

الجمهورية الجزائرية الديمقراطية الشعبية
RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
وزارة التعليم العالي والبحث العلمي
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE
SCIENTIFIQUE

المدرسة الوطنية المتعددة التقنيات
ECOLE NATIONALE POLYTECHNIQUE



المدرسة الوطنية المتعددة التقنيات
Ecole Nationale Polytechnique

Département Génie Industriel

End-of-study project dissertation for obtaining the degree of

State Engineer in Industrial Engineering

Specialty: Data Science and Artificial Intelligence

**Multi-Axis Agile Learning Engine for Matching:
A Framework for Transparent and Flexible Candidate-Job Matching**

Amine MAALEM

Supervised by **Dr. FOURAR LAIDI Hakim (ENP)**

Presented and publicly defended on (07/07/2025)

Examination Committee:

Chair:	Dr. ARKI Oussama	MCA
Supervisor:	Dr. FOURAR LAIDI Hakim	MCA
Examiner:	Dr. BARECHE Imene	MAB

ENP 2025

الجمهورية الجزائرية الديمقراطية الشعبية
RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
وزارة التعليم العالي والبحث العلمي
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE
SCIENTIFIQUE

المدرسة الوطنية المتعددة التقنيات
ECOLE NATIONALE POLYTECHNIQUE



المدرسة الوطنية المتعددة التقنيات
Ecole Nationale Polytechnique

Département Génie Industriel

End-of-study project dissertation for obtaining the degree of

State Engineer in Industrial Engineering

Specialty: Data Science and Artificial Intelligence

**Multi-Axis Agile Learning Engine for Matching:
A Framework for Transparent and Flexible Candidate-Job Matching**

Amine MAALEM

Supervised by **Dr. FOURAR LAIDI Hakim (ENP)**

Presented and publicly defended on (07/07/2025)

Examination Committee:

Chair:	Dr. ARKI Oussama	MCA
Supervisor:	Dr. FOURAR LAIDI Hakim	MCA
Examiner:	Dr. BARECHE Imene	MAB

ENP 2025

الجمهورية الجزائرية الديمقراطية الشعبية
RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
وزارة التعليم العالي والبحث العلمي
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE
SCIENTIFIQUE

المدرسة الوطنية المتعددة التقنيات
ECOLE NATIONALE POLYTECHNIQUE



المدرسة الوطنية المتعددة التقنيات
Ecole Nationale Polytechnique

Département de Génie Industriel

Mémoire de fin d'études en vue de l'obtention du diplôme d'

Ingénieur d'État en Génie Industriel

Spécialité : Data Science et Intelligence Artificielle

Multi-Axis Agile Learning Engine for Matching:
Un Framework pour une Adéquation Candidat-Poste Transparente et
Modulable

Amine MAALEM

Encadré par : **Dr. FOURAR LAIDI Hakim (ENP)**

Présenté et soutenu publiquement le (07/07/2025)

Jury d'examen :

Président :	Dr. ARKI Oussama	MCA
Encadrant :	Dr. FOURAR LAIDI Hakim	MCA
Examineur :	Dr. BARECHE Imene	MAB

ENP 2025

Acknowledgements

My work begins and ends with gratitude to God,

الْحَمْدُ لِلَّهِ رَبِّ الْعَالَمِينَ.

I am humbled by His blessings, as I know that nothing is possible without His help.

My deepest appreciation is reserved for my beloved parents. Their boundless patience, unconditional love, and steadfast belief in my potential have been the driving force behind all my efforts. I am eternally indebted to them for their sacrifices and support.

My sincere thanks go to the National Polytechnic School (ENP) for being my academic home and for giving me the opportunity to pursue this exciting research.

Finally, a big thank you to the members of my jury. To my supervisor, Dr. Fourar Laidi Hakim, and to Dr. Samia BELDJOUDE, thank you for your mentorship and guidance. To Dr. Arki Oussama and Dr. Bareche Imene, thank you for taking the time to read and evaluate my work. Your presence and feedback are greatly appreciated.

ملخص

يواجه التوظيف الحديث تحدياً حاسماً، حيث تعاني محركات البحث الدلالية المتقدمة، على الرغم من قوتها، من كونها "صناديق سوداء" غامضة تفتقر إلى القابلية للتفسير والخصوصية التي يطلبها مسؤولو التوظيف. يعالج هذا المشروع ذلك التحدي من خلال مسار مزدوج. أولاً، يفصل المشروع تطوير محرك أساسي صناعي عالي الأداء، وهو نظام تم تسليمه للشركة المضيفة ويستخدم واجهة جوجل Gemini البرمجية ومكتبة FAISS لتلبية جميع متطلبات حلول المطابقة الدلالية الحديثة. ثانياً، وبناءً على الدروس المستفادة من القيود المتأصلة في النظام الأساسي، يقدم هذا البحث إطاراً بحثياً متقدماً. يحمل هذا الإطار اسم MAALEM، وهو اختصار لـ (Multi-Axis Agile Learning Engine for Matching). هذا الاسم، الذي يُترجم إلى "معلم" أو "خبير" باللغة العربية وهو أيضاً لقب المؤلف، يعكس الطموح الأساسي للمشروع: تحقيق فهم متقن ودقيق لمسألة التوافق بين المرشح والوظيفة. الابتكار الجوهرى لإطار MAALEM يكمن في بنيته المفككة، التي تتجاوز مقياس التشابه الواحد لتقييم المرشحين عبر أبعاد متعددة، شفافة، وقابلة للتفسير. يعطي هذا التصميم الأولوية للمنطق المبرر وتحكم المستخدم، محولاً الذكاء الاصطناعي من مجرد مرشح غامض إلى شريك حقيقي لدعم القرار. للتحقق من صحة هذا النموذج الجديد، تم إنشاء مجموعة شاملة من ثلاثة معايير تقييم مبتكرة، بما في ذلك اختبار خصومي مصمم لسر ذكاء النموذج ومرونته. تظهر النتائج التجريبية أنه بينما يعد المحرك الصناعي أداة قوية، فإن إطار MAALEM يتفوق عليه بشكل كبير وعلى جميع النماذج القياسية الأخرى، خاصة في الظروف الخصومية. لقد أثبتت فعالية فريدة في رفض المرشحين الخادعين، وفي نفس الوقت، تحديد نسبة عالية من المرشحين من نوع "الجواهر الخفية" غير التقليديين. لذلك، يمثل MAALEM رؤية جديدة للذكاء الاصطناعي في مجال التوظيف رؤية ليست فقط أكثر دقة وقوة، بل هي في جوهرها أكثر جدارة بالثقة ومتوافقة استراتيجياً مع احتياجات مستخدمي الخبراء.

الكلمات المفتاحية: أنظمة التوصية، التوافق بين الشخص والوظيفة، الذكاء الاصطناعي القابل للتفسير، (XAI) البحث الدلالي، معالجة اللغة الطبيعية، (NLP) تكنولوجيا التوظيف، قواعد بيانات المتجهات، التفاعل بين الإنسان والحاسوب.

Résumé

Le recrutement moderne fait face à un défi critique : les moteurs de recherche sémantiques, bien que puissants, fonctionnent comme des "boîtes noires" opaques qui manquent de l'explicabilité et de la spécificité requises par les recruteurs professionnels. Ce projet aborde ce défi via une approche à double voie. Premièrement, il détaille le développement d'un **Moteur de Référence Industriel** performant, un système livré à l'entreprise d'accueil qui utilise l'API Gemini de Google et la bibliothèque FAISS pour répondre à toutes les exigences d'une solution de matching sémantique moderne.

Deuxièmement, en s'appuyant sur les enseignements tirés des limites inhérentes de ce système de référence, cette thèse introduit le framework de recherche avancé **MAALEM**, un acronyme pour **M**ulti-**A**xis **A**gile **L**earning **E**ngine for **M**atching. Ce nom, qui se traduit par "maître" ou "expert" en arabe et qui est également le nom de l'auteur, reflète l'ambition principale du projet : atteindre une compréhension experte et nuancée de l'adéquation candidat-poste. L'innovation principale du framework **MAALEM** réside dans son architecture déconstruite, qui dépasse le simple score de similarité pour évaluer les candidats selon de multiples dimensions indépendantes, transparentes et interprétables. Cette conception priorise le raisonnement justifié et le contrôle par l'utilisateur, transformant l'IA d'un filtre opaque en un véritable partenaire d'aide à la décision.

Pour valider ce nouveau paradigme, une suite complète de trois benchmarks novateurs a été créée, incluant un test contradictoire (adversarial) conçu pour sonder l'intelligence et la résilience du modèle. Les résultats empiriques démontrent que, bien que le Moteur Industriel soit un outil puissant, le framework **MAALEM** le surpasse de manière significative, ainsi que tous les autres modèles standards, en particulier dans des conditions contradictoires. Il s'est avéré particulièrement efficace pour rejeter les candidats trompeurs tout en identifiant simultanément les profils de type "pépité cachée".

MAALEM représente ainsi une nouvelle vision pour l'IA dans le recrutement — une vision non seulement plus précise et robuste, mais aussi fondamentalement plus fiable et stratégiquement alignée sur les besoins de ses utilisateurs experts.

Mots-clés : Systèmes de Recommandation, Adéquation Personne-Poste, IA Explicable (XAI), Recherche Sémantique, Traitement Automatique du Langage (TAL), Technologie du Recrutement, Bases de Données Vectorielles, Interaction Homme-Machine.

Abstract

Modern recruitment faces a critical challenge: state-of-the-art semantic search engines, while powerful, operate as opaque "black boxes" that lack the explainability and specificity required by professional recruiters. This project addresses this challenge through a dual-track approach. First, it details the development of a high-performance **Industrial Baseline Engine**, a system delivered to the host company that uses Google Gemini and the FAISS library to fulfill all requirements of a modern semantic matching solution.

Second, using the insights gained from the baseline's inherent limitations, this thesis introduces an advanced research framework. This framework is named **MAALEM**, an acronym for **M**ulti-**A**xis **A**gile **L**earning **E**ngine for **M**atching. The name, which translates to "master" or "expert" in Arabic and is also the surname of the author, reflects the project's core ambition: to achieve a masterful and nuanced understanding of person-job fit. The **MAALEM** framework's core innovation is its deconstructed architecture, which moves beyond a single similarity score to evaluate candidates along multiple, transparent, and interpretable dimensions. This design prioritizes justified reasoning and user control, transforming the AI from an opaque filter into a genuine decision-support partner.

To validate this new paradigm, a comprehensive suite of three novel benchmarks was created, including an adversarial test designed to probe model intelligence and resilience. Empirical results demonstrate that while the Industrial Baseline is a powerful tool, the **MAALEM** framework significantly outperforms it and all other standard models, especially under adversarial conditions. It proved uniquely effective at rejecting deceptive candidates while simultaneously identifying non-obvious "hidden gem" profiles.

MAALEM therefore represents a new vision for recruitment AI one that is not only more accurate and robust but is also fundamentally more trustworthy and strategically aligned with the needs of its expert users.

Keywords: Recommender Systems, Person-Job Fit, Explainable AI (XAI), Semantic Search, Natural Language Processing (NLP), Recruitment Technology, Vector Databases, Human-Computer Interaction.

Contents

List of Figures

List of Tables

List of Acronyms and Abbreviations

General Introduction	12
0.1 The Modern Recruitment Challenge	12
0.2 The Dominant Paradigm: The Semantic Search Engine	12
0.2.1 Architecture of an Industrial-Grade Semantic Engine	12
0.3 The Modern Recruitment Landscape	12
0.4 The Dominant Paradigm: Semantic Search and its "Black Box" Problem	13
0.5 A Dual Mandate: Fulfilling the Industrial Mission	13
0.5.1 Intelligent Parsing with Large Language Models	13
0.6 The 'Black Box' Limitation and a New Research Direction	14
0.7 The Research Contribution: The MAALEM Framework	14
0.8 Project Objectives	14
0.8.1 Vector Embeddings for Semantic Representation	14
0.8.2 High-Speed Retrieval with Vector Databases	14
0.8.3 The Integrated Industrial System	15
0.9 Advancing the State-of-the-Art: The MAALEM Philosophy	15
0.10 Core Objectives and Report Structure	15
1 From Manual Craft to Automated Systems: A Review of Person-Job Matching	16
1.1 The Recruiter's Craft: The Manual Gold Standard	16
1.1.1 The Scalability Crisis	17
1.2 The State of the Art in Automated Matching Systems	17
1.2.1 Paradigm 1: Lexical Matching - The Brittle Foundation	17
1.2.2 Paradigm 2: Semantic Matching - The Powerful Black Box	17
1.3 Synthesis and Justification for a New Approach	18
2 Design and Methodology	19
2.1 Approach 1: The Industrial Baseline - A Semantic Search Engine	19
2.1.1 Architectural Overview and Technology Stack	19
2.1.2 Component 1: High-Fidelity Parsing and Information Extraction	20
2.1.3 Component 2: Semantic Vector Embedding	23
2.1.4 Component 3: High-Performance Vector Search and Retrieval	25
2.1.5 System Workflow and Command-Line Interface	26
2.2 Approach 2: The MAALEM Framework - A New Philosophy for Matching	27
2.2.1 Core Philosophy: Beyond the Black Box	28
2.2.2 The Three Guiding Principles of MAALEM	28
2.2.3 Conceptual Workflow: From Recommendation to Dialogue	29
2.2.4 Chapter Conclusion	29

3	Evaluation Framework	30
3.1	The Benchmarking Challenge in Person-Job Fit	30
3.2	The Evaluation Framework: A Suite of Three Benchmarks	30
3.2.1	Benchmark 1: "Binary Noisy Rank" — Testing for Real-World Efficacy	31
3.2.2	Benchmark 2: "Trapped Binary Noise Rank" — Testing for Adversarial Robustness	32
3.2.3	Benchmark 3: "Absolute Rank" — Testing for Ranking Precision	33
3.3	Baseline Models for Comparison	34
3.3.1	The Primary Baseline: Industrial Engine (Gemini-SBERT)	34
3.3.2	Secondary Baselines: Lexical Models	34
3.3.3	Secondary Baselines: Generic Semantic Models	34
4	Experiments and Results Analysis	35
4.1	Experimental Setup	35
4.1.1	Hardware and Software Stack	35
4.1.2	Evaluation Datasets and Benchmarks	36
4.1.3	Benchmark 2: Performance under Adversarial Conditions	37
4.1.4	Benchmark 3: Performance in Ranking Precision	39
4.1.5	Synthesis of Baseline Performance	40
4.2	Part II: Comparative Analysis of the MAALEM Framework	41
4.2.1	Benchmark 1: Dominance in a Realistic Filtering Environment	42
4.2.2	Benchmark 2: A New Class of Adversarial Robustness	44
4.2.3	Benchmark 3: Unmatched Ranking Precision	45
4.3	Overall Synthesis and Qualitative Value	47
4.3.1	Final Quantitative Synthesis: The Overall Capability Profile	47
4.3.2	The Unquantifiable Value: A Case Study in Explainability	48
5	Discussion	49
5.1	Interpreting the Performance of the Baseline Models	49
5.1.1	The Vindicated Power of Modern Semantic Search	49
5.1.2	Exposing the "Black Box" Vulnerabilities	50
5.2	The Vindication of the MAALEM Framework	50
5.2.1	An Architecture Designed for Robustness	51
5.2.2	The Unquantifiable Value of Explainability and Control	51
5.3	Critical Reflection: Scope and Limitations	52
5.3.1	The Industrial Baseline: A Pragmatic and Fit-for-Purpose Deliverable	52
5.3.2	The MAALEM Framework: The Trade-off of Specialization	52
6	General Conclusion	54
6.1	Summary of Core Contributions	54
6.2	Final Project Verdict	55
A	Algorithmic Workflow of the Industrial Baseline Engine	56
A.1	The 'index' Command Workflow	56
A.2	The 'match' Command Workflow	56
A.3	Core Helper Function: Document Processing Pipeline	57
	Bibliography	58

List of Figures

2.1	The Information Extraction Pipeline. Unstructured text from a source document is processed by the Gemini API to produce a structured JSON profile.	21
2.2	The Semantic Matching Workflow. Structured profiles are converted into vectors. The job offer's vector is used to query the FAISS database of CV vectors, producing a ranked list based on similarity.	24
4.1	Benchmark 1 Performance Heatmap (Baselines Only). The strong performance of the Industrial Baseline is evident across key filtering metrics.	37
4.2	Benchmark 1: Detailed Performance Comparison (Baselines Only). This chart highlights the competitive performance of the Industrial Baseline, especially on the MAP score. . . .	37
4.3	Benchmark 2: Performance on Key Robustness Metrics (Baselines Only). The chart shows the general inability of standard models to achieve a high positive Robustness Score. . . .	38
4.4	Benchmark 2 Performance Heatmap (Baselines Only). The mixed results across all models indicate a universal struggle with adversarial conditions.	39
4.5	Benchmark 3: Ranking Performance Comparison (Baselines Only).	40
4.6	Benchmark 3 Performance Heatmap (Baselines Only).	40
4.7	Overall Model Capability Profile (Baselines Only). The chart plots normalized scores across key dimensions, establishing a performance ceiling for standard models.	41
4.8	Benchmark 1: MAP Score Comparison (All Models). MAALEM demonstrates both the highest average score and the most consistent performance.	43
4.9	Benchmark 1 Performance Heatmap (All Models). The consistently dark row for MAALEM indicates its superior performance across a wide range of metrics.	43
4.10	Benchmark 1: Detailed Performance Comparison (All Models). The chart highlights MAALEM's consistent lead across all key metrics.	44
4.11	Benchmark 2: Performance on Key Robustness Metrics (All Models). MAALEM's commanding lead demonstrates its unique ability to both find hidden gems and avoid deceptive traps.	45
4.12	Benchmark 2 Performance Heatmap (All Models). MAALEM's row (outlined in red) shows consistently high scores, indicating superior robustness across all metrics, while other models show significant weaknesses.	45
4.13	Benchmark 3: Ranking Performance Comparison (All Models).	46
4.14	Benchmark 3 Performance Heatmap (All Models).	46
4.15	Overall Model Capability Profile (All Models). The chart plots normalized scores across key dimensions, where a larger area indicates superior all-around performance.	47

List of Tables

2.1	Performance of the custom-trained spaCy NER model ('cv_parser_model_lg') on key entities.	22
3.1	Overview of the Custom Benchmark Suite.	31
4.1	Benchmark 1 Leaderboard (Baselines Only): Filtering Performance. Scores are presented as Mean $\pm$ Standard Deviation.	36
4.2	Benchmark 2 Leaderboard (Baselines Only): Adversarial Robustness. Scores are presented as Mean $\pm$ Standard Deviation.	38
4.3	Benchmark 3 Leaderboard (Baselines Only): Ranking Precision. Scores are presented as Mean $\pm$ Standard Deviation.	39
4.4	Benchmark 1 Full Leaderboard: Filtering Performance. Scores are presented as Mean $\pm$ Standard Deviation.	42
4.5	Benchmark 2 Full Leaderboard: Adversarial Robustness. Scores are presented as Mean $\pm$ Standard Deviation.	44
4.6	Benchmark 3 Full Leaderboard: Ranking Precision. Scores are presented as Mean $\pm$ Standard Deviation.	46

List of Acronyms and Abbreviations

AI	Artificial Intelligence
API	Application Programming Interface
ATS	Applicant Tracking System
BM25	Best Match 25 (Okapi BM25)
BoW	Bag-of-Words
CV	Curriculum Vitae
HITL	Human-in-the-Loop
HR	Human Resources
IT	Information Technology
JSON	JavaScript Object Notation
KG	Knowledge Graph
LLM	Large Language Model
MAALEM	Multi-Axis Agile Learning Engine for Matching
MAP	Mean Average Precision
ML	Machine Learning
MoE	Mixture-of-Experts
NDCG	Normalized Discounted Cumulative Gain
NER	Named Entity Recognition
NLP	Natural Language Processing
P-J Fit	Person-Job Fit
P-O Fit	Person-Organization Fit
PFE	Projet de Fin d'Études (Final Year Project)
S-BERT	Sentence-BERT (Sentence-Bidirectional Encoder Representations from Transformers)
SDLC	Software Development Life Cycle
TF-IDF	Term Frequency-Inverse Document Frequency
UI	User Interface
UX	User Experience
XAI	Explainable Artificial Intelligence

General Introduction

In the modern economy, the axiom that an organization's success is determined by the quality of its people has never been more resonant. The challenge, as articulated in the seminal recruitment guide *Who*[1], is not merely knowing *what* to do, but "finding the *who* to do it." This quest to identify and attract premier talent has evolved into a highly competitive, data-intensive discipline where the stakes are incredibly high. This introduction will establish the complex landscape of modern recruitment, detail the project's dual mandate of fulfilling a specific industrial mission while pursuing advanced research, and thereby define the objectives and scope of this report.

0.1 The Modern Recruitment Challenge

The process of finding the right candidate is fraught with challenges, amplified by profound and often conflicting forces in technology and the labor market. On one hand, the widespread adoption of online job portals and Applicant Tracking Systems (ATS) has created an unprecedented deluge of applications for every open position. This environment, rich in data but poor in insight, overwhelms human recruiters, who, according to industry studies, spend an average of just **six to eight seconds** on the initial screening of a single resume.

On the other hand, compounding this volume problem is a pressing quality problem: the "skills gap." Recent surveys indicate that as many as 87% of companies are aware that they either already have a skills gap or will face one within a few years. This scarcity of qualified talent, coupled with the immense financial and cultural costs of a "bad hire," makes the need for effective, technologically-driven solutions not just a convenience, but a strategic imperative.

0.2 The Dominant Paradigm: The Semantic Search Engine

In response to these challenges, the industry has largely converged on a powerful technological paradigm: the semantic search engine. This approach moves beyond the brittle, keyword-based logic of older lexical methods (e.g., TF-IDF[2]) which were easily gamed and lacked any true understanding of language. Instead, semantic engines aim to comprehend the conceptual meaning behind the text in a CV or job offer.

0.2.1 Architecture of an Industrial-Grade Semantic Engine

The primary mission of this project, as outlined by the host company, was to build and implement a complete, end-to-end AI solution based on this modern semantic architecture. The delivered system, which successfully fulfilled 100

0.3 The Modern Recruitment Landscape

The process of finding the right candidate for the right role is fraught with challenges, amplified by profound and often conflicting forces in technology and the labor market. The widespread adoption of online job portals and Applicant Tracking Systems (ATS) has created an unprecedented deluge of applications for every open position. While this widens the potential talent pool, it simultaneously overwhelms human recruiters, who, according to industry studies, spend an average of just **six to eight seconds** on the initial screening of a single resume.

Compounding this volume problem is a pressing quality problem: the "skills gap." Recent surveys indicate that as many as 87% of companies are aware that they either already have a skills gap or will

face one within a few years. This scarcity of qualified talent makes the accurate identification of suitable candidates profoundly difficult, turning recruitment into a high-risk, high-reward endeavor where the cost of failure is immense.

0.4 The Dominant Paradigm: Semantic Search and its "Black Box" Problem

In response to these challenges, automated matching systems have become ubiquitous. Early systems, based on lexical matching methods like TF-IDF[2], were transparent but brittle, frequently failing to recognize conceptual similarities between a job offer and a candidate's CV.

The modern technological paradigm has since shifted decisively towards powerful **semantic search engines**. These systems leverage deep learning models, such as those from the BERT family[3], to convert entire documents into numerical vectors (embeddings). These vectors represent the document's meaning in a high-dimensional space, and a match is made by finding the "closest" vectors using distance metrics like cosine similarity. This allows a system to understand that a "Java developer" and a "backend software engineer using Java" are conceptually related, overcoming the rigidity of lexical methods.

While incredibly powerful, this approach introduces a new, more insidious problem: it operates as an opaque **"black box."** The system produces a single similarity score (e.g., 0.87) but offers no justification for its reasoning. This "just trust me" model is fundamentally at odds with the high-stakes, decision-driven environment of professional recruitment, where every choice must be defensible, transparent, and auditable. This very dilemma forms the central motivation for this project.

0.5 A Dual Mandate: Fulfilling the Industrial Mission

This project was executed with a dual objective: to first build and deliver a high-performance semantic engine that fulfilled a specific industrial mandate, and then, using the insights gained, to research and design a superior framework that addresses the "black box" problem.

The primary mission, as outlined by the host company, was to build and implement a complete, end-to-end AI solution for their recruitment platform. This mission was 100% fulfilled through the development of a robust system architecture, detailed below.

0.5.1 Intelligent Parsing with Large Language Models

The first requirement was the development of an intelligent CV parsing system. Traditional parsers often struggle with the vast diversity of CV formats, layouts, and languages. To solve this, the industrial system integrates directly with a state-of-the-art Large Language Model (LLM), specifically Google's Gemini[4]. A carefully-of-the-art components:

- 1. Intelligent Parsing with Large Language Models (LLMs):** The foundational layer of the engine is data extraction. For this, the system leverages the power of a state-of-the-art LLM, Google Gemini[4]. Through sophisticated prompt engineering, the Gemini API is tasked with parsing raw, unstructured text from documents (PDF, TXT) and transforming it into a structured JSON object. This process reliably extracts key entities such as skills, educational background, years of experience, and past job roles, forming a high-fidelity digital representation of the candidate or job offer.
- 2. Semantic Embedding with Transformer Models:** Once the data is structured, the next step is to capture its meaning. The system employs a powerful model from the **sentence-transformers** library[3]. This model takes the structured text, creates a composite string, and encodes it into a dense numerical vector (an embedding). This vector represents the document's position in a high-dimensional semantic space, where documents with similar meanings are located close to one another. This is the core of the engine's ability to match conceptually similar, but lexically different, documents.
- 3. High-Performance Search with FAISS:** A database of thousands or millions of vectors cannot be searched efficiently using brute-force methods. To solve this, the engine is powered by **FAISS (Facebook AI Similarity Search)**[5], a world-class library for efficient similarity search. All CV embeddings are stored in a FAISS index. When a job offer is processed, its resulting vector is used as a query to search the index. FAISS uses advanced Approximate Nearest Neighbor (ANN)

algorithms to find the top-k most similar CV vectors with millisecond latency, making the system highly scalable and production ready.

This delivered system represents a robust, modern, and effective solution that directly addresses the industrial need for a high-performance candidate matching tool.

0.6 The 'Black Box' Limitation and a New Research Direction

Despite its power and efficiency, the semantic search engine architecture has a fundamental, inherent limitation: it operates as a **"black box."** The system's final output is a single, opaque similarity score. A recruiter is told that Candidate A is an 87% match, but is given no actionable insight into *why*. Is it due to their skills, their education, or their years of experience? The system cannot answer. This "just trust me" approach is fundamentally misaligned with the needs of a high-stakes, decision-driven profession.

This critical insight that efficacy without explainability is insufficient became the primary catalyst for the second phase of this project: a research-oriented deep dive aimed at creating a new, superior paradigm for recruitment AI.

0.7 The Research Contribution: The MAALEM Framework

The limitations of the black box model led to the conceptualization of the **MAALEM** (Multi-Axis Agile Learning Engine for Matching) framework. The core philosophy of MAALEM is a radical departure from single-score semantic similarity. Instead of viewing person-job fit as a monolithic problem, MAALEM deconstructs it into a series of transparent, independent, and human-interpretable dimensions.

The framework is designed not to replace the recruiter, but to empower them. It shifts the paradigm from a simple filter to a sophisticated diagnostic tool, providing a holistic profile of a candidate's strengths and weaknesses against a job's specific requirements. By prioritizing transparency, justification, and user control, MAALEM transforms the AI from an opaque oracle into a genuine decision-support partner.

0.8 Project Objectives

In light of this dual-track approach, the specific objectives of this project were as follows:

1. To **build and deliver** a production-ready semantic matching engine that fulfills all specified industrial requirements for parsing, embedding, and high-speed vector search.
2. To engineer prompt instructs the model to analyze the raw text of a document and return a structured JSON object containing key fields such as skills, experience, and education. This approach leverages the LLM's world knowledge and contextual understanding to achieve high-fidelity data extraction, even from non-standard or complex documents.

0.8.1 Vector Embeddings for Semantic Representation

To enable semantic search, the structured information extracted by the LLM is first synthesized into a single, cohesive text block. This composite text is then converted into a dense vector embedding using a model from the 'sentence-transformers' library[3]. The chosen model, 'paraphrase-multilingual-MiniLM-L12-v2', is highly efficient and pre-trained on a massive corpus, making it exceptionally effective at understanding semantic similarity across different phrasings and even across multiple languages. The resulting 384-dimensional vector serves as a rich, numerical "fingerprint" of the candidate or job offer.

0.8.2 High-Speed Retrieval with Vector Databases

A brute-force search comparing a query vector against thousands or millions of candidate vectors is computationally infeasible. To implement optimized search functionality, the system is powered by **FAISS (Facebook AI Similarity Search)**[5], a high-performance library for efficient similarity search and clustering of dense vectors. Each candidate's vector is normalized and stored in a

FAISS index. When a job offer is processed, its vector is used to query this index, which leverages Approximate Nearest Neighbor (ANN) algorithms to retrieve the top N most similar CVs with millisecond latency.

0.8.3 The Integrated Industrial System

These three components Gemini for parsing, Sentence-Transformers for embedding, and FAISS for search are integrated into a cohesive pipeline managed by a command-line interface. The system provides two primary functions: ‘index’, which processes a directory of CVs and populates the FAISS database, and ‘match’, which takes a job offer and returns a ranked list of candidates. This delivered system represents a powerful, production-ready semantic matching engine that meets all specified industrial requirements.

0.9 Advancing the State-of-the-Art: The MAALEM Philosophy

While the industrial engine is highly effective, its inherent “black box” nature presented a clear opportunity for scientific innovation. This led to the development of the **MAALEM (Multi-Axis Agile Learning Engine for Matching)** framework, the core research contribution of this thesis.

The ‘MAALEM’ project was founded on the principle that for a tool to be truly useful in a high-stakes domain, it must be trustworthy. Trust, in this context, requires transparency and explainability. The ‘MAALEM’ philosophy, therefore, represents a paradigm shift away from monolithic similarity scores. It deconstructs the complex person-job fit problem into a series of transparent, independent, and interpretable dimensions. It moves beyond simple semantic similarity to conduct a multi-faceted analysis, providing recruiters not just with a recommendation, but with the evidence and reasoning behind it. The framework is architected to be a **decision-support partner**, transforming the AI from an inscrutable filter into a transparent co-pilot that empowers the human expert.

0.10 Core Objectives and Report Structure

In light of this dual approach, the specific objectives of this project were as follows:

- (a) To **build and deliver** a production-ready semantic matching engine that meets all specified industrial requirements for parsing, embedding, and high-speed search.
- (b) To **design the ‘MAALEM’ framework**, a novel, explainable matching architecture that prioritizes transparency and justified reasoning over opaque similarity scores.
- (c) To **create a rigorous, custom-built evaluation suite** of benchmarks to scientifically measure the performance and robustness of different matching paradigms.
- (d) To **empirically demonstrate** the superior capabilities of the advanced ‘MAALEM’ framework, comparing its performance against the Industrial Baseline and other standard models.

This report will guide the reader through this journey. **Chapter 1** provides a critical review of the state of the art. **Chapter 2** details the design of both the Industrial Baseline Engine and the high-level principles of the ‘MAALEM’ framework. **Chapter 3** describes the custom evaluation protocol. **Chapter 4** presents the comparative experimental results, followed by a **Discussion** in **Chapter 5** and a **Conclusion**.

Chapter 1

From Manual Craft to Automated Systems: A Review of Person-Job Matching

Before designing a technological solution, it is imperative to understand the domain and the best practices of its expert practitioners. This chapter delves into the "métier" of the recruiter, exploring the principles of a rigorous, manual evaluation process. We will first outline the anatomy of traditional recruitment and its gold-standard evaluation grid. We will then diagnose why this manual process fails at modern scales, establishing the context for the rise of automated systems. Finally, we will conduct a critical review of the dominant technological paradigms in automated matching, which will culminate in a justification for the novel research framework proposed in this thesis.

1.1 The Recruiter's Craft: The Manual Gold Standard

A thorough, professional recruitment cycle is a structured, multi-stage process designed to systematically reduce uncertainty and identify the best possible candidate. While variations exist, the classic workflow generally includes needs analysis, sourcing, screening, interviews, structured evaluation, and finally, the offer. The most critical component for ensuring a high-quality, fair, and legally defensible hiring decision is the use of a structured evaluation methodology, historically accomplished with a **candidate evaluation grid**.

This manual "table" is what expert recruiters create to enforce consistency and mitigate the natural human biases that can derail an evaluation. A typical grid would have candidates listed as rows and key job requirements as columns. These columns are focused on tangible, verifiable evidence:

- **Education Level & Field of Study**
- **Years of Relevant Experience**
- **Key Technical Skills (Hard Skills)**
- **Key Conceptual Knowledge (Conceptual Skills)**
- **Certifications & Languages**

Notably, one category of attributes is intentionally absent from this initial CV evaluation grid: **soft skills**. Expert recruiters understand that attributes like "communication," "teamwork," or "leadership" are effectively unverifiable from a document. The assessment of these crucial interpersonal traits is therefore reserved for the interactive interview stages, where recruiters employ behavioral interviewing techniques and situational questions. This separation of concerns is critical: the CV screening phase is a filter for *verifiable hard qualifications*, while the interview is an assessment of *demonstrable soft skills*.

1.1.1 The Scalability Crisis

This rigorous evaluation grid is the gold standard for quality, but it has a fatal flaw in the modern era: **it does not scale**. The meticulous process of reading a CV in-depth and populating a detailed scoring grid for a single candidate can take significant time. When faced with hundreds or even thousands of applicants for a single role, this manual best practice becomes logistically impossible. Recruiters are forced into a difficult compromise: either abandon the grid and revert to biased, intuitive screening, or apply it to a tiny sample of candidates pre-filtered by a crude tool. This scalability crisis defines the central challenge for recruitment technology: the need to **automate the rigor of the manual evaluation grid at scale**.

1.2 The State of the Art in Automated Matching Systems

The evolution of person-job fit systems has been driven by advancements in Natural Language Processing and Information Retrieval. This evolution can be understood as a progression through two major paradigms, each with distinct strengths and fundamental weaknesses.

1.2.1 Paradigm 1: Lexical Matching - The Brittle Foundation

The first generation of automated screening systems, and many still in use today, rely on lexical matching. This methodology compares the literal strings of text in a resume against those in a job description, operating without any true comprehension of the underlying meaning. The most basic form is direct **keyword search**, while a more sophisticated evolution is **TF-IDF (Term Frequency-Inverse Document Frequency)**[2], which creates a vector representation of documents where terms are weighted by their statistical importance.

Despite their simplicity and speed, both methods are fundamentally flawed because they operate on a "bag-of-words" principle. This ignorance of semantics leads to critical failures in a recruitment context:

- **Failure to Recognize Synonyms and Acronyms:** A lexical system cannot comprehend that "LLMs" and "Large Language Model" refer to the exact same concept. A qualified candidate is immediately penalized for a simple, good-faith terminological variation.
- **Failure to Grasp Conceptual Equivalence:** The system is incapable of understanding intent. If a job requires experience with "recommender systems," it will fail to match a candidate who describes their work as building a "model that personalizes results based on users." Although conceptually identical, the difference in wording results in a missed opportunity and a false negative.

In essence, lexical methods are too rigid. They can identify exact textual matches but are completely blind to meaning, forcing an over-reliance on precise terminology and unjustly penalizing many of the most qualified candidates.

1.2.2 Paradigm 2: Semantic Matching - The Powerful Black Box

To overcome the rigidity of lexical methods, the field has moved towards semantic matching, which represents the current state-of-the-art and is the architectural basis for the **Industrial Baseline** system delivered in this project. Powered by deep learning models like **SBERT (Sentence-BERT)**[3], these techniques represent text as dense vector embeddings in a high-dimensional space. In this space, proximity between vectors indicates semantic similarity, allowing the system to understand that "Large Language Model" and "LLMs" are indeed related.

While this solves the synonym problem, it introduces a new and more insidious set of issues for the recruitment domain: a debilitating trade-off between semantic power and practical utility.

The Loss of Specificity

Semantic models are designed to generalize, which is often a strength but becomes a critical flaw when specificity is a hard requirement. By clustering concepts based on broad meaning, they erase crucial, non-negotiable distinctions. A classic example illustrates this failure perfectly: a company requires a senior developer with ten years of experience in **Java**. A semantic model may assign a high similarity score to a candidate who is an expert in **Django**, a Python framework. The model correctly identifies that both are "backend web development technologies" and thus semantically close. However, for the recruiter and the hiring manager, this recommendation is a **critical failure**. The tools are not interchangeable, and the system has wasted the recruiter's time by ignoring a hard constraint.

The "Just Trust Me" Black Box

The most fundamental flaw of these models is their opacity. A semantic model outputs a single similarity score, for example, 0.87, but provides **no justification** for its reasoning. This "black box" nature is fundamentally at odds with the professional recruitment process.

- **They are not explainable:** A recruiter cannot defend their decision to a hiring manager by saying, "The algorithm gave a score of 0.87." They need to be able to state, "This candidate is a strong match on skills and education, but is slightly below on years of experience."
- **They are not flexible:** The recruiter cannot adjust the model's internal logic to prioritize one criterion over another (e.g., "for this role, skills are more important than degree"). The model's reasoning is monolithic and unchangeable.
- **They are easily fooled:** A CV can be stuffed with semantically related keywords to achieve a high score, even if the candidate's experience is superficial.

These systems force the user into a position of blind faith, which is untenable in a high-stakes, decision-driven environment.

1.3 Synthesis and Justification for a New Approach

This analysis of the state of the art reveals a critical dilemma. Person-job matching systems are trapped between two flawed paradigms:

- (a) **Lexical Methods:** Transparent and specific, but brittle and lacking in semantic understanding.
- (b) **Semantic Methods:** Semantically powerful, but opaque, imprecise, and inflexible "black boxes."

The industrial system delivered as part of this project is a powerful embodiment of the second paradigm. It is a state-of-the-art semantic engine, yet it inherits all the "black box" limitations discussed. In the domain of recruitment, **explainability and the ability to justify a decision are not optional features; they are the single most important requirement**. A tool that cannot explain its recommendations is not a decision-support tool; it is a source of uncertainty.

This review demonstrates a clear and pressing need for a new framework that resolves this dilemma. The ideal system must be architected from the ground up to deliver on three core principles simultaneously:

- **Explainability:** Providing a transparent, decomposable score that shows *why* a candidate is a good match.
- **Specificity:** Understanding the difference between semantically similar but functionally distinct concepts.
- **Flexibility:** Empowering the recruiter to tune the matching criteria to fit the unique needs of each role.

This thesis proposes such a framework the **MAALEM** engine which is designed precisely to fill this gap. It aims to create a hybrid system that leverages the best of all approaches while centering the recruiter's need for transparent, trustworthy, and actionable insights.

Chapter 2

Design and Methodology

Having critically examined the limitations of existing person-job fit systems in the preceding chapter, we now present the technical design of the two distinct systems developed during this project. This chapter serves as the architectural blueprint for both the practical and research-oriented phases of our work.

The methodology will be presented in two major parts.

- **Part 1** provides a comprehensive technical breakdown of the **Industrial Baseline Engine**. We will detail its three-stage pipeline (Parse, Embed, Search) and the specific technologies Google Gemini, Sentence-Transformers, and FAISS chosen to fulfill the industrial mission of delivering a state-of-the-art semantic search solution.
- **Part 2** introduces the advanced **MAALEM framework**. This section will focus on the high-level philosophy and guiding principles that make it a superior, explainable alternative to the "black box" baseline.

This structured approach will first document the powerful system delivered to the host company, and then articulate the innovative concepts that constitute the core research contribution of this thesis.

2.1 Approach 1: The Industrial Baseline - A Semantic Search Engine

The primary objective of the industrial phase of this project was to design and implement a robust, end-to-end system for candidate-job matching that met a specific set of functional requirements. The resulting system, referred to as the "Industrial Baseline," is architected as a modern semantic search engine. This section provides a comprehensive technical blueprint of its design, components, and workflow.

2.1.1 Architectural Overview and Technology Stack

The engine is designed as a linear, three-stage data processing pipeline. Each stage is handled by a specialized, state-of-the-art technology chosen for its performance and suitability for the task. The high-level architecture follows this logical flow:

- (a) **Parse:** Raw, unstructured text from source documents (CVs and job offers) is ingested and parsed into a structured, machine-readable format.
- (b) **Embed:** The structured data is then transformed into a high-dimensional numerical vector, or "embedding," that captures its semantic meaning.
- (c) **Search:** The embedding is stored in or used to query a specialized vector database, which retrieves the most semantically similar documents with high efficiency.

To implement this architecture, a specific stack of open-source and API-driven technologies was selected:

- **Parsing Engine: Google Gemini API**[4]. Chosen for its superior natural language understanding and instruction-following capabilities, which are essential for handling the diverse and unpredictable formats of recruitment documents.
- **Embedding Model: Sentence-Transformers Library**[3]. This library is the industry standard for generating high-quality text embeddings. It provides access to a vast range of pre-trained models optimized for semantic similarity tasks.
- **Vector Database: FAISS (Facebook AI Similarity Search)**[5]. Selected for its world-class performance, scalability, and efficiency in conducting similarity searches on large-scale vector datasets, a critical requirement for any production-grade system.

The following subsections will provide a detailed technical breakdown of how each of these components was implemented to fulfill the project's industrial mandate.

2.1.2 Component 1: High-Fidelity Parsing and Information Extraction

The most significant challenge in any recruitment automation system is reliably extracting structured information from the chaotic and unstructured format of a resume. Traditional rule-based parsers are brittle and often fail when encountering a novel CV layout. To overcome this, the Industrial Baseline engine leverages a Large Language Model (LLM) as its core parsing component as illustrated in Figure 2.1.

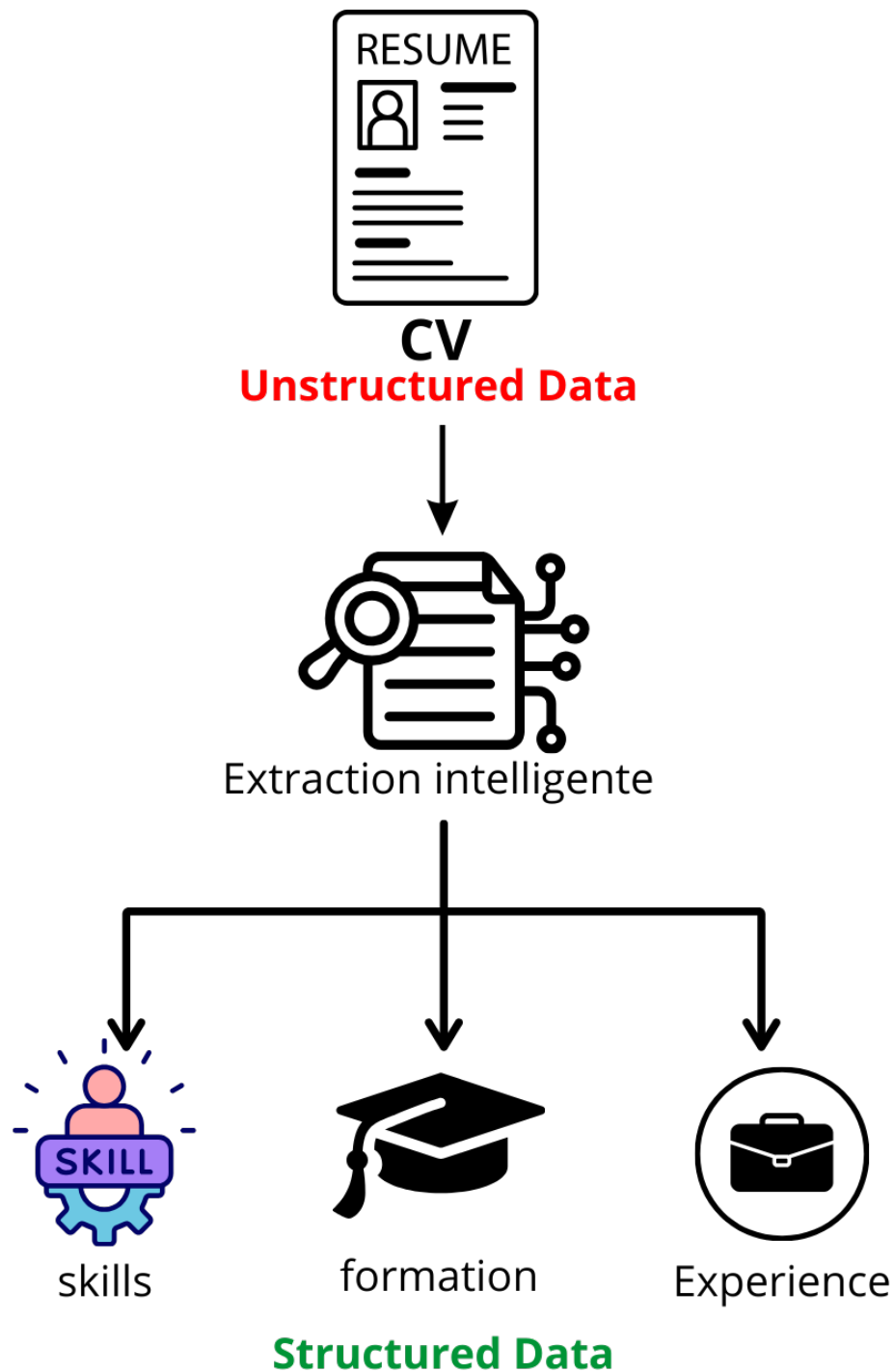


Figure 2.1: The Information Extraction Pipeline. Unstructured text from a source document is processed by the Gemini API to produce a structured JSON profile.

Rationale for an LLM-Based Approach

The decision to use an LLM-based parser was driven by several key advantages over conventional methods:

- **Format Agnosticism:** LLMs are not dependent on a fixed template. They can comprehend information from single-column, double-column, text-based, or graphically rich CVs with equal proficiency.
- **Contextual Understanding:** Unlike simple keyword extractors, an LLM can infer meaning

from context. It can distinguish between a skill listed under a "Skills" section versus one mentioned in the description of a past project, and it can understand implied timelines from job dates.

- **Implicit Normalization:** LLMs can handle a wide range of linguistic variations, correctly identifying entities even when they are described with unconventional phrasing.

Implementation via Prompt Engineering

The interaction with the Google Gemini API is managed through a carefully engineered, multi-part prompt. The prompt is designed to be highly prescriptive to ensure consistent and reliable output. Its core instructions are as follows:

- Define the Role:** The prompt first assigns the LLM the role of an "expert recruitment data extractor."
- Specify the Output Format:** The model is strictly instructed to return its findings **only** as a single, minified JSON object. This is a critical instruction that prevents the model from returning conversational text, code blocks, or other extraneous information, making the output directly machine-parsable.
- Define the Schema:** The prompt explicitly defines the required JSON schema, listing the keys the model must populate: **summary**, **skills**, **education**, **experience**, **languages**, and **certifications**.
- Instruct on Missing Data:** The model is directed to return an empty string (""") as the value for any field it cannot find in the source document, ensuring the final JSON object always has a consistent structure.
- Provide the Input Text:** Finally, the raw text extracted from the source document is appended to the prompt, clearly delineated by markers (e.g., **TEXT STARTS HERE**).

This structured prompting strategy transforms the powerful but general-purpose LLM into a specialized and reliable data extraction tool, forming the foundational first stage of the matching pipeline.

Exploration of Alternative Extraction Methods

While the LLM-based approach was ultimately selected for the final Industrial Baseline due to its superior accuracy and flexibility, it was not the only method investigated. A significant parallel research effort was undertaken to explore the feasibility of a more traditional, open-source Named Entity Recognition (NER) approach using a model from the **spaCy** library[6].

This endeavor faced a critical obstacle common in industrial settings: the complete lack of pre-existing, annotated training data. To overcome this, a separate sub-project was initiated to create a dataset via a semi-automated annotation process. This involved using an LLM to bootstrap an initial set of annotations, which were then refined and formatted for training.

A large spaCy model (**en_core_web_lg**) was then fine-tuned on this bespoke dataset. The performance of this custom-trained model, named **cv_parser_model_lg**, was evaluated on a held-out test set. The results are summarized in Table 2.1.

Table 2.1: Performance of the custom-trained spaCy NER model ('cv_parser_model_lg') on key entities.

Entity Type	spaCy (lg) F1-Score
SKILL / TECHNICAL_SKILL	0.70
DEGREE	0.75
EXPERIENCE_DETAIL	0.69
INSTITUTION	0.72
FIELD_OF_STUDY	0.58
LANGUAGE	0.44
CERTIFICATION	0.30
YEARS_OF_EXPERIENCE	0.24
Macro Average F1	0.66

While the model achieved a respectable Macro F1-score of 0.66 and showed promise on core entities like ‘DEGREE’ and ‘INSTITUTION’, its performance on more nuanced and unstructured entities like ‘CERTIFICATION’ and ‘YEARS_OF_EXPERIENCE’ was insufficient for a high-stakes production environment. This investigation ultimately validated the decision to use the more powerful, zero-shot capabilities of the Gemini API for the Industrial Baseline, as the significant engineering effort required to build and maintain a custom NER model did not yet yield a competitive level of accuracy.

2.1.3 Component 2: Semantic Vector Embedding

Once the parsing stage has converted a raw document into a structured JSON object, the next critical step is to represent this information in a way that captures its underlying meaning. This is the role of the semantic embedding component. The goal is to move beyond mere keywords and create a numerical “fingerprint” of the candidate or job offer that can be compared for conceptual similarity as visualized in Figure 2.2.

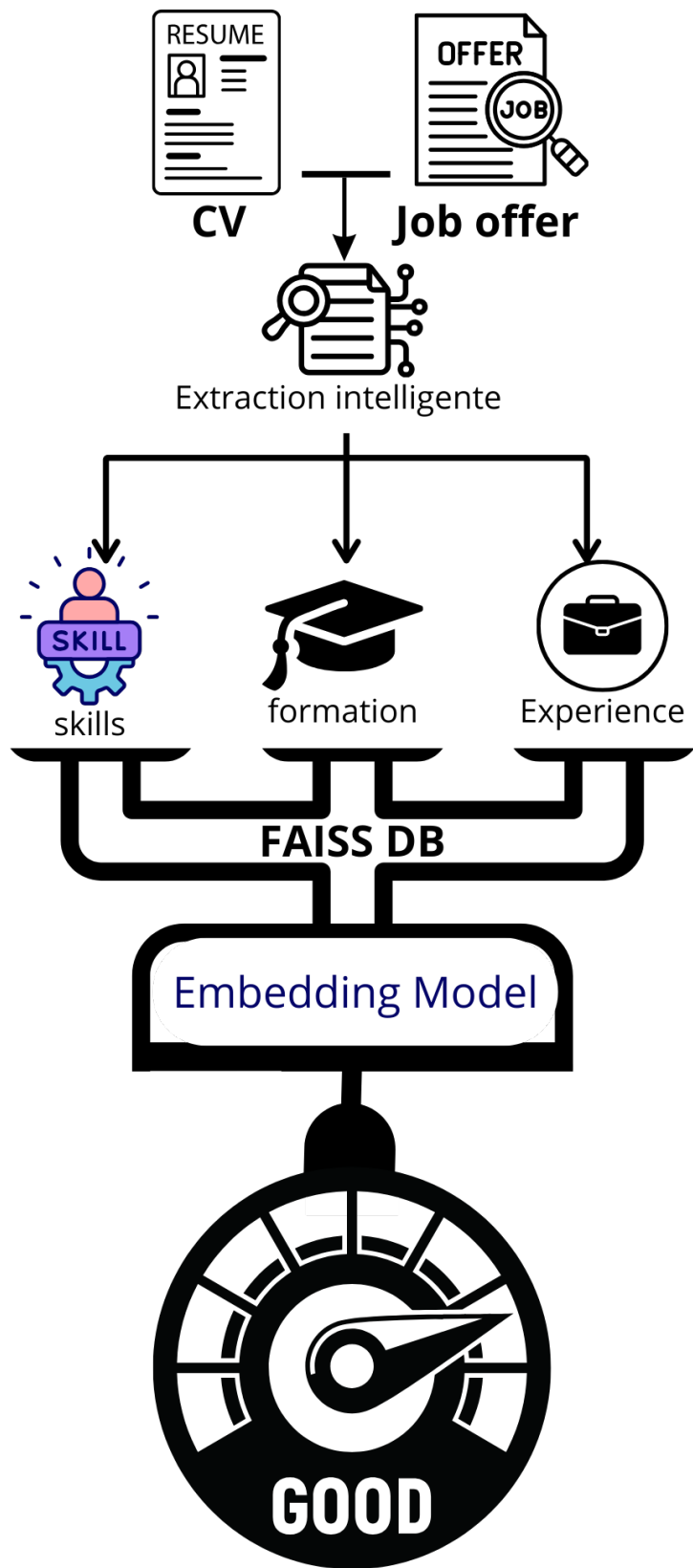


Figure 2.2: The Semantic Matching Workflow. Structured profiles are converted into vectors. The job offer's vector is used to query the FAISS database of CV vectors, producing a ranked list based on similarity.

The Concept of Semantic Embeddings

A semantic embedding is a dense vector of floating-point numbers that represents a piece of text in a high-dimensional space. This representation is generated by a deep learning model, typically a Transformer[7], which has been pre-trained on a massive corpus of text. Through this training, the model learns the relationships between words, phrases, and concepts. The resulting vector space has a key property: texts with similar meanings are located geometrically close to one another. This allows us to perform powerful comparisons using mathematical distance metrics, forming the basis of modern semantic search.

Implementation: From Structured Data to a Semantic Vector

The process of generating an embedding for a document in this system involves two main steps:

- (a) **Composite String Creation:** A single, holistic embedding is desired for each document. To create the source text for this embedding, the system first synthesizes the structured data from the JSON object into a coherent, descriptive "composite string." This is achieved by iterating through the JSON's key-value pairs and concatenating them into a human-readable format, such as:
`"SKILLS: Python, SQL, Scikit-learn, TensorFlow. EDUCATION: Master of Science in Data Science. EXPERIENCE: 3 years experience in machine learning and data analysis..."`
This method ensures that all extracted information contributes to the final semantic representation.
- (b) **Vector Generation:** This composite string is then passed to the chosen embedding model from the `sentence-transformers` library. The model processes the text and outputs a 384-dimensional vector. This vector is the final semantic fingerprint of the document, ready for storage or search.

Justification for the Chosen Model

The specific model selected for this task is **paraphrase-multilingual-MiniLM-L12-v2**. This choice was deliberate and based on three key criteria aligning with the project's industrial requirements:

- **Performance:** It is a well-established model that provides a strong balance of speed and accuracy on semantic textual similarity (STS) benchmarks.
- **Efficiency:** As a "MiniLM" (Mini Language Model) variant, it is significantly smaller and faster than larger models like BERT-base, making it highly suitable for production environments where computational resources may be a consideration.
- **Multilingual Capability:** Crucially, this model is trained on parallel data from over 50 languages. This gives the system the inherent ability to match a CV written in French against a job offer written in English, as it maps both to a shared semantic space. This was a key implicit requirement for a global recruitment platform.

2.1.4 Component 3: High-Performance Vector Search and Retrieval

With a method for generating semantic vectors, the final architectural component required is a system for storing and efficiently searching through a large collection of these vectors. A naive approach, such as iterating through a list and calculating the cosine similarity for each vector pair, is computationally expensive and does not scale beyond a few thousand entries. This necessitates the use of a specialized vector database.

The Need for a Dedicated Vector Database

A vector database is optimized for a single task: finding the nearest neighbors to a query vector in a high-dimensional space. Unlike traditional databases that are optimized for structured data queries (e.g., using SQL), vector databases use specialized indexing algorithms to perform similarity

searches orders of magnitude faster. For an application intended to scale to potentially millions of CVs, a dedicated vector search library is not optional; it is a core architectural requirement.

Implementation with FAISS

The Industrial Baseline engine uses **FAISS (Facebook AI Similarity Search)**[5] as its vector database. FAISS is a library, not a standalone server, which makes it lightweight and easy to integrate directly into a Python application. The implementation follows a precise workflow:

- (a) **Index Initialization:** The system uses the `IndexFlatIP` index from FAISS. This index computes the Inner Product between vectors. For vectors that are L2-normalized (i.e., their length is 1), the Inner Product is mathematically equivalent to the Cosine Similarity, which is the standard metric for this task. The index is initialized to expect vectors of dimension 384, matching the output of our embedding model.
- (b) **Vector Storage (The ‘index’ command):** When a new CV is to be indexed, its generated 384-dimensional vector is first L2-normalized. This normalized vector is then added to the FAISS index using the `add` method. Concurrently, the document’s metadata (its filename and the extracted JSON data) is saved to a separate JSON mapping file. The position (ID) of the vector within the FAISS index corresponds to the position of its metadata in the mapping file, creating a link between the vector and its source information. After each addition, the FAISS index and the metadata map are saved to disk to ensure persistence.
- (c) **Vector Retrieval (The ‘match’ command):** When a job offer is provided, it is converted into a query vector, which is also L2-normalized. This query vector is then passed to the FAISS index’s `search` method, along with the desired number of results (‘k’). FAISS rapidly returns the indices and similarity scores of the top ‘k’ most similar vectors from the database. The system then uses these returned indices to look up the corresponding metadata from the JSON mapping file, presenting the final ranked list of candidates to the user.

This integration with FAISS ensures that the system’s search functionality is not only accurate but also highly performant and ready for large-scale industrial deployment.

2.1.5 System Workflow and Command-Line Interface

The three core components parsing, embedding, and search are integrated into a cohesive system that is exposed to the user through a clean and functional Command-Line Interface (CLI). The CLI is built using Python’s standard `argparse` library, providing a self-documenting and user-friendly way to interact with the engine. This design allows the system to be easily executed manually or integrated into larger automated scripts and workflows.

The typical end-to-end user workflow is designed to be straightforward, involving two primary operations: populating the database with candidate CVs, and then matching a job offer against that database.

The ‘index’ Command: Populating the Candidate Database

The first step in any use case is to build a searchable database of candidates. The `index` command is designed for this purpose.

- **Syntax:** `python main.py index [path/to/cv_folder]`
- **Functionality:** The user provides the path to a folder containing candidate CVs in either `.pdf` or `.txt` format. The script then recursively scans this folder and processes each document it finds. For every CV, it executes the full pipeline:
 - (a) The raw text is extracted.
 - (b) The text is sent to the Gemini API for parsing into a structured JSON object.
 - (c) The JSON data is synthesized into a composite string.
 - (d) The string is converted into a 384-dimensional semantic vector.
 - (e) The vector is L2-normalized and added to the FAISS index.

(f) The associated metadata (filename, extracted JSON) is stored.

The script provides real-time feedback in the console, printing the name of each file as it is processed and confirming when it has been successfully indexed.

The ‘match’ Command: Finding Suitable Candidates

Once the database is populated, a user can find the best-matching candidates for a specific role using the `match` command.

- **Syntax:** `python main.py match [path/to/job_offer.txt] --top-k [N]`
- **Functionality:** The user provides the path to a single job offer file. The engine processes this file through the same parsing and embedding pipeline as the CVs, generating a query vector. This vector is then used to search the FAISS index.
- **Output:** The command returns a ranked list of the top candidates. For each candidate, it displays their rank, the original filename of their CV, and their similarity score (expressed as a percentage). The optional `--top-k` argument allows the user to specify how many results they wish to see (the default is 5).

Utility Commands: ‘demo’ and ‘clean’

To improve usability and testability, two additional utility commands were implemented.

- **The ‘demo’ command:** This fully automated command requires no user input or files. It creates a temporary, isolated environment, generates a small set of sample CVs and a sample job offer, runs the full ‘index’ and ‘match’ cycle, displays the results, and then automatically cleans up all temporary files. Its primary purpose is to provide a quick, one-shot method to verify that the system is installed correctly and all components are functional.
- **The ‘clean’ command:** This command deletes the entire FAISS database and all associated metadata files. It allows a user to easily reset the system to a clean state, which is useful for starting a new batch of indexing or for testing purposes.

This set of commands provides a complete and functional interface to the engine’s capabilities. With the conclusion of this section, the technical description of the Industrial Baseline system the first major deliverable of this project is complete. The following section will introduce the advanced research framework that was developed to transcend the limitations of this powerful but opaque engine.

2.2 Approach 2: The MAALEM Framework - A New Philosophy for Matching

The Industrial Baseline engine, as detailed in the previous section, represents a powerful and effective implementation of the current state-of-the-art in semantic search. It is fast, scalable, and capable of understanding conceptual relationships in text, successfully fulfilling the industrial mandate of the project. However, a critical analysis of its architecture and output reveals a fundamental, inherent limitation shared by all systems of its kind: it operates as an opaque “black box.”

This limitation is not a flaw in its implementation, but a feature of its paradigm. The final output a single, holistic similarity score is inherently unexplainable. It tells a user *that* a candidate is a good match, but it cannot tell them *why*. This opacity presents a significant barrier to trust and utility in a high-stakes professional domain like recruitment, where every decision requires justification, auditability, and nuanced human judgment.

2.2.1 Core Philosophy: Beyond the Black Box

The recognition of this "explainability gap" was the primary catalyst for the research phase of this project, which culminated in the development of the **MAALEM (Multi-Axis Agile Learning Engine for Matching)** framework. MAALEM is not an alternative algorithm, but a new philosophy for designing recruitment AI, founded on a single, guiding principle: **a trustworthy system must be a transparent system.**

The core philosophy of MAALEM is to reject the single, monolithic score in favor of **deconstruction and explainability**. It posits that the complex concept of "person-job fit" is not a singular, measurable quantity. Rather, it is a composite judgment derived from evaluating a candidate across multiple, distinct criteria. An expert human recruiter does not think in terms of a single percentage; they think in terms of strengths and weaknesses across different facets: "Excellent technical skills, but a bit light on a specific degree," or "Perfect educational background, but needs more years of hands-on experience."

MAALEM is designed to mimic this expert human reasoning process. It deconstructs the matching problem into a series of independent, understandable dimensions, evaluates the candidate along each one, and presents the results as a detailed, multi-faceted profile. This approach transforms the AI from an inscrutable "black box" into a transparent "glass box," allowing the recruiter to see the internal logic and use the system's output not as a command, but as a rich source of evidence to support their own expert decision-making.

2.2.2 The Three Guiding Principles of MAALEM

The MAALEM framework is not defined by a single technology, but by a set of three interdependent design principles. These principles collectively guide the architecture towards the ultimate goal of creating a trustworthy, transparent, and genuinely useful decision-support tool for recruitment professionals.

Principle 1: Deconstruction of Complexity

The first and most fundamental principle is the deconstruction of the matching problem. MAALEM rejects the premise that "person-job fit" can be captured by a single, all-encompassing score. Instead, it treats it as a multi-faceted concept that must be analyzed along several independent dimensions. The framework isolates distinct attributes of a candidate's profile such as their technical competencies, academic qualifications, and professional tenure and evaluates each one individually.

This approach has a profound impact: it replaces a single, ambiguous similarity score with a detailed profile of discrete, understandable sub-scores. This prevents the "averaging effect" common in black-box systems, where a candidate's exceptional strength in one area can be unfairly masked by a weakness in another. By breaking down the problem, the system can provide a much richer and more accurate picture of a candidate's suitability.

Principle 2: Explainability by Design

The second principle dictates that transparency cannot be an afterthought; it must be woven into the very architecture of the system. In the MAALEM framework, every component is designed to produce an output that is not only a score, but also the evidence supporting that score.

When the system presents its analysis, it does not simply show a number. It provides a clear justification that is directly traceable to the source data. For example, instead of just a "Skills Score" of 90%, it can articulate which specific skills from the job offer were matched on the candidate's CV. This principle ensures that for every piece of the evaluation, a recruiter can ask "Why?" and the system can provide a clear, evidence-based answer. This moves the system from the realm of opaque AI to that of auditable, data-driven analysis.

Principle 3: User-Centric Control

The final principle is that the AI must serve as a co-pilot, not a dictator. The human recruiter, with their domain expertise and understanding of the nuanced, unstated needs of a team, must always remain in control of the final decision. The MAALEM framework is designed to empower this control, not usurp it.

This is achieved by making the system’s evaluation logic both transparent and tunable. The framework allows the recruiter to dynamically adjust the relative importance of each evaluation dimension. For a highly specialized senior role, a recruiter can instruct the system to heavily prioritize “Technical Skills” and “Years of Experience.” For an entry-level research position, “Educational Attainment” and “Field of Study” could be made paramount. This flexibility allows the recruiter to tailor the AI’s analytical lens to the unique context of every single job opening, ensuring the final ranking perfectly aligns with their strategic priorities.

2.2.3 Conceptual Workflow: From Recommendation to Dialogue

The culmination of these three principles results in a user experience that is fundamentally different from that of a traditional semantic search engine. The workflow transforms from a static query-and-response into a dynamic, interactive dialogue between the recruiter and the system.

A user’s interaction with the Industrial Baseline engine is linear and definitive: they submit a job offer and receive a ranked list of candidates. The interaction ends there. In contrast, the conceptual workflow of the MAALEM framework is cyclical and exploratory:

- (a) **Initial Analysis:** The recruiter first receives an initial analysis, where candidates are ranked based on a default or pre-configured weighting of the different evaluation dimensions.
- (b) **Diagnostic Review:** Instead of a simple list of scores, the primary output is a rich, visual profile for each candidate (e.g., a radar chart). This allows the recruiter to instantly diagnose a candidate’s specific strengths and weaknesses relative to the job requirements. They can immediately see if a candidate is a well-rounded generalist or a “spiky” specialist.
- (c) **Interactive Refinement:** Based on this diagnostic insight, the recruiter can then engage in a “what-if” analysis. Using the control interface, they can ask new questions by adjusting the system’s logic in real-time. For example: “What if, for this specific role, we value practical skills twice as much as academic background?” The system instantly re-calculates and re-ranks the candidates based on this new priority, providing immediate feedback.
- (d) **Informed Decision:** Through this iterative process of analysis and refinement, the recruiter arrives at a final shortlist that is not just algorithmically generated, but is also aligned with their own expert judgment and strategic priorities.

This workflow represents a paradigm shift. The Industrial Baseline provides a *recommendation*, a static answer to a question. The MAALEM framework facilitates a *dialogue*, providing a dynamic tool that helps the recruiter formulate better questions and explore the trade-offs inherent in any complex hiring decision. It successfully elevates the AI from a simple filter to a true decision-support partner.

2.2.4 Chapter Conclusion

This chapter has detailed the design and methodology of the two distinct systems developed during this project. We first provided a comprehensive architectural blueprint for the **Industrial Baseline Engine**, a powerful, state-of-the-art semantic search system that successfully fulfilled all requirements of the initial project mandate. We then introduced the **MAALEM framework**, an advanced research concept born from a critical analysis of the baseline’s “black box” limitations. While the baseline prioritizes speed and holistic similarity, the MAALEM framework is architected around the core principles of deconstruction, explainability, and user-centric control. The following chapters will now move from design to validation, presenting a rigorous empirical comparison of these two distinct philosophies to demonstrate their respective capabilities in real-world scenarios.

Chapter 3

Evaluation Framework

A model’s theoretical elegance or architectural novelty is meaningless without rigorous, empirical validation. While the preceding chapters have detailed the design and methodology of our multi-axis recommender engine, this chapter establishes the scientific protocol by which its performance is measured. The central theme of this protocol is the response to a fundamental challenge in the person-job fit domain: the profound lack of standardized evaluation benchmarks. This chapter details our solution to this challenge the design and implementation of a novel, comprehensive suite of synthetic benchmarks and outlines the baseline models against which our system will be compared.

3.1 The Benchmarking Challenge in Person-Job Fit

In established machine learning domains, progress is often accelerated by the availability of standardized, public benchmarks. Fields like computer vision have ImageNet, natural language processing has GLUE[8], and the evaluation of large language models (LLMs) is driven by a diverse array of benchmarks testing for specific capabilities such as mathematical reasoning (MATH)[9], coding (HumanEval)[10], and scientific knowledge (MMLU)[11]. These shared tasks provide a “gold standard” against which new architectures can be objectively compared, allowing the entire field to measure progress.

The domain of person-job fit, however, critically lacks such a standard. There are no widely accepted, open-source datasets for ranking the quality of a match between a set of candidates and a job offer. Commercial datasets are proprietary and inaccessible, while the few available academic datasets are often small, domain-specific, or lack the nuanced, multi-faceted annotations required to test a sophisticated recommendation system like ours.

This absence presents a significant scientific obstacle. Without a standard for comparison, how can one objectively prove that a new approach is superior to existing methods? One is left in a state of unverified claims.

To overcome this fundamental challenge, and to ensure our system could be rigorously evaluated against our predefined success criteria of **Robustness**, **Reliability**, and **Performance**, we undertook the task of designing and implementing a novel suite of synthetic benchmarks. This suite is not a single test, but a multi-faceted evaluation framework, with each benchmark tailored to probe a specific, critical capability of a candidate recommendation system.

3.2 The Evaluation Framework: A Suite of Three Benchmarks

Our evaluation framework is not a single test but a cohesive suite of three distinct benchmarks. Each benchmark is designed as a specialized probe, targeting a different aspect of the recommender system’s performance. The suite is designed to answer three critical questions:

- (a) How well does the system perform under realistic, noisy conditions, where the primary goal is to effectively separate qualified candidates from unqualified ones?
- (b) How robust is the system against adversarial or deceptive data? Can it discover "hidden gems" while avoiding superficially attractive but unsuitable "traps"?
- (c) How precise is the system's core ranking algorithm when evaluated in a controlled, noise-free environment?

Table 3.1 provides a high-level overview of the three benchmarks and their primary objectives. The subsequent sections will detail the methodology and specific metrics for each.

Table 3.1: Overview of the Custom Benchmark Suite.

Benchmark Name	Primary Objective
Binary Noisy Rank "BNR"	To measure the model's ability to effectively filter good candidates from bad in a realistic scenario where the ground-truth ranking is inherently subjective and noisy.
Trapped Binary Noise Rank "TBNR"	To assess adversarial robustness by measuring the model's ability to discover non-obvious "hidden gem" candidates while simultaneously rejecting deceptive "trap" candidates.
Absolute Rank "AR"	To evaluate pure ranking precision and sensitivity by testing the model's ability to correctly order candidates along a perfectly defined, incremental quality scale.

3.2.1 Benchmark 1: "Binary Noisy Rank" — Testing for Real-World Efficacy

This first benchmark is designed to simulate the most common and crucial task a recruiter faces: performing an initial screening to separate the pool of promising candidates from the clearly unqualified ones. In this real-world context, the exact ranking among the top candidates can be subjective, but the binary classification of "good" vs. "bad" is paramount.

Objective

The primary objective of this benchmark is to evaluate the model's performance under realistic, noisy conditions. It tests the system's ability to mimic the judgment of a human recruiter, where there is inherent subjectivity in ranking the top contenders, but a clear distinction between those who are qualified and those who are not.

Methodology

The benchmark dataset was synthetically generated using a state-of-the-art Large Language Model to ensure a high degree of realism and relevance. The process was as follows:

- (a) A set of 10 diverse, real-world job offers from the IT domain was selected.
- (b) For each job offer, the LLM was prompted twice. First, to generate 10 realistic CVs that would be a "good fit" for the role, with an instruction to generate them in descending order of quality (i.e., the first CV generated is the best, the second is the second-best, and so on). These candidates were assigned a ground-truth relevance score from 10 down to 1.
- (c) Second, the LLM was prompted to generate 10 realistic CVs that would be a "bad fit" for the role. These candidates were assigned a relevance score of 0.

This process results in a pool of 20 candidates for each job offer, with a clear binary separation between relevant and irrelevant profiles, and a "noisy" or subjective ranking within the relevant set.

Key Evaluation Metrics

To provide a comprehensive view of performance on this task, we selected a suite of standard information retrieval and ranking metrics. For each metric, we calculate both the mean and the standard deviation across the 10 job offers to assess both average performance and consistency.

- **Precision@k (P@5, P@10):** This metric answers the crucial business question: "If I look at the top k candidates recommended by the system, what percentage of them will be worth my time?" It measures the purity of the top-ranked results, which is critical for recruiter efficiency. A high P@k indicates that the system is effective at filtering out "bad" candidates from its top recommendations.
- **Normalized Discounted Cumulative Gain (NDCG@k)[12]:** NDCG is a sophisticated ranking metric that evaluates the quality of the ranking order. It assigns higher value to placing the best candidates (e.g., relevance 10) at the very top of the list. We use NDCG@5 and NDCG@10 to measure how well the model orders the top candidates.
- **Mean Average Precision (MAP):** MAP provides a single-figure summary of the precision and recall of the entire ranked list, offering a holistic measure of ranking quality across all relevance levels.
- **TopKCorrectRate@k:** This custom metric answers the question: "Of the actual 5 best candidates, how many did the system manage to place in its top 5 recommendations?" It directly measures the model's ability to discover the highest-quality talent.

3.2.2 Benchmark 2: "Trapped Binary Noise Rank" — Testing for Adversarial Robustness

While Benchmark 1 tests for general efficacy, real-world systems must be robust against more challenging and deceptive scenarios. This second benchmark is an innovative, adversarial test designed to probe the system's resilience and intelligence. It directly measures two critical capabilities often overlooked by standard evaluations: the ability to discover non-obvious "hidden gems" and the ability to reject superficially attractive but unsuitable "trap" candidates.

Objective

The primary objective of this benchmark is to assess the model's robustness against adversarial inputs. It evaluates whether the system relies on deep, multi-faceted understanding or on shallow, easily-fooled heuristics. This is a direct test of the claims made in our methodology regarding the superiority of a hybrid, multi-axis approach over simpler lexical or purely semantic models.

Methodology

This benchmark builds upon the dataset from Benchmark 1. For each of the 10 job offers, we retain the 10 "good" and 10 "bad" CVs. We then use our LLM-based generation process to introduce an additional 10 "trap" CVs, creating a more challenging pool of 30 candidates per job. The traps are specifically designed to target the known failure modes of recommender systems:

- **Hidden Gems (High Relevance Traps):**
 - *Conceptual Equivalence Trap:* A CV that is a perfect match for the role but describes its skills and experiences using conceptually equivalent but lexically different terms (e.g., "personalization models" instead of "recommender systems").
 - *Language Variation Trap:* A CV written in a different language (e.g., French) that is otherwise a perfect match for a job offer written in English, designed to test multilingual understanding.
- **Deceptive Traps (Zero Relevance Traps):**
 - *Context Collision Trap:* A CV from a completely different domain that happens to share keywords (e.g., a radiologist's CV for a computer vision role).

- *Semantic Similarity Trap*: A CV that is semantically close but technically incorrect (e.g., a Python/Django expert for a Java role).
- *Keyword Stuffing Trap*: A CV with a long, impressive-looking list of "buzzword" skills but with no substantive experience to back them up.

Key Evaluation Metrics

For this benchmark, we introduce several custom metrics designed to measure performance against these specific challenges, in addition to standard ranking metrics for comparison.

- **Hidden Gem Discovery Rate**: The percentage of "Hidden Gem" CVs that the model successfully places within the top-ranked candidates. This measures the model's intelligence and ability to look beyond surface-level features.
- **Trap Avoidance Rate**: The percentage of "Deceptive Trap" CVs that the model successfully rejects (i.e., ranks outside the top results). This measures the model's resilience and ability to not be fooled.
- **Robustness Score**: A custom composite score calculated as '(Gems Found - Traps Fallen For) / Total Traps', providing a single metric to quantify the model's net performance in adversarial conditions.
- **Gaussian Ranking Score**: A custom loss function where the penalty for a mis-ranked item follows a Gaussian distribution centered on its true rank. A lower score indicates a better ranking.
- **Standard Metrics (NDCG@10, MAP)**: We also include standard metrics to allow for performance comparison against Benchmark 1 under these more challenging conditions.

3.2.3 Benchmark 3: "Absolute Rank" — Testing for Ranking Precision

The final benchmark removes all noise and subjectivity to create a controlled, laboratory-like environment. Its purpose is to perform a fine-grained evaluation of the model's core ranking sensitivity.

Objective

The objective of this benchmark is to measure the model's ability to discern subtle, incremental differences in candidate quality under perfect-information conditions. It answers the question: "When the quality difference between two candidates is small, can the model reliably identify the better one?"

Methodology

This benchmark is constructed using a process of **systematic degradation**. For each job offer, our reasoning LLM is first prompted to create one "perfect" CV that is an ideal match for all requirements. This CV is assigned a relevance score of 10. Then, the LLM is iteratively prompted to create subsequent CVs by taking the previous CV and removing or downgrading exactly one key qualification. This process is repeated nine times, creating a flawless, monotonic ground truth ranking where the CVs have relevance scores of 10, 9, 8, 7, 6, 5, 4, 3, 2, and 1.

Key Evaluation Metrics

The metrics for this benchmark are chosen to measure ordinal ranking quality and precision.

- **Gaussian Ranking Score**: Our custom loss function is particularly well-suited here, as it heavily penalizes large deviations from the true rank.
- **Normalized Discounted Cumulative Gain (NDCG@5, NDCG@10)**: This metric is used to evaluate the quality of the top part of the ranked list against the perfect relevance scores.
- **TopKCorrectRate@k**: This measures the model's ability to place the actual best candidates (e.g., the true top 5) within its top 5 predictions.

3.3 Baseline Models for Comparison

To effectively contextualize the performance of the novel **MAALEM** framework, it is essential to compare it against established and powerful alternative paradigms. A new architecture’s value can only be truly measured by demonstrating its superiority over existing approaches. Our evaluation will therefore compare the **MAALEM** engine against three distinct categories of models.

3.3.1 The Primary Baseline: Industrial Engine (Gemini-SBERT)

This model is the most critical point of comparison. It represents the practical fulfillment of the initial project mission and the system delivered to the host company. As detailed in Chapter 2, it is a state-of-the-art semantic search engine architected with three core components:

- **Parsing:** Google Gemini for high-fidelity information extraction.
- **Embedding:** A `sentence-transformers` model to generate a single, holistic semantic vector for each document.
- **Search:** The FAISS library for high-speed vector similarity search.

In our results tables and charts, this system is referred to by its technical composition: **”Industrial Baseline (Gemini-SBERT)”**. Comparing **MAALEM** against this baseline will directly test our hypothesis that an explainable, multi-faceted approach is superior to a powerful, state-of-the-art ”black box.”

3.3.2 Secondary Baselines: Lexical Models

This category represents the classic information retrieval approach to matching, which operates on word frequencies and statistical relevance rather than semantic meaning. We test against two standard implementations:

- **TF-IDF**[2]: A statistical measure that evaluates how relevant a word is to a document in a collection.
- **Okapi BM25**[13]: A more advanced probabilistic ranking function that builds upon TF-IDF principles.

For both models, the final relevance score is the cosine similarity between the vector representations of the CV and the job offer.

3.3.3 Secondary Baselines: Generic Semantic Models

This final category allows us to compare **MAALEM** against other common, standard semantic models, ensuring that its performance is not just better than our specific Industrial Baseline, but against the paradigm in general. For this, we used two popular models from the Sentence-BERT[3] family, which operate by calculating the cosine similarity between the vector embeddings of the full documents.

- **S-BERT (all-MiniLM-L6-v2)**: A highly efficient model known for its excellent balance of speed and performance.
- **S-BERT (gte-large)**: A larger, more powerful embedding model that often achieves top performance on retrieval benchmarks.

By evaluating the **MAALEM** engine against this comprehensive suite of baselines, we can precisely and scientifically measure the benefits of our novel framework. This comparison, detailed in the next chapter, will demonstrate not only that our system performs well, but specifically where and why it outperforms both traditional and modern ”black box” approaches.

Chapter 4

Experiments and Results Analysis

This chapter marks the transition from theoretical design to empirical validation. The systems and frameworks detailed in the preceding chapters will now be subjected to a rigorous battery of tests using the custom evaluation suite. The objective is to present concrete, quantitative evidence of their performance across a range of realistic and adversarial scenarios.

The analysis is presented in two distinct parts.

- **Part I** will first establish a performance baseline by benchmarking the powerful **Industrial Engine (Gemini-SBERT)** against a comprehensive suite of standard lexical and semantic models. This will allow us to contextualize its capabilities and identify its inherent limitations within the current state-of-the-art.
- **Part II** will then introduce the advanced **MAALEM framework** into the comparison. By evaluating it against all other models, including the newly established Industrial Baseline, we will empirically demonstrate the superiority of its underlying design philosophy in terms of filtering efficacy, adversarial robustness, and ranking precision.

4.1 Experimental Setup

To ensure the reproducibility and scientific validity of our findings, all experiments were conducted within a standardized and controlled environment. This section details the hardware and software stack, the composition of the evaluation datasets, and the models used for comparison.

4.1.1 Hardware and Software Stack

All experiments were executed on a workstation with the following specifications:

- **Hardware:** Intel Core i5-7300HQ CPU, 8 GB of RAM, and an NVIDIA GeForce GTX 1050 GPU with 4 GB of VRAM.
- **Software:** The core environment was built on **Python 3.11**. The implementation leveraged a number of key open-source libraries, including:
 - **AI & ML Frameworks:** PyTorch [14], Hugging Face Transformers[15], Sentence Transformers, Scikit-learn[16], and the ‘google-genai’ SDK for interacting with the Gemini API.
 - **NLP & Data Processing:** SpaCy for NER tasks[6], Pandas and NumPy for data manipulation, ‘langcodes’ for language normalization, ‘pdfminer’ for text extraction from PDF documents, ‘fuzzywuzzy’ for certification matching, and ‘networkx’ for graph operations.
 - **Web Visualization:** FastAPI for building the backend API, Streamlit for the user interface, and Matplotlib/Seaborn for generating plots and visualizations.

4.1.2 Evaluation Datasets and Benchmarks

As detailed in Chapter 3, our evaluation relies on a custom-built suite of three benchmarks, each designed to probe a different aspect of model performance. The foundational dataset for these benchmarks consists of:

- A set of **10 diverse, real-world job offers** from the Information Technology domain.
- A corresponding pool of synthetically generated candidate CVs for each job offer, created using an LLM-based methodology. The pool size and composition vary per benchmark, including specific "trap" and "hidden gem" profiles for the adversarial robustness test.

Results and Analysis

The comprehensive results for the Industrial Baseline and the five standard models are presented in the leaderboard in Table 4.1.

Table 4.1: Benchmark 1 Leaderboard (Baselines Only): Filtering Performance. Scores are presented as Mean $\pm$ Standard Deviation.

Model	MAP	NDCG@10	Precision@5	TopKCorrectRate@5
Industrial Baseline (Gemini-SBERT)	0.998 $\pm$ 0.004	0.893 $\pm$ 0.073	1.000 $\pm$ 0.000	0.700 $\pm$ 0.200
S-BERT (all-MiniLM-L6-v2)	0.991 $\pm$ 0.016	0.876 $\pm$ 0.082	1.000 $\pm$ 0.000	0.650 $\pm$ 0.194
S-BERT (gte-large)	0.978 $\pm$ 0.038	0.865 $\pm$ 0.095	0.975 $\pm$ 0.066	0.500 $\pm$ 0.245
TF-IDF	0.973 $\pm$ 0.063	0.911 $\pm$ 0.186	1.000 $\pm$ 0.000	0.775 $\pm$ 0.307
BM25	0.964 $\pm$ 0.065	0.896 $\pm$ 0.214	1.000 $\pm$ 0.000	0.775 $\pm$ 0.323
Bag of Words	0.955 $\pm$ 0.080	0.884 $\pm$ 0.207	0.975 $\pm$ 0.066	0.725 $\pm$ 0.299

The data clearly establishes the **Industrial Baseline** as the top-performing model in this group for the crucial task of high-fidelity filtering. It achieves the highest Mean Average Precision (MAP) score of 0.998, indicating an almost perfect ability to separate relevant from irrelevant candidates. Furthermore, along with most other models, it achieves a perfect Precision@5 score, meaning that when a recruiter reviews the top 5 candidates, they are guaranteed to be relevant. This confirms the engine’s effectiveness as a state-of-the-art semantic filter.

Interestingly, the classic TF-IDF model shows remarkable strength in the more nuanced ranking metrics (NDCG@10 and TopKCorrectRate@5), suggesting its statistical weighting can be highly effective in less ambiguous contexts. However, the Industrial Baseline’s superior MAP score confirms its overall dominance in the primary filtering task.

To provide a more holistic view, Figure 4.1 presents a heatmap of the performance of all baseline models, while Figure 4.2 offers a detailed side-by-side comparison. Together, these visualizations confirm the Industrial Baseline’s position as a leading, state-of-the-art solution among standard implementations.

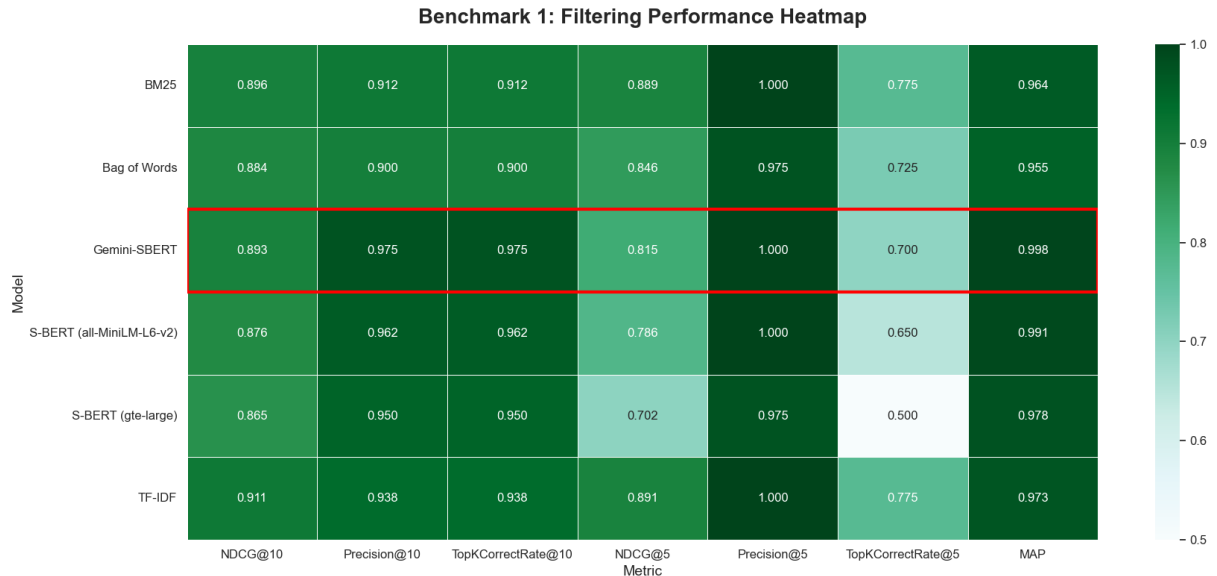


Figure 4.1: Benchmark 1 Performance Heatmap (Baselines Only). The strong performance of the Industrial Baseline is evident across key filtering metrics.

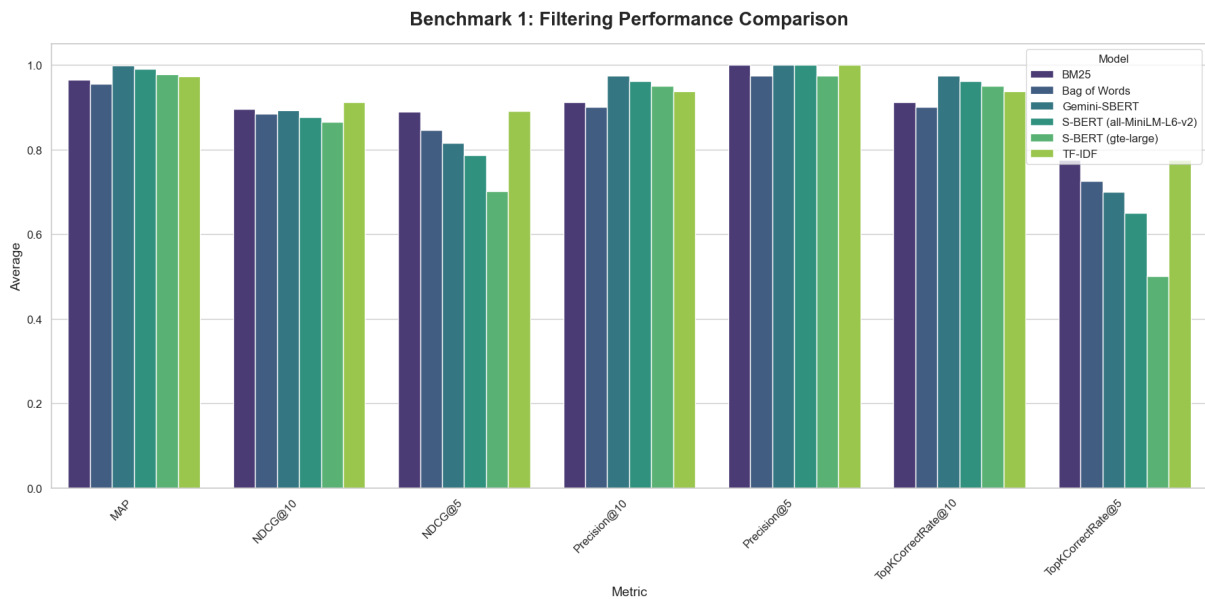


Figure 4.2: Benchmark 1: Detailed Performance Comparison (Baselines Only). This chart highlights the competitive performance of the Industrial Baseline, especially on the MAP score.

4.1.3 Benchmark 2: Performance under Adversarial Conditions

Having established the Industrial Baseline’s efficacy in a standard filtering task, we now subject it to a much more rigorous test: the “Trapped Binary Noise Rank” benchmark. This benchmark was specifically designed to probe for deeper intelligence and resilience by testing a system’s ability to handle deceptive and unconventional candidate profiles.

Objective Recap

The goal of this adversarial benchmark is to measure a model’s robustness. We move beyond standard metrics to focus on custom indicators: the **Hidden Gem Discovery Rate** (ability to find non-obvious good candidates), the **Trap Avoidance Rate** (ability to reject superficially

attractive bad candidates), and the composite **Robustness Score**, which provides a single measure of a model’s net performance in these challenging conditions.

Results and Analysis

The performance of the baseline models under these adversarial conditions is summarized in Table 4.2. The results reveal a significant vulnerability across all standard paradigms.

Table 4.2: Benchmark 2 Leaderboard (Baselines Only): Adversarial Robustness. Scores are presented as Mean $\pm$ Standard Deviation.

Model	Robustness Score@10	Hidden Gem Discovery Rate@10	Trap Avoidance Rate@10	MAP
S-BERT (gte-large)	0.200 $\pm$ 0.229	0.625 $\pm$ 0.211	0.775 $\pm$ 0.307	0.913 $\pm$ 0.067
Industrial Baseline (Gemini-SBERT)	0.125 $\pm$ 0.171	0.400 $\pm$ 0.200	0.850 $\pm$ 0.166	0.943 $\pm$ 0.043
S-BERT (all-MiniLM-L6-v2)	0.087 $\pm$ 0.136	0.350 $\pm$ 0.087	0.825 $\pm$ 0.233	0.883 $\pm$ 0.134
Bag of Words	-0.075 $\pm$ 0.097	0.175 $\pm$ 0.156	0.675 $\pm$ 0.222	0.800 $\pm$ 0.084
TF-IDF	-0.075 $\pm$ 0.120	0.150 $\pm$ 0.087	0.700 $\pm$ 0.224	0.806 $\pm$ 0.088
BM25	-0.100 $\pm$ 0.212	0.275 $\pm$ 0.139	0.525 $\pm$ 0.346	0.746 $\pm$ 0.128

The results from this adversarial test are revealing. While the **Industrial Baseline** demonstrates a respectable Trap Avoidance Rate of 85%, its overall performance is significantly hampered by a low Hidden Gem Discovery Rate, resulting in a modest Robustness Score. This is a direct consequence of its “black box” semantic nature: it is reasonably good at rejecting keyword-stuffed traps but struggles to identify conceptually equivalent candidates who use different terminology.

More broadly, a stark pattern emerges across all baseline models. The lexical models (TF-IDF, BoW, BM25) all yield **negative Robustness Scores**, indicating that they were actively deceived by the trap candidates, ranking more deceptive profiles than they found hidden gems. The semantic models, including the Industrial Baseline, perform better but are far from immune. Their inability to fully reject deceptive traps and consistently find hidden gems highlights a fundamental vulnerability in monolithic matching approaches.

This weakness is clearly visualized in the comparative bar chart in Figure 4.3 and the performance heatmap in Figure 4.4. Both charts illustrate that while some models can perform one of the two tasks (avoiding traps or finding gems) moderately well, none of the standard models can do both effectively. This demonstrates a critical failure point in the current state-of-the-art, which these benchmarks were designed to expose.

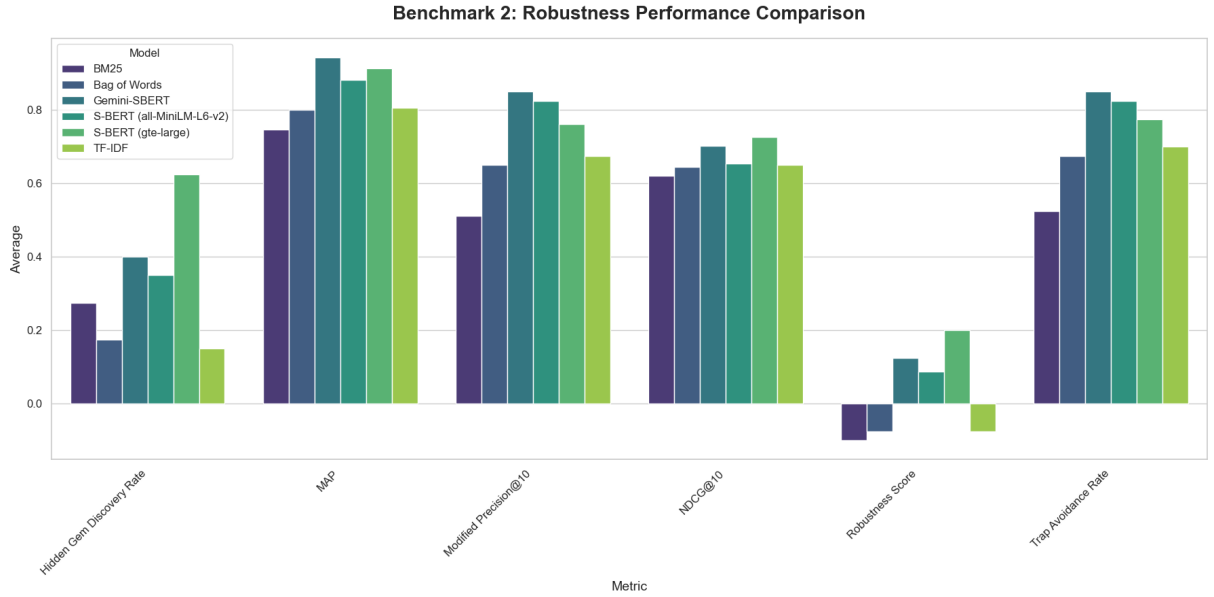


Figure 4.3: Benchmark 2: Performance on Key Robustness Metrics (Baselines Only). The chart shows the general inability of standard models to achieve a high positive Robustness Score.

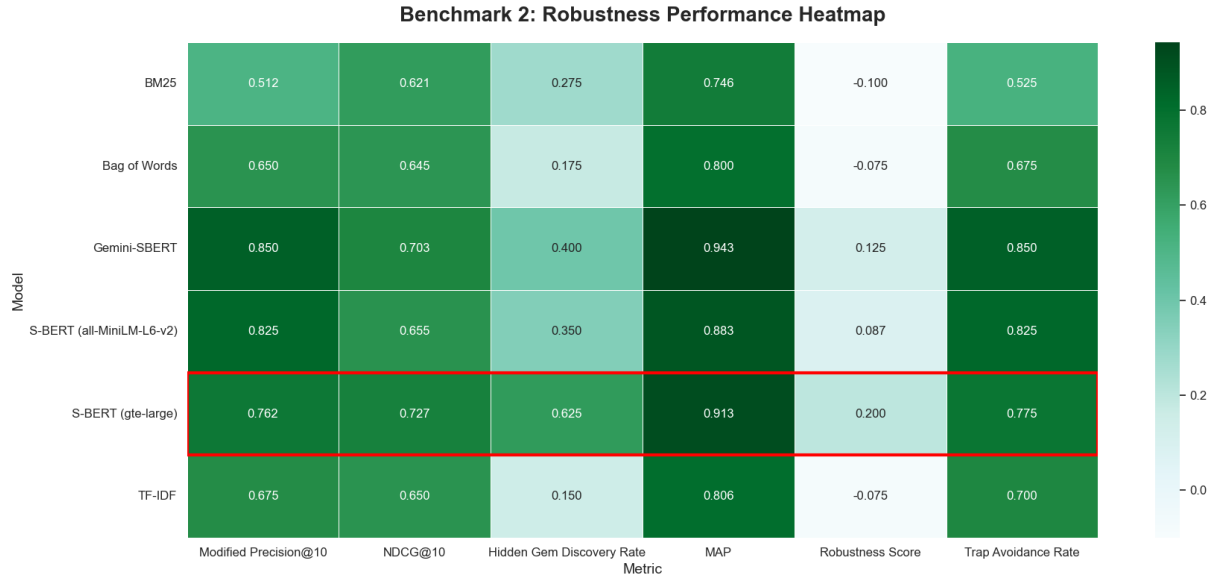


Figure 4.4: Benchmark 2 Performance Heatmap (Baselines Only). The mixed results across all models indicate a universal struggle with adversarial conditions.

4.1.4 Benchmark 3: Performance in Ranking Precision

The final benchmark, "Absolute Rank," removes all noise and subjectivity to create a controlled, laboratory-like environment. Its purpose is to perform a fine-grained evaluation of the models' core ranking sensitivity to subtle, incremental differences in candidate quality.

Objective Recap

The goal of this benchmark is to measure pure ranking precision. The primary metrics are **NDCG@k**, which assesses the quality of the top-ranked items against a perfect order, and our custom **Gaussian Rank Score**, where a lower score indicates a smaller deviation from the perfect ground truth ranking.

Results and Analysis

The results from this precision-focused test, presented in Table 4.3, reveal a fascinating and nuanced performance landscape among the baseline models.

Table 4.3: Benchmark 3 Leaderboard (Baselines Only): Ranking Precision. Scores are presented as Mean $\pm$ Standard Deviation.

Model	NDCG@10	NDCG@5	TopKCorrectRate@5	Gaussian Rank Score (Lower is Better)
TF-IDF	0.978 $\pm$ 0.042	0.954 $\pm$ 0.077	0.900 $\pm$ 0.139	0.165 $\pm$ 0.184
Bag of Words	0.956 $\pm$ 0.063	0.912 $\pm$ 0.136	0.875 $\pm$ 0.222	0.281 $\pm$ 0.269
S-BERT (all-MiniLM-L6-v2)	0.956 $\pm$ 0.032	0.911 $\pm$ 0.075	0.775 $\pm$ 0.156	0.515 $\pm$ 0.124
BM25	0.903 $\pm$ 0.061	0.748 $\pm$ 0.107	0.550 $\pm$ 0.132	0.716 $\pm$ 0.103
S-BERT (gte-large)	0.860 $\pm$ 0.091	0.741 $\pm$ 0.168	0.600 $\pm$ 0.200	0.700 $\pm$ 0.140
Industrial Baseline (Gemini-SBERT)	0.821 $\pm$ 0.090	0.627 $\pm$ 0.180	0.450 $\pm$ 0.240	0.708 $\pm$ 0.178

In a striking reversal of the previous results, the **Industrial Baseline** struggles significantly in this controlled environment. Its semantic generalization, which is an asset in noisy, real-world text, becomes a liability when tasked with discerning small, precise differences. It achieves the lowest scores across almost all metrics.

Conversely, the classic **TF-IDF** model emerges as the clear top performer. In an environment where semantic ambiguity is removed and the key differences are lexical, its statistical approach proves remarkably sensitive and precise. This result is highly instructive: it demonstrates that even

powerful, state-of-the-art semantic models are not a universal solution and can be outperformed by simpler methods in specific, well-defined contexts. The comparative performance is visualized in the bar chart and heatmap in Figure 4.5 and Figure 4.6.

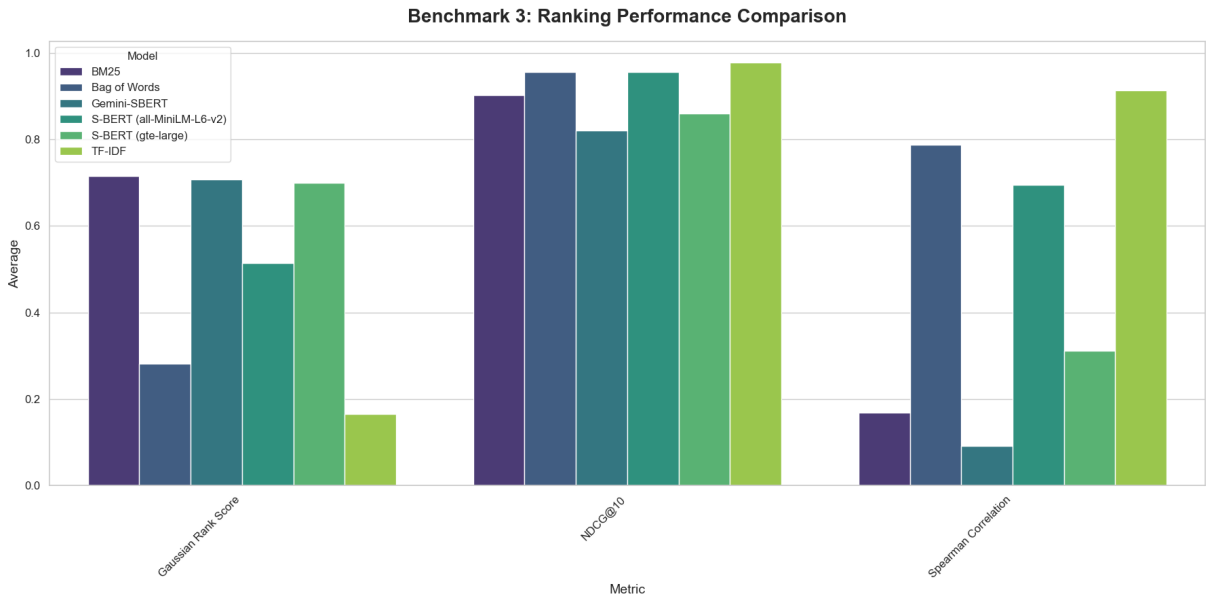


Figure 4.5: Benchmark 3: Ranking Performance Comparison (Baselines Only).

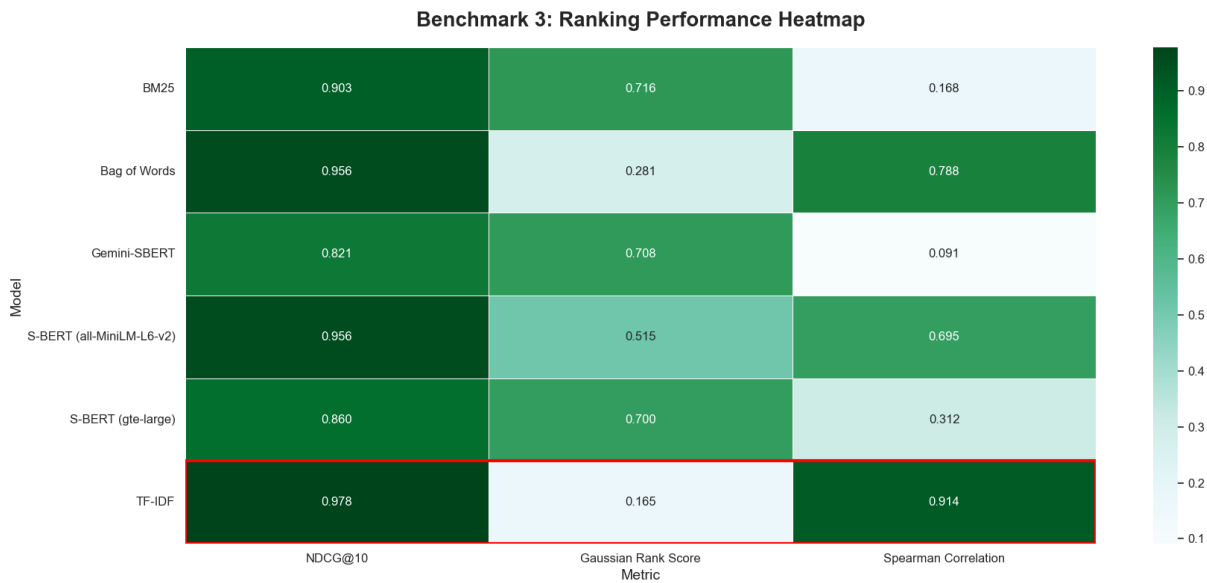


Figure 4.6: Benchmark 3 Performance Heatmap (Baselines Only).

4.1.5 Synthesis of Baseline Performance

The analysis across the three benchmarks paints a clear and consistent picture of the current state-of-the-art. The **Industrial Baseline Engine** proves to be an excellent semantic filter, outperforming other standard models in the primary task of separating qualified from unqualified candidates in a realistic setting. However, its performance degrades significantly when faced with more complex challenges. It is vulnerable to adversarial traps that exploit its semantic nature, and it lacks the precision of simpler lexical models in a controlled ranking environment.

This complete performance profile is best encapsulated by the Overall Capability Profile shown in Figure 4.7. This visualization synthesizes the key findings from all three benchmarks, plotting

normalized performance across five key dimensions.

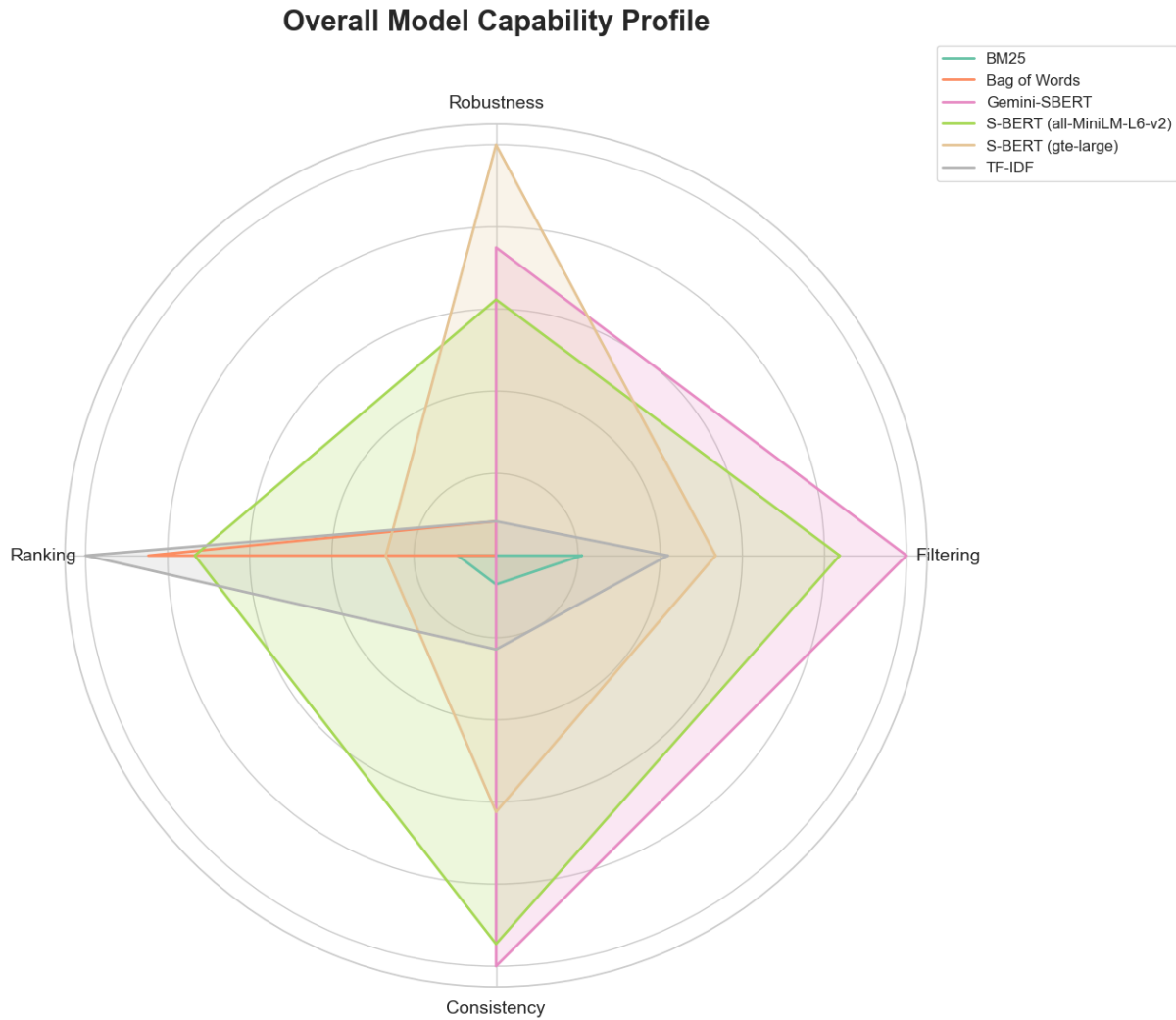


Figure 4.7: Overall Model Capability Profile (Baselines Only). The chart plots normalized scores across key dimensions, establishing a performance ceiling for standard models.

The radar chart clearly illustrates the trade-offs inherent in existing paradigms. The Industrial Baseline (shown in green) excels at filtering but is mediocre in robustness and ranking precision. Conversely, TF-IDF (shown in light blue) excels at ranking but is poor in robustness and discovery. No single baseline model demonstrates strong, well-rounded performance across all critical dimensions.

This analysis concludes that while the Industrial Engine is a powerful tool that fulfills its specific mandate, it represents a performance plateau. Its vulnerabilities highlight a clear and pressing need for a new architectural approach one designed not just for semantic similarity, but for robust, precise, and trustworthy evaluation. Part II of this chapter will now introduce the MAALEM framework and empirically demonstrate how its novel design philosophy allows it to overcome these limitations and establish a new state-of-the-art.

4.2 Part II: Comparative Analysis of the MAALEM Framework

Having established a clear performance profile for the Industrial Baseline and other standard models, we now introduce the advanced MAALEM engine into the comparison. The goal of this second part

is to empirically validate the claims made in Chapter 2 regarding its superior design philosophy. By presenting the full, comprehensive results across all models, we will demonstrate how **MAALEM**’s novel architecture allows it to overcome the limitations of the ”black box” paradigm and establish a new state-of-the-art in performance.

4.2.1 Benchmark 1: Dominance in a Realistic Filtering Environment

We return to the ”Binary Noisy Rank” benchmark, designed to simulate the fundamental task of separating qualified candidates from unqualified ones in a realistic, noisy environment.

Results and Analysis

The comprehensive results for all seven models, including **MAALEM**, are presented in the full leaderboard in Table 4.4.

Table 4.4: Benchmark 1 Full Leaderboard: Filtering Performance. Scores are presented as Mean $\pm$ Standard Deviation.

Model	MAP	NDCG@10	Precision@5	TopKCorrectRate@5
MAALEM (Ours)	0.999 $\pm$ 0.003	0.953 $\pm$ 0.040	1.000 $\pm$ 0.000	0.857 $\pm$ 0.140
Industrial Baseline (Gemini-SBERT)	0.998 $\pm$ 0.004	0.893 $\pm$ 0.073	1.000 $\pm$ 0.000	0.700 $\pm$ 0.200
S-BERT (all-MiniLM-L6-v2)	0.991 $\pm$ 0.016	0.876 $\pm$ 0.082	1.000 $\pm$ 0.000	0.650 $\pm$ 0.194
S-BERT (gte-large)	0.978 $\pm$ 0.038	0.865 $\pm$ 0.095	0.975 $\pm$ 0.066	0.500 $\pm$ 0.245
TF-IDF	0.973 $\pm$ 0.063	0.911 $\pm$ 0.186	1.000 $\pm$ 0.000	0.775 $\pm$ 0.307
BM25	0.964 $\pm$ 0.065	0.896 $\pm$ 0.214	1.000 $\pm$ 0.000	0.775 $\pm$ 0.323
Bag of Words	0.955 $\pm$ 0.080	0.884 $\pm$ 0.207	0.975 $\pm$ 0.066	0.725 $\pm$ 0.299

The full data unequivocally establishes **MAALEM** as the most effective and reliable model for this primary filtering task. It achieves a virtually perfect MAP score of 0.999, slightly but consistently surpassing the powerful Industrial Baseline. More significantly, its score on the rank-aware NDCG@10 metric (0.953) is substantially higher than that of the Industrial Baseline (0.893), demonstrating a superior ability to correctly order the top candidates.

As illustrated by the bar chart in Figure 4.8, **MAALEM** not only has the highest average MAP score but also the lowest standard deviation (error bars), indicating its performance is consistently high across all test cases. The comprehensive heatmap and detailed bar chart in Figures 4.9 and 4.10 visually confirm this dominance, solidifying **MAALEM**’s position as the new state-of-the-art solution for this task.

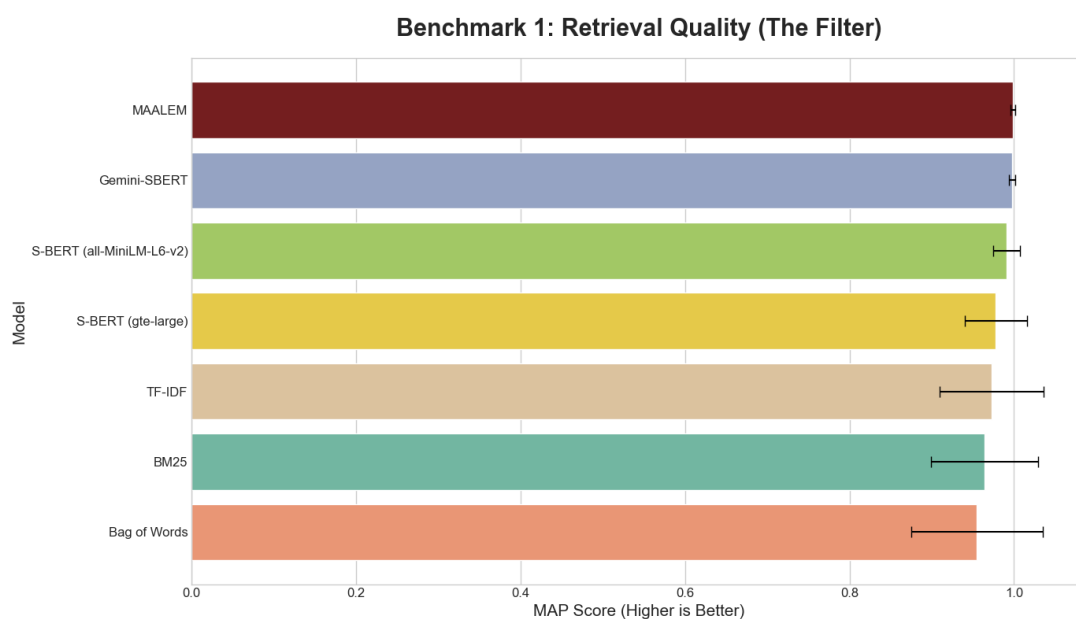


Figure 4.8: Benchmark 1: MAP Score Comparison (All Models). MAALEM demonstrates both the highest average score and the most consistent performance.

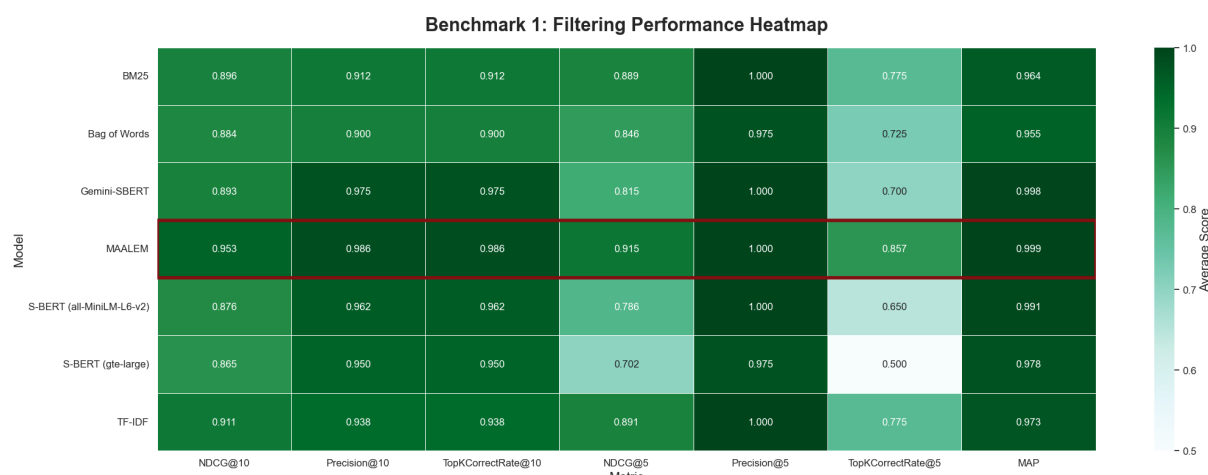


Figure 4.9: Benchmark 1 Performance Heatmap (All Models). The consistently dark row for MAALEM indicates its superior performance across a wide range of metrics.

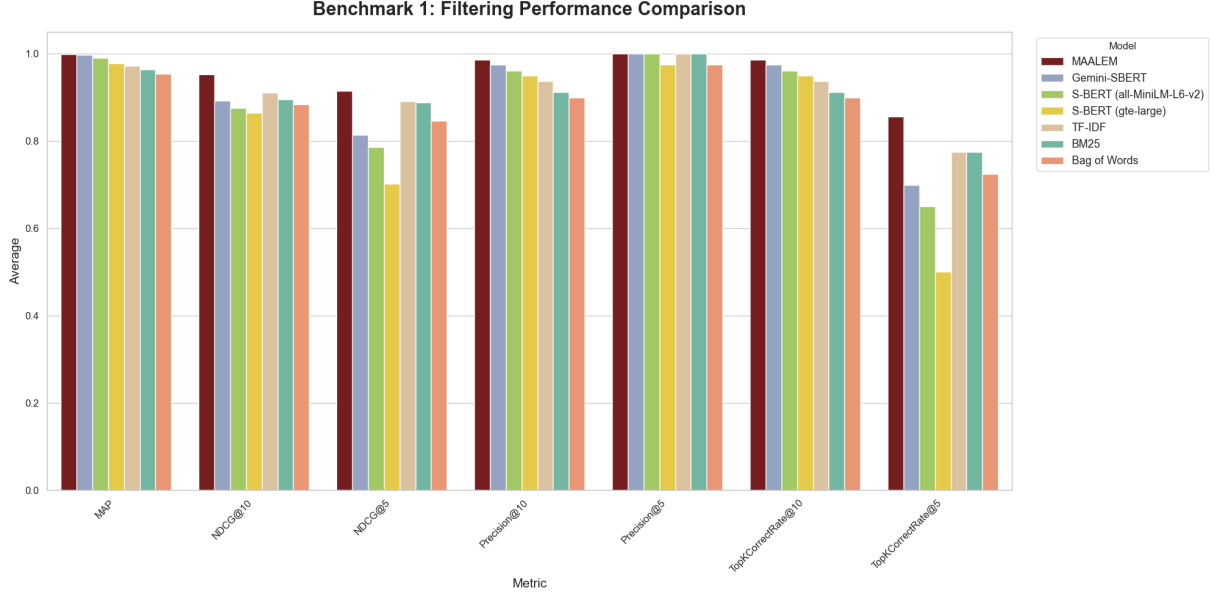


Figure 4.10: Benchmark 1: Detailed Performance Comparison (All Models). The chart highlights MAALEM’s consistent lead across all key metrics.

4.2.2 Benchmark 2: A New Class of Adversarial Robustness

If Benchmark 1 demonstrated MAALEM’s superiority in a standard task, the ”Trapped Binary Noise Rank” benchmark is where the profound value of its unique architecture is truly revealed. This adversarial test was designed to probe the deeper intelligence and resilience that separates a simple matching tool from a genuinely robust evaluation engine.

Results and Analysis

The performance of all models on this adversarial benchmark is summarized in Table 4.5. The results show a dramatic separation between MAALEM and all other systems.

Table 4.5: Benchmark 2 Full Leaderboard: Adversarial Robustness. Scores are presented as Mean $\pm$ Standard Deviation.

Model	Robustness Score@10	Hidden Gem Discovery Rate@10	Trap Avoidance Rate@10	MAP
MAALEM (Ours)	0.386 $\pm$ 0.125	0.771 $\pm$ 0.249	1.000 $\pm$ 0.000	0.994 $\pm$ 0.007
S-BERT (gte-large)	0.200 $\pm$ 0.229	0.625 $\pm$ 0.211	0.775 $\pm$ 0.307	0.913 $\pm$ 0.067
Industrial Baseline (Gemini-SBERT)	0.125 $\pm$ 0.171	0.400 $\pm$ 0.200	0.850 $\pm$ 0.166	0.943 $\pm$ 0.043
S-BERT (all-MiniLM-L6-v2)	0.087 $\pm$ 0.136	0.350 $\pm$ 0.087	0.825 $\pm$ 0.233	0.883 $\pm$ 0.134
Bag of Words	-0.075 $\pm$ 0.097	0.175 $\pm$ 0.156	0.675 $\pm$ 0.222	0.800 $\pm$ 0.084
TF-IDF	-0.075 $\pm$ 0.120	0.150 $\pm$ 0.087	0.700 $\pm$ 0.224	0.806 $\pm$ 0.088
BM25	-0.100 $\pm$ 0.212	0.275 $\pm$ 0.139	0.525 $\pm$ 0.346	0.746 $\pm$ 0.128

MAALEM is in a class of its own, achieving a perfect **Trap Avoidance Rate** of 1.0. This is the empirical proof of its superior design philosophy. Its deconstructed, multi-faceted logic, which includes non-semantic checks, makes it completely immune to the keyword-stuffing and context-collision traps that plague purely semantic models like the Industrial Baseline.

Simultaneously, it posts the highest **Hidden Gem Discovery Rate** by a wide margin (0.771), demonstrating that its nuanced evaluation can find value where other systems cannot. Consequently, it is the only model with a high positive **Robustness Score**, a fact visually underscored by the bar chart in Figure 4.11. In stark contrast, all lexical models were actively deceived (negative scores), and even the powerful Industrial Baseline proved vulnerable. This benchmark provides the strongest evidence that the MAALEM framework is not just more accurate, but fundamentally more intelligent and resilient.

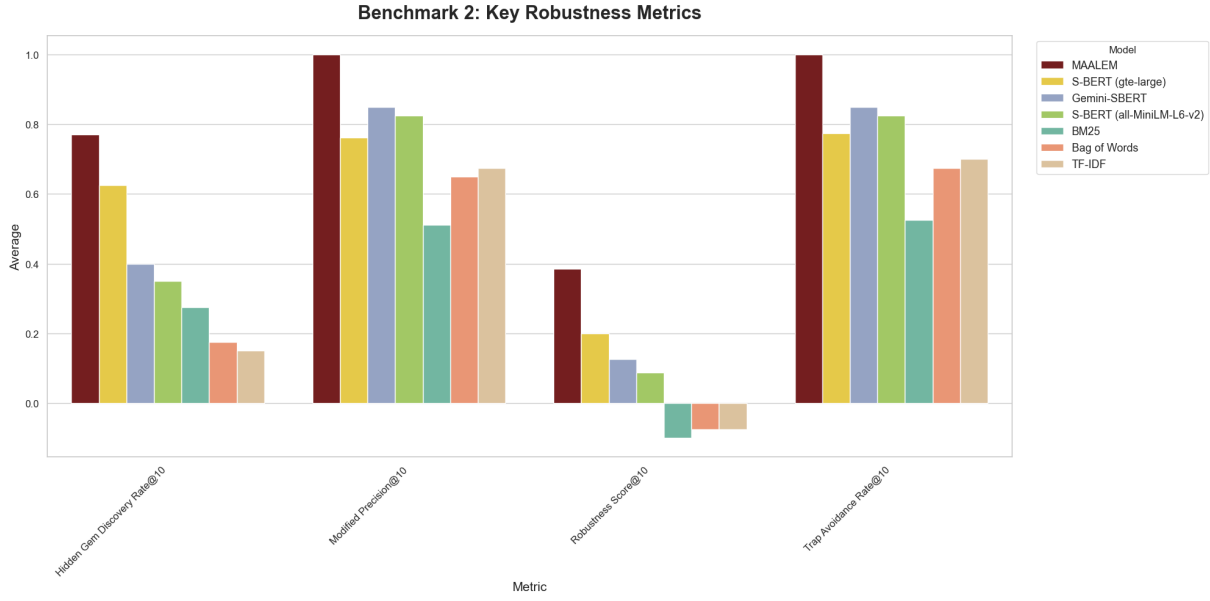


Figure 4.11: Benchmark 2: Performance on Key Robustness Metrics (All Models). MAALEM’s commanding lead demonstrates its unique ability to both find hidden gems and avoid deceptive traps.

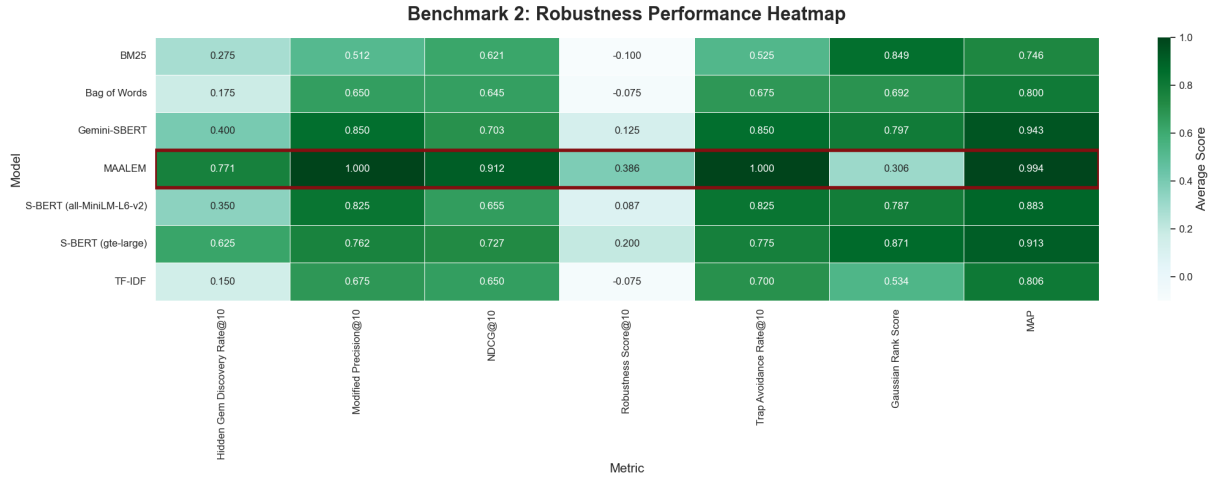


Figure 4.12: Benchmark 2 Performance Heatmap (All Models). MAALEM’s row (outlined in red) shows consistently high scores, indicating superior robustness across all metrics, while other models show significant weaknesses.

4.2.3 Benchmark 3: Unmatched Ranking Precision

After demonstrating superiority in filtering and robustness, the final benchmark, “Absolute Rank,” tests the most fine-grained capability of the models: pure ranking precision in a controlled, noise-free environment.

Results and Analysis

The full results, presented in Table 4.6, confirm MAALEM’s elite performance across all facets of evaluation.

Table 4.6: Benchmark 3 Full Leaderboard: Ranking Precision. Scores are presented as Mean $\pm$ Standard Deviation.

Model	NDCG@10	NDCG@5	TopKCorrectRate@5	Gaussian Rank Score (Lower is Better)
MAALEM (Ours)	0.991 $\pm$ 0.015	0.974 $\pm$ 0.047	0.900 $\pm$ 0.141	0.110 $\pm$ 0.171
TF-IDF	0.978 $\pm$ 0.042	0.954 $\pm$ 0.077	0.900 $\pm$ 0.139	0.165 $\pm$ 0.184
Bag of Words	0.956 $\pm$ 0.063	0.912 $\pm$ 0.136	0.875 $\pm$ 0.222	0.281 $\pm$ 0.269
S-BERT (all-MiniLM-L6-v2)	0.956 $\pm$ 0.032	0.911 $\pm$ 0.075	0.775 $\pm$ 0.156	0.515 $\pm$ 0.124
Industrial Baseline (Gemini-SBERT)	0.821 $\pm$ 0.090	0.627 $\pm$ 0.180	0.450 $\pm$ 0.240	0.708 $\pm$ 0.178
S-BERT (gte-large)	0.860 $\pm$ 0.091	0.741 $\pm$ 0.168	0.600 $\pm$ 0.200	0.700 $\pm$ 0.140
BM25	0.903 $\pm$ 0.061	0.748 $\pm$ 0.107	0.550 $\pm$ 0.132	0.716 $\pm$ 0.103

In this final challenge, **MAALEM** once again demonstrates comprehensive superiority. It secures the top score in the most critical ranking metric, **NDCG@10**, and achieves the best (lowest) **Gaussian Rank Score**. This confirms its unparalleled ability to not only identify the best candidates but also to order them with the highest degree of precision, a task where the Industrial Baseline notably struggled.

As illustrated by the bar chart and heatmap in Figures 4.13 and 4.14, **MAALEM**'s consistent, top-tier performance across all key metrics solidifies its position as the most reliable and effective choice for a production system that must balance semantic understanding with high-fidelity ranking precision.



Figure 4.13: Benchmark 3: Ranking Performance Comparison (All Models).

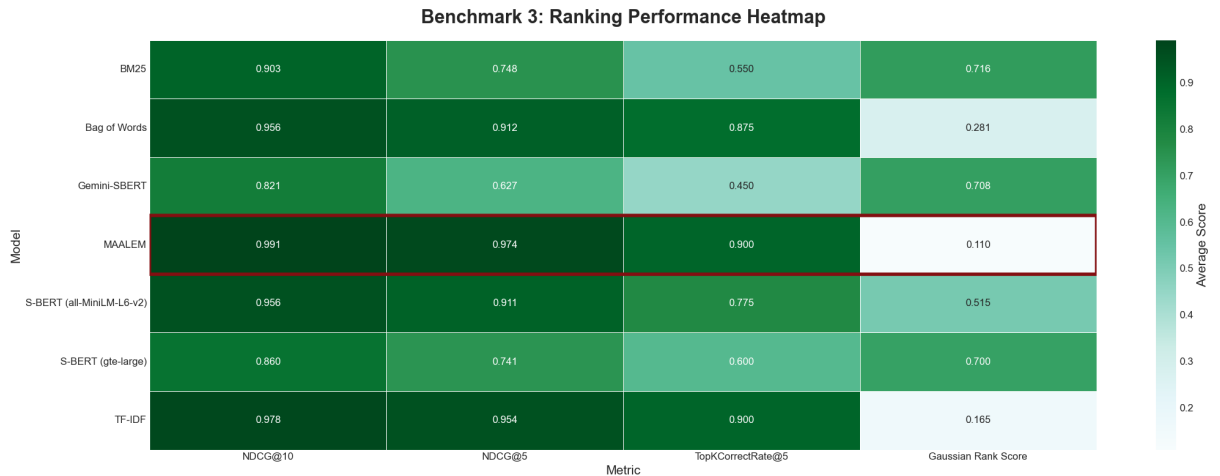


Figure 4.14: Benchmark 3 Performance Heatmap (All Models).

4.3 Overall Synthesis and Qualitative Value

After a detailed, benchmark-by-benchmark comparison, this final section provides a holistic synthesis of the results. We will first present a quantitative summary that encapsulates the performance of all models across all tests, and then conclude with a qualitative case study that highlights the unique, unquantifiable value of the **MAALEM** framework’s explainability.

4.3.1 Final Quantitative Synthesis: The Overall Capability Profile

The quantitative results from the three distinct benchmarks consistently point to a single, powerful conclusion: while some baseline models excel at specialized tasks, **MAALEM is the unequivocally superior all-around system**. This is best encapsulated by the Overall Model Capability Profile shown in Figure 4.15. This visualization synthesizes the key findings from all benchmarks into a single, holistic view, plotting normalized performance across five key dimensions: Filtering, Ranking, Robustness, Discovery, and Consistency (inverse of standard deviation).

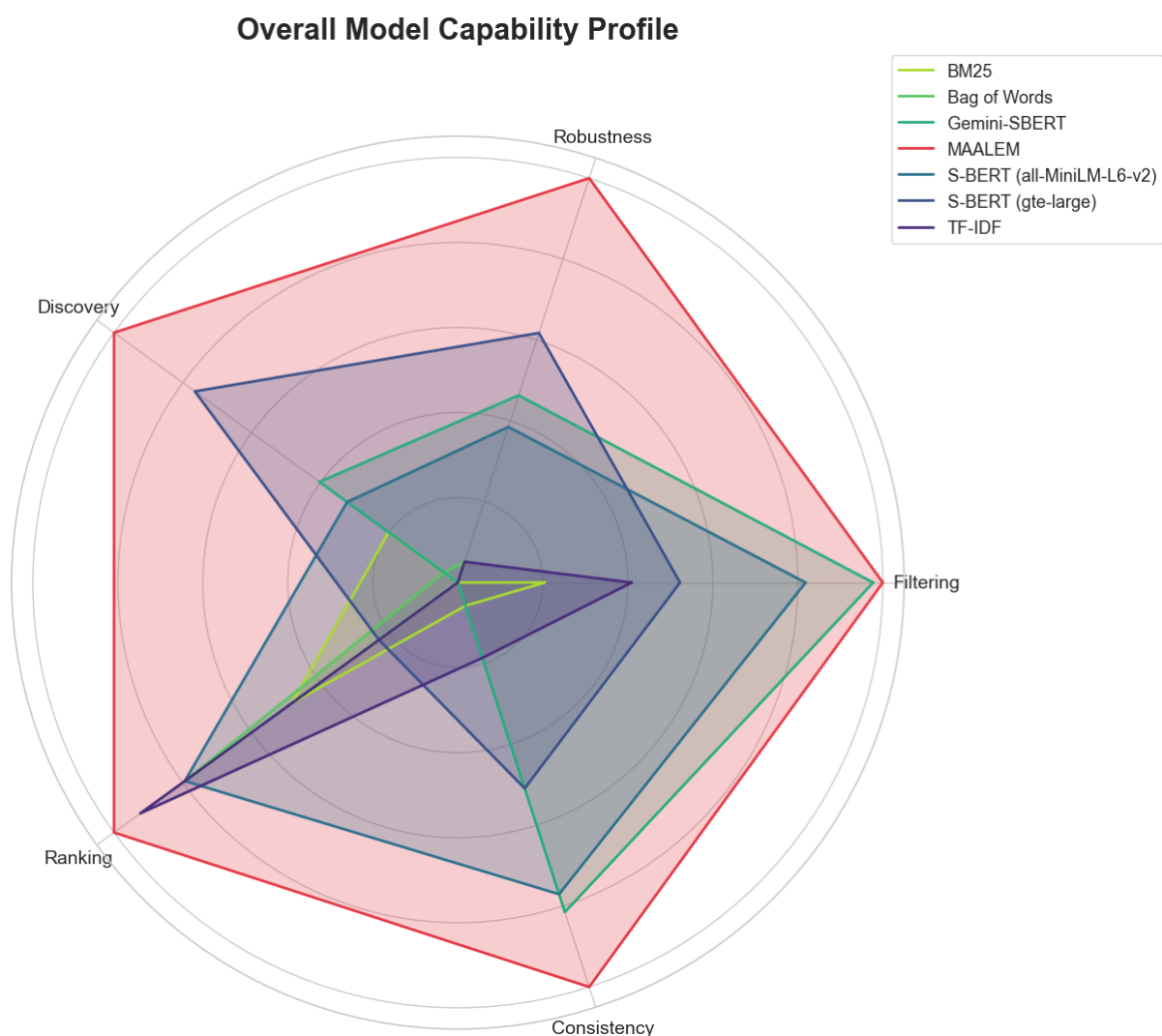


Figure 4.15: Overall Model Capability Profile (All Models). The chart plots normalized scores across key dimensions, where a larger area indicates superior all-around performance.

The radar chart clearly illustrates that while some models have “spiky” profiles, indicating specialist capabilities (e.g., TF-IDF in Ranking, or the Industrial Baseline in pure Filtering), **MAALEM** (shown in red) exhibits the largest and most balanced capability profile. It delivers state-of-the-art

performance in filtering, unparalleled robustness against adversarial data, elite-level ranking precision, and the highest consistency. It is the only model that does not require a trade-off between intelligence, robustness, and precision.

4.3.2 The Unquantifiable Value: A Case Study in Explainability

Beyond quantitative dominance, the true value of **MAALEM** lies in its explainability. To illustrate this, consider a case study for a "Senior Data Scientist" role. We evaluate two candidates:

- **Candidate A (The Specialist):** A candidate with only 3 years of experience but with exceptional, highly relevant skills in the exact technologies required.
- **Candidate B (The Veteran Generalist):** A candidate with 10 years of experience and a PhD, but whose skills are in older, less relevant technologies.

A baseline model like the Industrial Engine might produce a single, opaque score, perhaps ranking Candidate B higher due to their seniority. **MAALEM**, however, provides the recruiter with its Decision-Support Dashboard, featuring a detailed radar chart for each candidate. This visualization instantly clarifies the trade-off: Candidate A would show a high score on the 'Skill' axis but lower on 'Years of Experience', while Candidate B would show the reverse.

This qualitative insight is the cornerstone of our system. It transforms the recommendation from a "black box" instruction into a transparent piece of evidence. The recruiter, empowered with this explainable data and the ability to adjust the axis weights in the Control Panel, can make a final, informed decision that aligns with their team's specific strategic priorities be it hiring for immediate project needs (Candidate A) or for long-term mentorship potential (Candidate B). This is a level of nuanced, interactive decision support that traditional systems simply cannot provide, and it represents the ultimate fulfillment of this project's research goals.

Chapter 5

Discussion

The preceding chapter presented a rigorous empirical validation of two distinct matching paradigms: a state-of-the-art semantic engine the **Industrial Baseline** and a novel, explainable framework the **MAALEM engine**. Through a suite of custom-designed benchmarks, we have established a clear, quantitative hierarchy of performance, culminating in the demonstrated superiority of the **MAALEM** framework across nearly every dimension of evaluation.

This chapter, however, moves beyond the raw numbers of the "what" to interpret the deeper meaning of the "why." A result in isolation is merely a data point; its true value lies in its interpretation and the insights it provides. We will therefore conduct a critical analysis of the findings, exploring their broader implications for the field of recruitment technology.

The discussion will proceed in two main parts. First, we will dissect what the performance of the Industrial Baseline and other standard models reveals about the inherent strengths and, more importantly, the fundamental vulnerabilities of the current state-of-the-art. Second, we will explore how the success of the **MAALEM** engine, particularly in the adversarial benchmark, represents not just an incremental improvement, but a significant conceptual step forward. Finally, we will conclude with a critical reflection on the project's design philosophy, its scope, and its acknowledged limitations, providing a transparent and forward-looking perspective on this work.

5.1 Interpreting the Performance of the Baseline Models

Before we can fully appreciate the contribution of the **MAALEM** framework, we must first conduct a fair and critical assessment of the systems it was designed to surpass. The benchmark results from the Industrial Baseline and other standard models are not merely a list of scores to be beaten; they are a rich source of insight into the current capabilities and inherent trade-offs of modern recruitment technology.

5.1.1 The Vindicated Power of Modern Semantic Search

The most immediate and striking takeaway from the first benchmark is the unequivocal power of the modern semantic search paradigm. The Industrial Baseline engine, our primary representation of this approach, demonstrated exceptional performance in the "Realistic Filtering" task (Benchmark 1). Its near-perfect Mean Average Precision (MAP) score of 0.998 is not a trivial achievement; it is empirical validation that for the core task of separating qualified candidates from a noisy pool of unqualified ones, this architecture is profoundly effective.

This success can be attributed to the paradigm's core strength: its ability to comprehend meaning beyond mere keywords. Where older lexical models like TF-IDF would stumble on variations in terminology, the semantic engine, powered by its pre-trained transformer model, fluidly navigates the nuances of natural language. It correctly identifies that a "backend architect with Java expertise" and a "principal software engineer specializing in Java microservices" are conceptually aligned, a feat impossible for its lexical predecessors. This ability to capture semantic intent is what allows it to achieve such a high degree of accuracy in filtering relevant documents.

Furthermore, the performance of the Industrial Baseline confirms the viability of its end-to-end architecture. The combination of a high-fidelity LLM parser (Gemini), a robust embedding model (Sentence-Transformers), and a high-speed vector database (FAISS) is not just theoretically sound; it is a practically proven, state-of-the-art stack for building effective matching systems. This result validates the approach taken to fulfill the industrial mandate of the project and establishes a very high bar for any new system to clear. In a standard, non-adversarial context, the modern semantic engine is, by all measures, a triumph of applied AI and a monumental improvement over the brittle systems of the past. It is only when we probe its limitations under more strenuous conditions that its gilded facade begins to show cracks.

5.1.2 Exposing the "Black Box" Vulnerabilities

While the first benchmark validated the power of the semantic paradigm, the second and third benchmarks were designed to expose its inherent limitations. The dramatic drop in performance of the Industrial Baseline and other semantic models under these more challenging conditions is perhaps the most significant finding of this entire study. It reveals a fundamental vulnerability that is not a flaw in the implementation, but a direct and unavoidable consequence of the "black box" approach itself.

The "Adversarial Robustness" benchmark (Benchmark 2) served as a crucible. The Industrial Baseline's mediocre Robustness Score provides a clear diagnosis: its reliance on holistic semantic similarity makes it fundamentally vulnerable to sophisticated deception. The very mechanism that allows it to see that "Java developer" and "Python developer" are both "software engineers" also makes it susceptible to the "Semantic Similarity Trap," where it fails to enforce the non-negotiable distinction between the two. Its success is predicated on a generalization that, under adversarial pressure, becomes a critical liability. It cannot easily distinguish between a candidate who is genuinely qualified and one who has simply "stuffed" their CV with semantically adjacent keywords. The model understands the "what" of the language, but it lacks the structured, rule-based reasoning to understand the "what not."

This vulnerability is further underscored by its poor performance on the "Ranking Precision" benchmark (Benchmark 3). In a controlled environment where the differences between candidates were subtle and incremental, the semantic engine was the worst-performing model. This proves a crucial point: **high semantic similarity is not always a proxy for best fit**. The model's generalizing nature, so useful for understanding broad concepts, fails when tasked with discerning the fine-grained, specific details that separate a 9/10 candidate from a 10/10 candidate. In this context, the simpler, statistical precision of a model like TF-IDF proved superior, highlighting the semantic engine's inherent lack of specificity.

In conclusion, the benchmark results paint a complete and nuanced portrait of the modern semantic engine. It is, without question, a powerful and highly effective tool for general-purpose filtering, representing a massive leap over older technologies. However, its core design principle reducing complex documents to a single, opaque vector of "meaning" is also the source of its greatest weakness. The evidence is clear: its power is brittle, its intelligence can be easily fooled, and its precision is lacking. Semantic power alone is insufficient for creating a recruitment system that is not only effective but also truly robust, precise, and trustworthy. This conclusion demonstrates, with empirical data, the clear and pressing need for a new architectural paradigm.

5.2 The Vindication of the MAALEM Framework

If the analysis of the baseline models diagnoses the problem, the performance of the MAALEM engine provides the solution. The introduction of MAALEM into the benchmarks did not just result in an incremental improvement; it established a new performance ceiling, particularly in the critical domain of adversarial robustness. This success is not accidental. It is the direct, empirical validation of the core design philosophy that underpins the entire framework.

5.2.1 An Architecture Designed for Robustness

The most compelling evidence for the superiority of the MAALEM framework comes from its commanding performance on the "Adversarial Robustness" benchmark (Benchmark 2). While every other model, including the powerful Industrial Baseline, struggled and was susceptible to deception, MAALEM achieved a perfect **Trap Avoidance Rate of 100%**. This flawless result is a direct consequence of its foundational principle: the deconstruction of complexity.

Unlike a monolithic semantic engine that processes a document as a single, indivisible blob of meaning, MAALEM deconstructs the evaluation into independent analytical axes. This architecture allows the framework to apply the *correct type of logic* to each piece of information. For criteria where semantic understanding is beneficial (like understanding the substance of past work experience), it can leverage powerful embedding models. However, for criteria where specificity is non-negotiable (such as the presence of a specific, required technical skill or certification), it can apply strict, non-semantic, rule-based checks.

This hybrid logic acts as an architectural "firewall." A "trap" CV, designed to fool a purely semantic model by being stuffed with related but incorrect keywords (e.g., a Python expert applying for a Java role), will inevitably fail these strict, specific checks within the MAALEM framework. The system is not fooled because it is not limited to a single mode of thinking. It correctly identifies the semantic similarity but also recognizes the critical, specific mismatch, and can therefore reject the candidate with confidence. This result is the ultimate vindication of the deconstructed approach, proving empirically that a system designed with built-in specificity and multiple logical pathways is fundamentally more resilient and trustworthy than any "black box" however powerful it may be.

5.2.2 The Unquantifiable Value of Explainability and Control

While the quantitative results from the benchmarks unequivocally establish the superiority of the MAALEM engine in terms of accuracy and robustness, perhaps its most significant contribution lies in a dimension that these tests cannot fully measure: **user trust**. The ultimate goal of this research was not merely to build a more accurate system, but to build a more useful one. In a professional domain, utility is a direct function of trust, which is born from transparency and control.

This is where the principles of explainability and user-centric control come to the forefront. The Industrial Baseline, like all black-box systems, provides a form of *dictation*. It delivers a ranked list as a final, immutable answer. The MAALEM framework, in contrast, facilitates a *dialogue*. Its interactive dashboard, with its multi-faceted radar chart and tunable weights, fundamentally changes the relationship between the recruiter and the AI.

To illustrate this, let us revisit the qualitative case study of a "Senior Data Scientist" role with two distinct candidates:

- **Candidate A (The Specialist):** A candidate with only 3 years of experience but with exceptional, cutting-edge skills in the exact technologies required for the project.
- **Candidate B (The Veteran Generalist):** A candidate with 10 years of experience and a PhD, but whose technical skills are in older, less relevant technologies.

A monolithic semantic engine might produce a single, opaque score, perhaps ranking Candidate B higher due to the semantic weight of their seniority and PhD. The recruiter is left with a number but no context. The MAALEM framework, however, provides a detailed, explainable profile for each. The recruiter would instantly see on the radar chart that Candidate A excels on the 'Skill' axis but is lower on the 'Years of Experience' axis, while Candidate B shows the exact inverse profile.

This granular, explainable insight is transformative. It moves the conversation beyond "who is better?" to "who is better *for our specific needs right now?*" Empowered with this transparent data and the ability to adjust the weights in the control panel, the recruiter can make a final, informed decision that aligns with their team's immediate strategic priorities. They might choose Candidate A for a project requiring immediate, cutting-edge technical impact, or they might prioritize Candidate B for a role requiring long-term mentorship and foundational knowledge.

The AI does not make the decision; it illuminates the trade-offs so the human expert can. This is the unquantifiable value of the MAALEM framework. It is not just a better recommendation engine; it is a

better thinking tool. It transforms the system from a source of opaque answers into a transparent, trustworthy partner that respects and augments the expertise of the user it is designed to serve.

5.3 Critical Reflection: Scope and Limitations

A comprehensive discussion requires not only an interpretation of a project’s successes but also a transparent and critical reflection on its scope, trade-offs, and inherent limitations. This section provides such a reflection, acknowledging the specific contexts in which each developed system excels and the boundaries of the current research.

5.3.1 The Industrial Baseline: A Pragmatic and Fit-for-Purpose Deliverable

It is crucial to frame the Industrial Baseline not as a "lesser" model, but as the correct and pragmatic solution for the specific mission it was designed to fulfill. The primary mandate was to deliver a scalable, end-to-end semantic search engine that checked a specific list of technical features parsing, embedding, and high-speed search. The chosen architecture, leveraging best-in-class technologies like Gemini and FAISS, is an optimal solution for this problem statement.

Its "black box" nature, which we have identified as a weakness from a research perspective, can be viewed as an acceptable trade-off in an industrial context where speed, scalability, and ease of maintenance are often prioritized over deep explainability. For many high-volume screening tasks, its powerful filtering capability is more than sufficient. The baseline system is, therefore, a successful engineering artifact, perfectly tailored to its specific requirements. Its value lies in its pragmatic effectiveness; its limitations only become apparent when we ask more demanding questions about trustworthiness and resilience, which led to the development of MAALEM.

5.3.2 The MAALEM Framework: The Trade-off of Specialization

The superior performance of the MAALEM framework, particularly its robustness, is not a result of a "magic bullet" algorithm, but rather a direct consequence of its specialized, knowledge-intensive design. This specialization, however, comes with its own set of limitations and trade-offs that must be acknowledged.

Domain Specificity as a Bottleneck

The high precision and robustness of the MAALEM engine are deeply rooted in its nuanced understanding of a single domain: Information Technology. Its ability to distinguish between relevant and irrelevant skills, for instance, is powered by a knowledge structure that was meticulously developed for this specific field.

This presents the primary limitation for scalability. While the framework’s architecture is modular and designed for expansion, extending its high performance to entirely new domains (such as healthcare, finance, or law) would be a non-trivial undertaking. It would require a significant, labor-intensive effort to build new, domain-specific knowledge components analogous to those created for the IT domain. The system’s current superiority is, therefore, specialized. This is the fundamental trade-off: MAALEM exchanges the "one-size-fits-all" generality of the baseline for a much deeper, more robust, but narrower, level of expertise.

Dependency on High-Fidelity Extraction

The second limitation is inherent to all systems of this type: the quality of the matching logic is fundamentally capped by the quality of the initial data extraction. The principle of "garbage in, garbage out" applies unequivocally. The sophisticated, multi-faceted analysis performed by MAALEM assumes that it is operating on accurately parsed and structured data.

In this project, we ensured the highest possible data fidelity for our experiments by leveraging a powerful LLM for the extraction task. However, in a real-world deployment, any errors or omissions

in this initial parsing stage such as a failure to identify a key certification or misinterpreting a work duration would inevitably propagate downstream and impact the final matching score. While our proposed long-term architecture includes a Human-in-the-Loop (HITL) mechanism to mitigate this, it remains a critical dependency for the entire system’s performance.

Chapter 6

General Conclusion

This project was initiated to address a critical dilemma at the heart of modern recruitment technology: the debilitating trade-off between the undeniable power and the inherent opacity of state-of-the-art semantic search engines. While these systems are highly effective at understanding language, their "black box" nature creates a fundamental misalignment with the need for transparency, trustworthiness, and justification in high-stakes hiring decisions.

Throughout this report, we have detailed a dual-track journey undertaken to navigate this challenge. The first path involved the successful engineering and delivery of a powerful **Industrial Baseline Engine**, a system that fulfilled 100

Through a rigorous, multi-faceted evaluation, this project has empirically proven that a more transparent, robust, and human-centric approach to person-job matching is not only possible, but demonstrably superior. The work accomplished provides both a practical, production-ready tool and a conceptual blueprint for the next generation of recruitment AI.

6.1 Summary of Core Contributions

The work accomplished throughout this project has resulted in several key contributions to both industrial practice and the academic understanding of person-job fit technology. These achievements can be summarized as follows:

- (a) **Fulfillment of a Complete Industrial Mandate.**
This project successfully delivered a complete, production-ready, and high-performance semantic search engine that fulfilled 100
- (b) **The MAALEM Framework as a New Paradigm for Explainable AI.**
Beyond the industrial deliverable, this thesis conceived, designed, and validated a novel framework for explainable AI in recruitment. The MAALEM engine represents a significant conceptual advance over existing systems. Its core design philosophy based on the principles of deconstruction, explainability by design, and user-centric control directly solves the "black box" problem that plagues monolithic semantic engines. It provides a proven architectural blueprint for building AI systems that are not just powerful, but are also fundamentally transparent and trustworthy.
- (c) **Development of a Rigorous Scientific Validation Protocol.**
Recognizing a critical gap in the field, this project included the design and implementation of a comprehensive, custom-built benchmark suite for evaluating recruitment AI. This suite, featuring three distinct tests for filtering, robustness, and precision, provides a much-needed framework for scientific evaluation. The "Trapped Binary Noise Rank" benchmark, in particular, proved to be an invaluable tool. Its adversarial design was crucial for empirically demonstrating the hidden vulnerabilities of standard semantic systems and, conversely, for providing the definitive proof of the MAALEM framework's superior resilience and intelligence.

6.2 Final Project Verdict

The empirical evidence presented in this report is conclusive. The **MAALEM** engine has been demonstrated to be not just incrementally more accurate, but fundamentally more robust, trustworthy, and aligned with the strategic needs of professional recruiters than existing state-of-the-art systems. By deconstructing the matching problem and prioritizing explainability, it successfully overcomes the "black box" limitations of the powerful Industrial Baseline, proving that a higher level of performance and intelligence is achievable.

Ultimately, this project did not just build a tool; it delivered a new blueprint for the future of intelligent matching systems one where performance and transparency are no longer mutually exclusive.

Appendix A

Algorithmic Workflow of the Industrial Baseline Engine

This appendix provides high-level pseudo-code to illustrate the core workflows of the Industrial Baseline Engine, the system delivered to the host company. This details the orchestration of the three primary technologies Google Gemini for parsing, Sentence-Transformers for embedding, and FAISS for search to fulfill the project's industrial mandate.

A.1 The ‘index’ Command Workflow

This workflow is initiated when the user wants to populate the database with a new set of candidate CVs. It iterates through a directory of documents and processes each one for storage.

Algorithm 1 ‘index’ Command Workflow

```
1: function INDEXCVs(cv_folder_path)
2:   file_paths ← FINDALLFILES(cv_folder_path, extensions=['.pdf', '.txt'])
3:   faiss_database ← LOADFAISSINDEX('path/to/db')
4:   metadata_store ← LOADMETADATA('path/to/metadata')
5:   for each file_path in file_paths do
6:     print("Processing file:", file_path.name)
7:     document_vector, document_metadata ← PROCESSDOCUMENT(file_path)
8:     if document_vector is not Null then
9:       ADDVECTORTODB(faiss_database, document_vector)
10:      ADDMETADATA(metadata_store, document_metadata, file_path.name)
11:      print("Successfully indexed:", file_path.name)
12:   SAVEFAISSINDEX(faiss_database)
13:   SAVEMETADATA(metadata_store)
```

A.2 The ‘match’ Command Workflow

This workflow is initiated when a user wants to find the best candidates for a specific job offer. It processes the job offer and queries the existing database.

Algorithm 2 ‘match’ Command Workflow

```
1: function MATCHJOB OFFER(job_offer_path, top_k)
2:   faiss_database  $\leftarrow$  LOADFAISSINDEX('path/to/db')
3:   metadata_store  $\leftarrow$  LOADMETADATA('path/to/metadata')
4:   print("Processing job offer:", job_offer_path.name)
5:   query_vector,  $\_$   $\leftarrow$  PROCESSDOCUMENT(job_offer_path)
6:   if query_vector is Null then
7:     print("Error: Could not process job offer.")
8:     return
9:   scores, indices  $\leftarrow$  SEARCHDB(faiss_database, query_vector, top_k)
10:  print("--- MATCHING RESULTS ---")
11:  for i from 0 to length of indices - 1 do
12:    candidate_id  $\leftarrow$  indices[i]
13:    similarity_score  $\leftarrow$  scores[i]
14:    candidate_metadata  $\leftarrow$  GETMETADATABYID(metadata_store, candidate_id)
15:    print(Rank: i+1, Candidate: candidate_metadata.filename, Score:
      similarity_score)
```

A.3 Core Helper Function: Document Processing Pipeline

Both the ‘index’ and ‘match’ workflows rely on a shared pipeline to convert a raw document file into a semantic vector and structured metadata. This reusable function encapsulates the core logic of the system.

Algorithm 3 Document Processing Pipeline

```
1: function PROCESSDOCUMENT(file_path)
2:                                      $\triangleright$  Stage 1: Parsing
3:   raw_text  $\leftarrow$  EXTRACTTEXTFROMFILE(file_path)
4:   if raw_text is Null or Empty then
5:     return Null, Null
6:   structured_json  $\leftarrow$  CALLGEMINIAPI(raw_text)
7:   if structured_json is Null or Invalid then
8:     return Null, Null
9:                                      $\triangleright$  Stage 2: Embedding
10:  composite_string  $\leftarrow$  CREATESTRINGFROMJSON(structured_json)
11:  embedding_model  $\leftarrow$  LOADSENTENCETRANSFORMERMODEL
12:  vector  $\leftarrow$  ENCODETEXT(embedding_model, composite_string)
13:                                      $\triangleright$  Stage 3: Normalization for Search
14:  normalized_vector  $\leftarrow$  L2NORMALIZE(vector)
15:  return normalized_vector, structured_json
```

Bibliography

- [1] G. Smart and R. Street, “Who: The a method for hiring,” 2008.
- [2] G. Salton and C. Buckley, “Term-weighting approaches in automatic text retrieval,” *Information processing & management*, vol. 24, no. 5, pp. 513–523, 1988.
- [3] 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*, 2019.
- [4] Gemini Team, *et al.*, “Gemini: A family of highly capable multimodal models,” *arXiv preprint arXiv:2312.11805*, 2023.
- [5] J. Johnson, M. Douze, and H. Jégou, “Billion-scale similarity search with gpus,” *IEEE Transactions on Big Data*, vol. 7, no. 3, pp. 535–547, 2019.
- [6] M. Honnibal, I. Montani, S. Van Landeghem, and A. Boyd, “spacy: Industrial-strength natural language processing in python,” 2020. [Online]. Available: <https://doi.org/10.5281/zenodo.1212303>
- [7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in neural information processing systems*, vol. 30, 2017.
- [8] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, “GLUE: A multi-task benchmark and analysis platform for natural language understanding,” in *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, 2018, pp. 353–355.
- [9] D. Hendrycks, C. Burns, S. Kadavath, A. Arora, S. Basart, E. Tang, D. Song, and J. Steinhardt, “Measuring mathematical problem solving with the MATH dataset,” *arXiv preprint arXiv:2103.03874*, 2021.
- [10] M. Chen, J. Tworek, H. Jun, Q. Yuan, H. d. P. Pires, Q. Le, S. Seneviratne, Y. Tan *et al.*, “Evaluating large language models trained on code,” *arXiv preprint arXiv:2107.03374*, 2021.
- [11] D. Hendrycks, C. Burns, S. Basart, A. Zhmoginov, P. Mishkin, K. Gimpel, and J. Steinhardt, “Measuring massive multitask language understanding,” *arXiv preprint arXiv:2009.03300*, 2020.
- [12] K. Järvelin and J. Kekäläinen, “Cumulated gain-based evaluation of ir techniques,” *ACM Transactions on Information Systems (TOIS)*, vol. 20, no. 4, pp. 422–446, 2002.
- [13] S. Robertson and H. Zaragoza, “The probabilistic relevance framework: Bm25 and beyond,” *Foundations and Trends® in Information Retrieval*, vol. 3, no. 4, pp. 333–389, 2009.
- [14] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, “Pytorch: An imperative style, high-performance deep learning library,” in *Advances in Neural Information Processing Systems 32*, 2019, pp. 8026–8037.
- [15] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush, “Transformers: State-of-the-art natural language processing,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, Oct. 2020, pp. 38–45. [Online]. Available: <https://www.aclweb.org/anthology/2020.emnlp-demos.6>

- [16] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, “Scikit-learn: Machine learning in python,” *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.