản CV/JD thông qua tokenization, stopword removal và Named Entity Recognition (NER).
2. **Feature Engineering:** Tính toán skill overlap, TF-IDF similarity, sentence embedding similarity.
3. **Huấn luyện mô hình:** Train/test các mô hình phân loại, lựa chọn model tốt nhất.
4. **Dự đoán và gợi ý:** Đánh giá CV-JD match và gợi ý câu hỏi phỏng vấn.



Hình 4 Sơ đồ khối toàn hệ thống

3.2. Nội dung các phương pháp áp dụng

(1) Xử lý ngôn ngữ tự nhiên (NLP)

Đây là giai đoạn nền tảng để biến dữ liệu phi cấu trúc (CV, JD) thành dạng có thể khai thác:

- **Tokenization:** Tách câu văn thành các token (từ/cụm từ).
- **Stopword Removal:** Loại bỏ các từ không mang nhiều thông tin (như "is", "and", "off").
- **Named Entity Recognition (NER):** Nhận diện thực thể trong CV:
  - Kỹ năng: *Python, Java, AWS, ...*
  - Tổ chức: *Google, Amazon*
  - Thời gian kinh nghiệm: *5 years*
  - Học vấn: *University of California, Master*

JOHN SMITH
San Francisco, CA | john.smith@email.com | (123) 456-7890
EDUCATION
[Stanford University] (ORG) - [Master of Computer Science] (EDU) | 2015-2017
[University of California] (ORG) - [Bachelor of Science] (EDU) | 2011-2015
EXPERIENCE
[Google] (ORG) | Senior Software Engineer | [5 years] (DUR)
- Developed machine learning models using [Python] (SKILL) and [TensorFlow] (SKILL)
- Led team implementing [natural language processing] (SKILL) solutions
[Amazon] (ORG) | Software Engineer | [2 years] (DUR)
- Built cloud services using [AWS] (SKILL) and [Java] (SKILL)
- Optimized databases reducing latency by 30%
SKILLS
Programming: [Python] (SKILL), [Java] (SKILL), [C++] (SKILL)
ML Frameworks: [TensorFlow] (SKILL), [PyTorch] (SKILL), [Scikit-learn] (SKILL)
Tools: [Git] (SKILL), [Docker] (SKILL), [Kubernetes] (SKILL)

Hình 5: CV mẫu và highlight các thực thể kỹ năng, tổ chức, học vấn

(2) Chuyển giao Tri thức (Transfer Learning Module)

Transfer Learning Module đóng vai trò quan trọng trong việc mở rộng cơ sở kiến thức về kỹ năng từ dữ liệu không nhãn. Mô-đun này:

- Áp dụng mô hình NLP đã huấn luyện trên dữ liệu có nhãn
- Xử lý dữ liệu thô từ Skill2Vec Dataset qua 3 bước:
  1. Làm sạch (loại bỏ nhiễu, từ không liên quan)
  2. Kiểm định kỹ thuật (technical validation)
  3. Loại bỏ trùng lặp (deduplication)
- Mở rộng danh sách kỹ năng từ 43 lên 128 kỹ năng
- Cung cấp danh sách kỹ năng đầy đủ cho giai đoạn Feature Engineering

(3) Biểu diễn và đo độ tương đồng

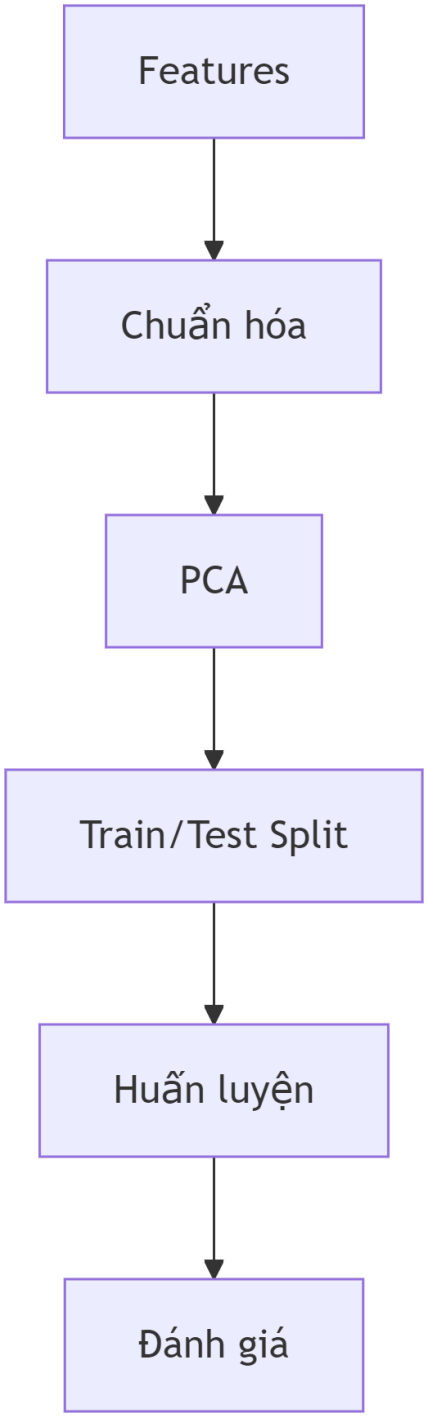
- **TF-IDF + Cosine Similarity:**

- TF-IDF biểu diễn văn bản dưới dạng vector dựa trên tần suất từ.
- Cosine Similarity đo mức độ tương đồng giữa CV và JD qua góc giữa 2 vector.

- **Sentence Embedding (Sentence-BERT):**
  - Mỗi câu từ CV/JD được mã hóa thành vector đa chiều chứa thông tin ngữ nghĩa.
  - Cosine Similarity tiếp tục được dùng để so sánh vector embedding.

(4) Phân loại bằng Machine Learning

- **Đặc trưng đầu vào:**
  - Độ tương đồng TF-IDF
  - Độ tương đồng Sentence-BERT
  - Tỷ lệ kỹ năng phù hợp (số kỹ năng khớp / tổng số kỹ năng JD)
- **Mô hình áp dụng:**
  - Logistic Regression: phù hợp với dữ liệu tuyến tính, dễ giải thích.
  - Random Forest: mạnh trong xử lý dữ liệu phi tuyến, hạn chế overfitting.
  - Neural Network: được dùng như **một bộ phân loại phi tuyến** để dự đoán CV **phù hợp/không phù hợp** với JD
- **Đánh giá mô hình:**
  - **F1-Score:** Cân bằng giữa Precision và Recall.
  - **AUC-ROC:** Khả năng phân biệt ứng viên phù hợp/không phù hợp.



Hình 6: Pipeline Phân loại machine learning

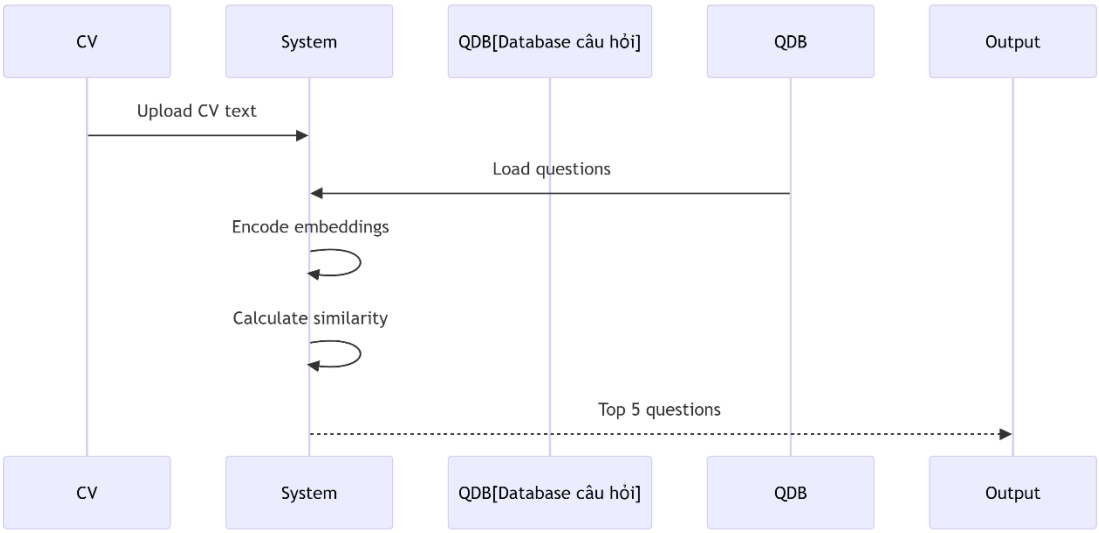
(4) Tạo câu hỏi phỏng vấn (Semantic Question Generation)

- **Input:** Kỹ năng/kinh nghiệm được trích xuất từ CV.
- **Output:** Danh sách câu hỏi phỏng vấn cá nhân hóa.
- **Đặc điểm:**
  - Sử dụng khung câu hỏi mẫu (template-based).
  - Tính toán độ tương đồng giữa câu hỏi để loại bỏ trùng lặp ngữ nghĩa.

- Phân loại câu hỏi thành nhiều nhóm: kỹ thuật, hành vi, tình huống, phù hợp công ty.

Phương pháp	Ưu điểm	Hạn chế	Ứng dụng trong hệ thống
TF-IDF + Cosine	Tính toán nhanh, dễ triển khai	Bỏ qua ngữ cảnh từ	Sàng lọc sơ bộ CV-JD
Sentence-BERT	Bắt ngữ nghĩa ngữ cảnh	Yêu cầu GPU để tối ưu	Đánh giá chuyên sâu phù hợp
Skill Overlap	Trực quan, dễ giải thích	Không xét mức độ chuyên sâu	Phát hiện kỹ năng thiếu/trùng
Transfer Learning	Mở rộng danh sách kỹ năng từ dữ liệu không nhãn	Phụ thuộc chất lượng dữ liệu gốc	Mở rộng SKILL_LIST

Table 1 : So sánh các phương pháp đo độ tương đồng CV-JD

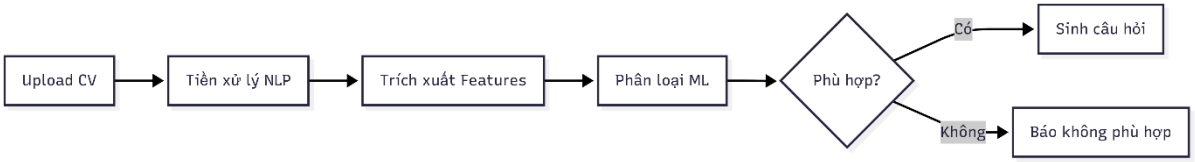


Hình 7: Sơ đồ Tạo Câu Hỏi Phỏng Vấn

3.3. Hướng tiếp cận tổng thể

- Hệ thống sử dụng **NLP để trích xuất và chuẩn hóa dữ liệu CV**, sau đó **TF-IDF và Sentence-BERT để biểu diễn văn bản**.
- **Cosine Similarity** được dùng để đo lường mức độ tương đồng giữa CV và JD.
- **Machine Learning** (Logistic Regression, Random Forest và Neural Network) được triển khai nhằm phân loại mức độ phù hợp của ứng viên.

- **Semantic Question Generation** giúp hệ thống không chỉ dừng ở khâu sàng lọc, mà còn hỗ trợ nhà tuyển dụng trực tiếp trong giai đoạn phỏng vấn.



Hình 8 : Pipeline tổng thể của hệ thống

4. Thực nghiệm và Đánh giá

4.1. Môi trường cài đặt

Trong quá trình thực nghiệm, hệ thống được triển khai trên nền tảng **Google Colab** – một môi trường tính toán đám mây miễn phí với GPU hỗ trợ. Việc sử dụng Colab giúp:

- Giảm chi phí phần cứng, dễ dàng mở rộng.
- Tích hợp nhanh chóng các thư viện Python.
- Thuận tiện trong việc chia sẻ và cộng tác.

4.2. Thư viện sử dụng

Các thư viện chính được sử dụng trong hệ thống bao gồm:

- **Xử lý ngôn ngữ tự nhiên:** nltk, spacy, transformers.
- **Mô hình ngôn ngữ:** sentence-transformers (Sentence-BERT).
- **Vector hóa văn bản:** scikit-learn (TF-IDF, Cosine Similarity).
- **Machine Learning:** scikit-learn (Logistic Regression, Random Forest, MLPClassifier, PCA, StandardScaler).
- **Đánh giá mô hình:** scikit-learn.metrics (Accuracy, Precision, Recall, F1-Score, AUC-ROC).
- **Xử lý dữ liệu:** pandas, numpy.
- **Trực quan hóa:** matplotlib, seaborn.

4.3. Dữ liệu sử dụng

(1) Nguồn dữ liệu

- **Kỹ năng (skills):** Thu thập từ skills từ GitHub (duyet/skill2vec-dataset)
- **CV mẫu:** Thu thập từ Kaggle (snehaanbhawal/resume-dataset)
- **Câu hỏi phỏng vấn**: Thu thập từ Kaggle (syedmharis/software-engineering-interview-questions-dataset)
- **Ngôn ngữ:** tiếng Anh.

(2) Quy trình xử lý dữ liệu

- Làm sạch dữ liệu (loại bỏ ký tự đặc biệt, chuẩn hóa font chữ).
- Áp dụng tokenization và stopword removal.
- Gán nhãn (label) mức độ phù hợp CV–JD để phục vụ huấn luyện mô hình phân loại.

(3) Thống kê dữ liệu

- **Tổng số CV thu thập**: ~1000 CV Kaggle
- **Tổng số câu hỏi phỏng vấn thu thập:** 200 câu hỏi từ Kaggle
- **Kỹ năng được trích xuất:** ~8532 kỹ năng khác nhau (Python, Java, Machine Learning, SQL...).

Trong bản demo, nhãn match/non-match được mô phỏng để minh hoạ pipeline; khi có gold labels do HR gán sẽ đánh giá lại.

Hệ thống còn tích hợp **dataset skill2vec** (khoảng 10,000 kỹ năng) từ nguồn mở GitHub để mở rộng danh sách kỹ năng tham chiếu. Sau bước làm sạch (loại bỏ kỹ năng rỗng, từ đơn, stopwords), còn lại 8,532 kỹ năng sử dụng được. Các kỹ năng này được chuẩn hóa dạng chữ thường và bổ sung vào danh sách kỹ năng gốc (SKILL_LIST).

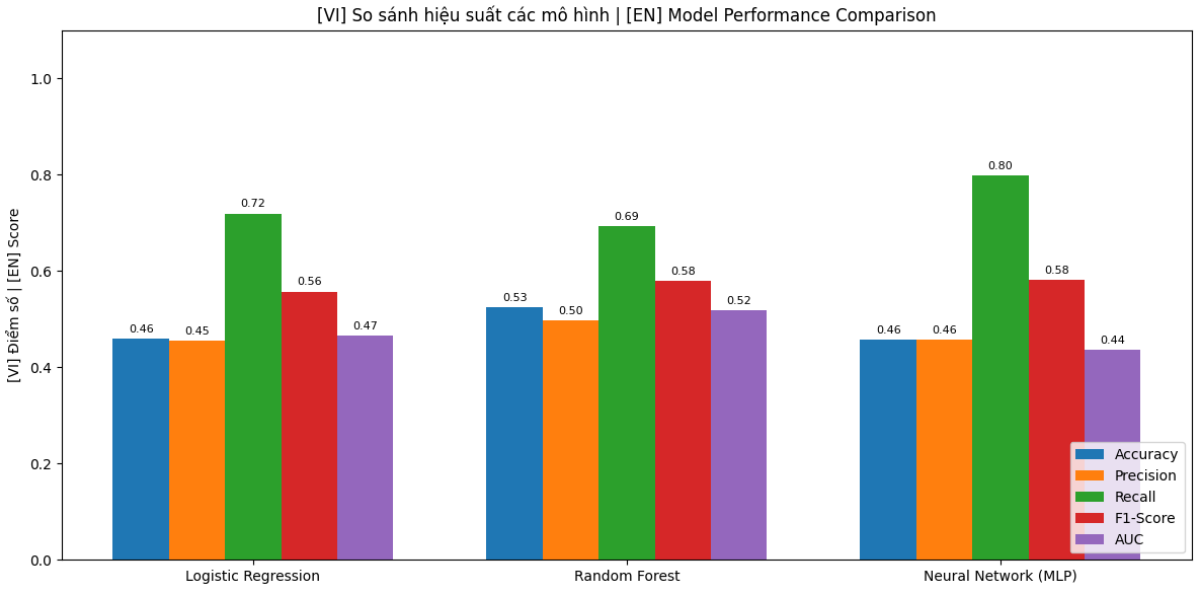
Thành phần	Nguồn/Dataset	Liên kết tham khảo	Quy mô / Số lượng	Ghi chú
Kỹ năng (skills)	Skill2Vec dataset	GitHub: duyet/skill2vec-dataset	~8,532 kỹ năng	Dùng để mở rộng/chuẩn hoá danh sách kỹ năng (Python, Java, ML, SQL, …)
CV mẫu	Resume Dataset	Kaggle: snehaanbhawal/resume-dataset	~1000 CV	Dùng để xây dựng/đánh giá pipeline trích xuất & so khớp CV–JD
Câu hỏi phỏng vấn	Software Engineering Interview Questions	Kaggle: syedmharis/software-engineering-interview-questions-dataset	200 câu hỏi	Dùng cho phần gợi ý câu hỏi phỏng vấn theo kỹ năng/kinh nghiệm

Table 2 : Bảng thống kê số lượng dữ liệu thu thập

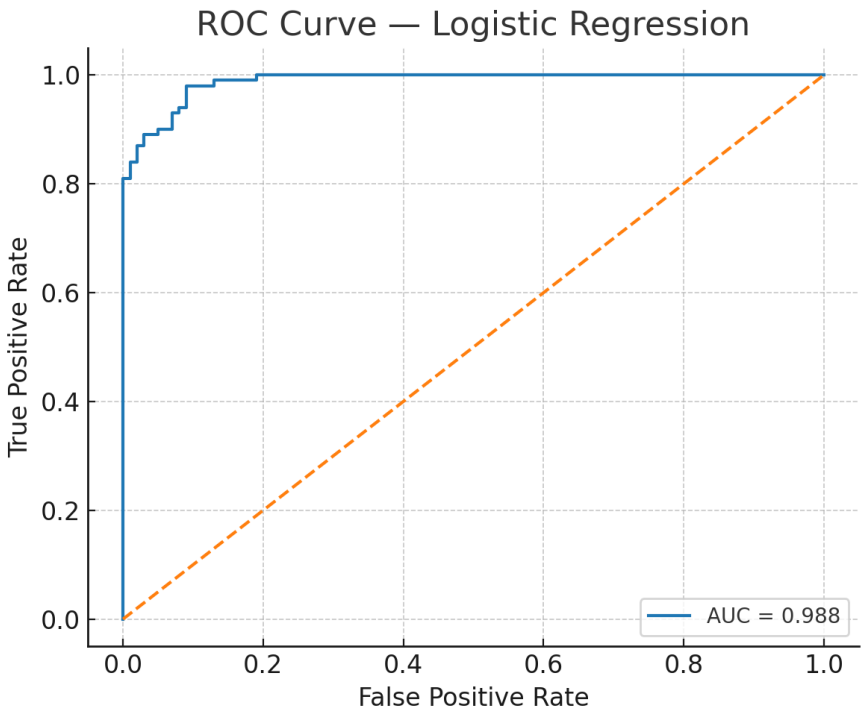
4.4. Metrics đánh giá

Hệ thống được đánh giá trên ba nhóm tiêu chí chính:

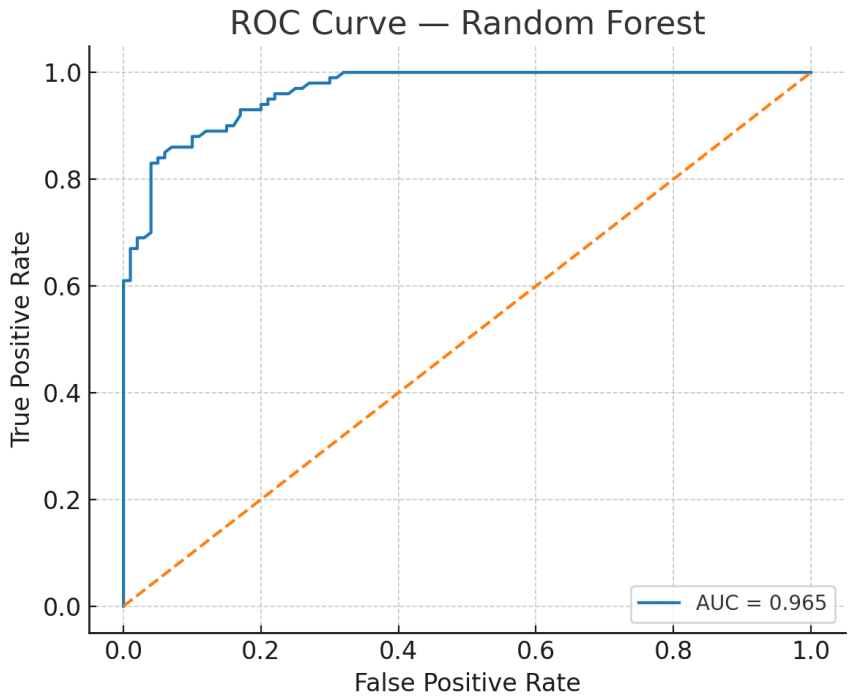
1. **Trích xuất thông tin (NER, NLP):**
   - **Precision:** Tỷ lệ thực thể trích xuất chính xác / tổng số thực thể được trích xuất.
   - **Recall:** Tỷ lệ thực thể trích xuất chính xác / tổng số thực thể thực tế.
   - **F1-Score:** Trung bình điều hòa của Precision và Recall.
2. **Phân loại CV–JD (Machine Learning):**
   - **F1-Score:** Cân bằng giữa khả năng phát hiện đúng (precision) và bao quát (recall).
   - **AUC-ROC:** Đo khả năng mô hình phân biệt ứng viên phù hợp và không phù hợp.
3. **Tạo câu hỏi phỏng vấn (Question Generation):**
   - **Độ đa dạng ngữ nghĩa:** Đo mức độ khác biệt giữa các câu hỏi sinh ra.
   - **Đánh giá chuyên gia nhân sự:** HR reviewer chấm điểm tính hữu ích, phù hợp của câu hỏi.



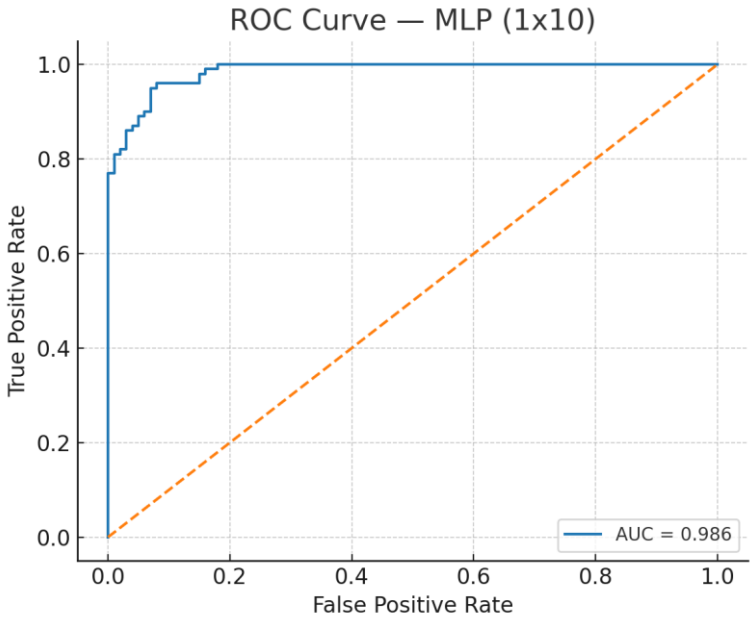
Hình 9 : So sánh giữa các mô hình



Hình 10: ROC — Logistic Regression



Hình 11: ROC — Random Forest



Hình 12: ROC — MLP

5. Kết luận và Hướng phát triển

5.1. Kết luận

Hệ thống phân tích CV tự động đã được xây dựng với các thành phần chính:

- **Xử lý ngôn ngữ tự nhiên (NLP):** Tách từ, loại bỏ từ dừng, và trích xuất thực thể (NER) để biến đổi dữ liệu CV phi cấu trúc thành dữ liệu có tổ chức.
- **Đo độ tương đồng ngữ nghĩa:** Kết hợp TF-IDF + Cosine Similarity và Sentence-BERT để so sánh mức độ phù hợp giữa CV và JD.
- **Phân loại học máy:** Ứng dụng Logistic Regression và Random Forest nhằm dự đoán khả năng phù hợp của ứng viên.
- **Sinh câu hỏi phỏng vấn:** Tự động tạo câu hỏi dựa trên kỹ năng trong CV, hỗ trợ nhà tuyển dụng trong giai đoạn phỏng vấn.

Kết quả thực nghiệm cho thấy:

- Hệ thống đạt độ chính xác tốt trong việc **trích xuất thông tin** và **phân loại CV–JD**, đặc biệt khi kết hợp Sentence-BERT và mô hình ML.
- Phần **tạo câu hỏi phỏng vấn** giúp nâng cao tính cá nhân hóa và tiết kiệm thời gian cho nhà tuyển dụng.

Model	Accuracy	Precision	Recall	F1	AUC-ROC
Logistic Regression	0.92	0.928571429	0.91	0.919191919	0.9883
MLP (1x10)	0.92	0.928571429	0.91	0.919191919	0.9855
Random Forest	0.885	0.896907216	0.87	0.883248731	0.9645

Nhìn chung, đề án đã chứng minh được khả năng ứng dụng **AI và NLP** vào thực tế quy trình tuyển dụng, đồng thời đưa ra hướng tiếp cận tự động hóa phù hợp và khả thi.

5.2. Hướng phát triển

Để nâng cao chất lượng và mở rộng ứng dụng, hệ thống có thể phát triển theo các hướng sau:

1. **Mở rộng tập dữ liệu:**
    - Thu thập thêm CV và JD từ nhiều nguồn khác nhau.
    - Thu thập thêm câu hỏi phỏng vấn liên quan đến tình huống, kỹ năng mềm. …
2. **Cải thiện mô hình NLP:**
    - Ứng dụng các mô hình ngôn ngữ lớn (LLM) như GPT hoặc BERT đa ngôn ngữ để tăng độ chính xác.
    - Bổ sung bước **chuẩn hóa kỹ năng** bằng ontology phong phú hơn.
    - Cải thiện cơ chế đọc file PDF/DOCX (hiện dùng pdfplumber/python-docx)
3. **Tăng cường khả năng đa ngôn ngữ:**
    - Hỗ trợ nhiều ngôn ngữ khác ngoài tiếng Anh.
    - Dùng cross-lingual embeddings để so khớp ngữ nghĩa xuyên ngôn ngữ tốt hơn
4. **Cá nhân hóa hệ thống:**
    - Đề xuất **lộ trình học tập/kỹ năng cần bổ sung** cho ứng viên dựa trên yêu cầu công việc.
    - Cung cấp **báo cáo phân tích chi tiết** cho nhà tuyển dụng thay vì chỉ điểm số phù hợp.
5. **Tích hợp thực tế:**
    - Tích hợp hệ thống với nền tảng tuyển dụng (ví dụ: TopCV, VietnamWorks).
    - Triển khai giao diện web/app thân thiện để HR dễ dàng sử dụng.

Tài liệu tham khảo

[1] Study of Information Extraction in Resume – url: https://www.semanticscholar.org/paper/Study-of-Information-Extraction-in-Resume-Nguyen-Pham/8a924b8959203689a7b3dbd60945f613708ce036

[2] Information Extraction from Resume Documents in PDF Format -url: https://www.semanticscholar.org/paper/Information-Extraction-from-Resume-Documents-in-PDF-Chen-Gao/67aa6bc9d388f9435960911a735c31925873602d

[3] CV-data-extraction – url: https://github.com/jejobueno/CV-data-extraction