الجمهورية الجزائرية الديمقراطية الشعبية
People's Democratic Republic of Algeria
Ministry of higher education and scientific research
University Akli Mohand Oulhadj of Bouira
ⵜⴰⵙⴷⴰⵡⵉⵜ ⴰⴽⵍⵉ ⵎⵓⵃⴻⵏⴷ ⵓⵍⵃⴰⵊ - ⵜⵓⴱⵉⵔⴻⵜ -
Faculty of Exact Sciences
Department of Computer Science
وزارة التعليم العالي والبحث العلمي
جامعة أكلي محند أولحاج
- البويرة -
كلية العلوم الدقيقة
قسم الإعلام الآلي

الإذن بإيداع مذكرة مشروع التخرج لطور الماستر من أجل المناقشة

لقب واسم المشرف طه زروقي

الطالب الأول نسرين بن صادوق

الطالب الثاني مروة مقراب

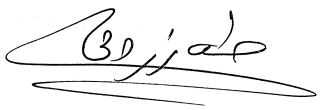
السيد رئيس قسم الإعلام الآلي المحترم، أحيطكم علما بأن الطالبين المذكور اسميهما أعلاه قد أتما عملهما اللازم لمشروع تخرجهما والذي يحمل العنوان:

Evaluation of LLM Models and AI Engines: Case of Arabic NLP Task – إعراب (Arabic Syntax Analysis)

تقييم النماذج اللغوية الضخمة وأدوات الذكاء الاصطناعي في معالجة اللغة العربية، مهمة الإعراب.

ولذا فإني أسمح لهما بتقديم العمل المنجز أمام لجنة التحكيم التي يعينها القسم، وأعلمكم أيضا أن المذكرة خالية من أي اقتباس غير شرعي أو سرقة علمية حسب علمي.

توقيع المشرف **تاريخ: 13-06-2026**



People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
Akli Mohand Oulhadj University of Bouira
Faculty of Exact Sciences
Department of Computer Science

Master Thesis

in Computer Science

Specialty: *ISIL*

Topic

Evaluation of LLM Models and AI Engines: Case of Arabic NLP Task – I'rab (Arabic Syntax Analysis)

Supervised by

- Dr.TAHA Zerrouki

Prepared by

- NESRINE Bensadok
- MEROUA Mokrab

Academic Year 2025/2026

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire

وزارة التعليم العالي و البحث العلمي
جامعة أكلي محند أولحاج - البويرة

Ministère de l'Enseignement Supérieur
et de la Recherche Scientifique
Université Akli Mohand Oulhadj - Bouira
Tasdawit Akli Muḥend Ulḥağ - Tubirett

التصريح الشرفي الخاص بالالتزام بقواعد النزاهة العلمية
لانجاز بحث

انا الممضي اسفله،

السيد(ة)..بن حمادوش نسرين..الصفة: طالب (ماستر / دكتوراه)

الحامل(ة) لبطاقة التعريف الوطنية: ..140446269..والصادرة بتاريخ..2023-01-25..

المسجل(ة) بكلية / معهد..العلوم الدقيقة..قسم..الإعلام الآلي..

تخصص: ..هندسة أنظمة المعلومات و البرمجيات..

والمكلف(ة) بإنجاز اعمال بحث(مذكرة، التخرج، مذكرة ماستر، مذكرة ماجستير، اطروحة دكتوراه).

عنوانها:..تقييم النماذج اللغوية المخصصة وأدوات الذكاء الاصطناعي في معالجة اللغة العربية من صعبة الاعراب..

أصرح بشرفي اني ألتزم بمراعاة المعايير العلمية والمنهجية الاخلاقيات المهنية والنزاهة الاكاديمية المطلوبة في انجاز البحث المذكور أعلاه.

التاريخ: 2026-06-14

توقيع المعني (ة)

هيئة مراقبة السرقة العلمية:

البويرة في:.....

النسبة: %

الامضاء

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire

وزارة التعليم العالي و البحث العلمي
جامعة أكلي محند أولحاج - البويرة

Ministère de l'Enseignement Supérieur
et de la Recherche Scientifique
Université Akli Mohand Oulhadj - Bouira
Tasdawit Akli Muhend Ulhağ - Tubiratt

التصريح الشرفي الخاص بالالتزام بقواعد النزاهة العلمية

لانجاز بحث

انا الممضي اسفله،

السيد(ة) مقران حمزة الصفة: طالب (ماستر / دكتوراه)

الحامل(ة) لبطاقة التعريف الوطنية: 117838445 والصادرة بتاريخ 2026.02.16

المسجل(ة) بكلية / معهد العلوم الدقيقة قسم الإعلام الآلي

تخصص: هندسة أنظمة المعلومات والبرمجيات

والمكلف(ة) بإنجاز اعمال بحث(مذكرة، التخرج، مذكرة ماستر، مذكرة ماجستير، اطروحة دكتوراه).

عنوانها: تقييم النماذج اللغوية الضخمة وأدوات الذكاء الاصطناعي في معالجة اللغة العربية وترجمة الأحاديث

أصرح بشرفي اني ألتزم بمراعاة المعايير العلمية والمنهجية الاخلاقيات المهنية والنزاهة الاكاديمية المطلوبة في انجاز البحث المذكور أعلاه.

التاريخ: 2026/06/14

توقيع المعني (ة)

هيئة مراقبة السرقة العلمية:

البويرة في: .....

النسبة: %

الامضاء

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire

وزارة التعليم العالي و البحث العلمي
جامعة أكلي محند أولحاج - البويرة

Ministère de l'Enseignement Supérieur
et de la Recherche Scientifique
Université Akli Mohand Oulhadj - Bouira
Tasdawit Akli Muhend Ulhağ - Tubirett

نيابة المديرية للتكوين العالي في الطورين الأول والثاني
والتكوين المتواصل والشهادات وكذا التكوين العالي في التدرج
رقم: 338/ ن.م.ت.ع.ط.أ.ث.ت.م.ش.ت.ع.ت/ 2023

إلى السيدة والسادة: عمداء الكليات ومديري المعهدين

الموضوع: ف/ي السرقات العلمية

المرجع: القرار رقم 1082 المؤرخ في 27 ديسمبر 2020 يحدد القواعد المتعلقة بالسرقة العلمية ومكافحتها.

تنفيذا لتعليمات السيد مدير الجامعة وتطبيقا للقرار المذكور في المرجع أعلاه، يطلب منكم السهر على تنفيذ التدابير التالية:

1- قبل إيداع مذكرات الماستر أو أطروحات الدكتوراه يجب أن يخضع العمل المنجز إلى مراقبة نسبة الانتحال من طرف الهيئة المخولة بذلك والتي تمتلك برنامج اكتشاف السرقات العلمية المعتمد بجامعتنا.
2- يمضي الطالب تصريح شرفي خاص بالالتزام بقواعد النزاهة العلمية (وثيقة مرفقة).
3- تسجل الهيئة المكلفة بمراقبة نسبة الانتحال النسبة الناتجة عن البرنامج على ذات التصريح المقدم من طرف الطالب في الإطار المخصص لذلك، وتختم وتمضى من طرف المكلف بالمراقبة.
4- تعتبر أطروحات الدكتوراه ومذكرات الماستر قابلة للمناقشة التي لا تتعدى نسبة الانتحال فيها 30%.
5- تدرج هذه الوثيقة ضمن المذكرة أو الأطروحة مباشرة بعد الغلاف (Page de garde).

يولي السيد مدير الجامعة العناية الصارمة لتطبيق فحوى هذه المراسلة.

تقبلوا فائق التقدير والاحترام.

07 JUIN 2023

نائب مدير مكلف بالتكوين العالي في الطورين الأول والثاني والتكوين المتواصل والشهادات وكذا التكوين العالي في التدرج
الأستاذة: شرفة جميلة علي

المرفقات:

- نموذج التصريح الشرفي الخاص بالالتزام بالنزاهة العلمية لإنجاز بحث.

Dedication

First and foremost, I praise and thank **Allah, the Almighty**, for His guidance and blessings, through which this work has been successfully completed. I pray that this work will be beneficial and accepted as a sincere effort.

I would also like to express my deepest gratitude to my beloved parents and all my family members for their continuous support, encouragement, and sacrifices throughout my academic journey.

My sincere appreciation also goes to our supervisor, whose guidance, valuable advice, and constant support were instrumental in the completion of this work. His expertise and dedication greatly contributed to the success of this project.

I would also like to extend my heartfelt thanks to my friend and project partner, **Nesrine Bensadok**, for her cooperation, commitment, and valuable contributions throughout this work.

Finally, I dedicate this work to everyone who supported, encouraged, and helped me, directly or indirectly, during this journey.

MEROUA Mokrab

Dedication

First and foremost, I thank **Allah Almighty** for the blessing of guidance and success, and for the support that enabled me to achieve my goal and complete this thesis.

I dedicate this humble work to my dear parents, may Allah protect and bless them, who have been my main support throughout my academic journey, and to my sisters, my brother, and my fiancé, in appreciation of their continuous support, encouragement, and constant motivation.

I also extend my sincere gratitude and appreciation to our respected supervisor, who accompanied this work throughout all its stages, providing valuable guidance and sound advice, and performing his duties with dedication, professionalism, and high commitment. His continuous support and follow-up were essential to the completion of this thesis.

I also dedicate this work to my dear friend and partner, **Mokrab Meroua**, in recognition of her cooperation, valuable efforts, and strong team spirit that greatly contributed to the success of this project.

NESRINE Bensadok

Abstract

Arabic syntactic analysis (I‘rab) is a challenging task in Arabic Natural Language Processing due to the language’s rich morphology and complex grammatical structures. Although Large Language Models (LLMs) have achieved remarkable success in many NLP tasks, their ability to perform Arabic I‘rab remains insufficiently explored.

This work presents a comparative evaluation of several modern LLMs on a benchmark dataset of 450 Arabic sentences covering different sentence types, text sources, and levels of syntactic complexity. An automatic evaluation framework was developed using text normalization, token overlap, TF-IDF similarity, grammatical term coverage, and grammatical conflict detection.

The results show significant differences among the evaluated models in terms of accuracy and consistency. Model performance was affected by sentence complexity and source, with generally better results on simple and standard Arabic texts than on complex, poetic, or Qur’anic sentences. Among the evaluated models, Qwen achieved the best overall performance.

The study also highlights limitations of strict automatic evaluation, as some correct answers may be penalized because of differences in wording or detail despite conveying the same grammatical meaning. These findings contribute to a better understanding of LLM capabilities in Arabic I‘rab and provide a foundation for future research on automatic evaluation of Arabic linguistic tasks.

Keywords: Arabic NLP, I‘rab, Syntactic Analysis, Large Language Models, LLM Evaluation, Arabic Grammar.

الملخص

يُعدّ التحليل النحوي العربي (الإعراب) من التحديات الصعبة في معالجة اللغة العربية الطبيعية، وذلك بسبب ثراء اللغة من الناحية الصرفية وتعقيد بنيتها النحوية. وعلى الرغم من تحقيق النماذج اللغوية الكبيرة (LLMs) نجاحًا ملحوظًا في العديد من مهام معالجة اللغة الطبيعية، إلا أن قدرتها على أداء الإعراب العربي ما تزال غير مستكشفة بشكل كافٍ.

تقدّم هذه الدراسة تقييماً مقارنًا لعدة نماذج لغوية حديثة على مجموعة بيانات معيارية تضم 450 جملة عربية تغطي أنماطًا مختلفة من الجمل، ومصادر نصوص متعددة، ومستويات متنوعة من التعقيد النحوي. وقد تم تطوير إطار تقييم آلي يعتمد على تطبيع النصوص، وتطابق الرموز، وقياس تشابه TF-IDF، وتغطية المصطلحات النحوية، وكشف التعارضات النحوية.

تُظهر النتائج وجود اختلافات كبيرة بين النماذج المُقيّمة من حيث الدقة والاتساق. وقد تأثرت أداءات النماذج بدرجة تعقيد الجمل ومصدرها، حيث كانت النتائج أفضل عمومًا في النصوص العربية البسيطة والمعيارية مقارنة بالجمل المعقدة أو الشعرية أو القرآنية. ومن بين النماذج المُختبرة، حقق نموذج Qwen أفضل أداء إجمالي.

كما تسلط الدراسة الضوء على قيود التقييم الآلي الصارم، إذ قد يتم احتساب بعض الإجابات الصحيحة كأخطاء بسبب اختلاف الصياغة أو مستوى التفصيل رغم تطابق المعنى النحوي. وتساهم هذه النتائج في تحسين فهم قدرات نماذج اللغة الكبيرة في الإعراب العربي، وتوفر أساسًا لأبحاث مستقبلية في مجال التقييم الآلي للمهام اللغوية العربية.

الكلمات المفتاحية: معالجة اللغة العربية الطبيعية، الإعراب، التحليل النحوي، النماذج اللغوية الكبيرة، تقييم النماذج، النحو العربي.

Résumé

L'analyse syntaxique arabe (I'rab) constitue une tâche complexe du traitement automatique de la langue arabe en raison de sa richesse morphologique et de ses structures grammaticales élaborées. Malgré les progrès réalisés par les grands modèles de langage (LLMs), leur capacité à effectuer l'I'rab arabe reste encore peu étudiée.

Ce travail présente une évaluation comparative de plusieurs LLMs modernes à l'aide d'un corpus de référence composé de 450 phrases arabes couvrant différents types de phrases, sources textuelles et niveaux de complexité syntaxique. Un cadre d'évaluation automatique a été développé en s'appuyant sur la normalisation des textes, le chevauchement lexical, la similarité TF-IDF, la couverture des termes grammaticaux et la détection des contradictions grammaticales.

Les résultats montrent des différences notables entre les modèles en matière de précision et de stabilité. Les performances sont généralement meilleures sur les phrases simples et les textes arabes standards que sur les phrases complexes, poétiques ou coraniques. Parmi les modèles évalués, Qwen a obtenu les meilleures performances globales.

L'étude met également en évidence certaines limites de l'évaluation automatique stricte, qui peut pénaliser des réponses correctes en raison de différences de formulation ou de niveau de détail. Ces résultats contribuent à une meilleure compréhension des capacités actuelles des LLMs pour l'I'rab arabe et ouvrent la voie à de futurs travaux sur l'évaluation automatique des tâches linguistiques arabes.

Mots-clés : TAL arabe, I'rab, Analyse Syntaxique, Grands Modèles de Langage, Évaluation des LLMs, Grammaire Arabe.

Contents

List of Figures

List of Tables

Introduction

Natural Language Processing (NLP) is a branch of artificial intelligence that enables computers to understand, interpret, and generate human language. Arabic NLP presents unique challenges due to its rich morphology, complex syntax, and diverse dialects. One of the fundamental tasks in Arabic NLP is morphological analysis and syntactic annotation (i'rab), which involves identifying the grammatical roles and structures of words in sentences.

This project focuses on evaluating the performance of existing Arabic NLP models in performing i'rab. Several state-of-the-art tools and models, including ChatGPT, Gemeni, DeepSeek, AraBERT and other large language models (LLMs), will be assessed to determine their accuracy and reliability in assigning correct morphological and syntactic tags.

The main goal of this work is to provide a comparative analysis of these models, highlighting their strengths and limitations in Arabic morphological analysis. The results aim to inform future improvements in NLP systems for accurate Arabic text processing.

Research Problematic

Despite the impressive linguistic fluency demonstrated by Large Language Models (LLMs) in text generation, their accuracy in applying strict grammatical rules (I'rāb) remains a subject of debate. This research seeks to answer the following key questions:

- What is the accuracy of current Large Language Models (LLMs) and AI engines in performing Arabic syntactic parsing (i'rab)?

Research Objectives

To address these questions, this research sets out the following objectives:

1. **Performance Evaluation**: Measure the accuracy and correctness of parsing outputs generated by prominent global and local models (e.g., ChatGPT, Gemini, Jais).

2. **Results Analysis**: Assess the overall performance of the evaluated models in Arabic i'rab, with a comparative focus on their effectiveness in correctly identifying syntactic functions.

3. **Benchmarking**: Conduct a comprehensive comparative evaluation of all selected models and tools, including modern LLMs and I'rab tools, to analyze their performance, strengths, and limitations in performing Arabic i'rab.

Chapter 1

Arabic Language and Grammatical Analysis

1.1 Introduction

The Arabic language is considered one of the most prominent world languages, distinguished by its linguistic richness and deep historical heritage [1]. It is an ancient Semitic language that has preserved its structure over long centuries and has remained an essential tool for cultural, scientific, and civilizational communication [2]. Moreover, it is the language of the Holy Quran, which has granted it a special status among the languages of the world [1], [3].

Arabic is characterized by a comprehensive linguistic system that combines precision and flexibility, relying on grammatical and morphological rules that regulate sentence structure and define the relationships between its components [4]. Grammatical analysis is considered one of the most important of these features, as it enables the understanding of the precise meanings of texts through identifying the grammatical functions of words [4].

1.2 Characteristics of the Arabic Language

Arabic is distinguished by several unique characteristics [3]:

Morphological Richness Arabic relies on a root-and-pattern system, often based on triliteral roots, from which a large number of words can be derived [5]. For example, from the root "كتب" one can derive "كاتب" "مكتوب" "كتابة" "مكتبة" and others [5]. These roots are considered fundamental forms used to generate words and represent limited sets summarizing the linguistic features of a word [6].

Derivation Derivation is one of the most important means of generating vocabulary in Arabic, giving the language a great ability to express new concepts [5]. Arabic is highly derivational, as many affixes can be attached to each root or stem, increasing the number of possible inflected words [7].

Diacritization (Harakat) Diacritical marks play an important role in determining meaning, such as the difference between "عَلِمَ" (verb) and "عِلْم" (noun) [4].

Flexible Sentence Structure Sentence elements can be reordered without affecting the essential meaning, as in:

"كتبَ الطالبُ الدرسَ".
"الدرسَ كتبَ الطالبُ".

"قرأَ أحمدُ الكتابَ في المكتبةِ".
"في المكتبةِ قرأَ أحمدُ الكتابَ".

1.3 Grammatical Analysis (الإعراب)

Grammatical analysis refers to the change in word endings according to the grammatical factors affecting them, either explicitly or implicitly, and it is an essential means of understanding grammatical relationships within a sentence [4]. It includes identifying [4]:

- The type of the word (noun (اسم), verb (فعل), or particle (حرف)).
- Its position in the sentence.

- Its grammatical case.
- Its grammatical marker.

Illustrative Example "نجحَ الطالبُ المجتهدُ":

- نجحَ: فعل ماضٍ .
- الطالبُ: فاعل مرفوع.
- المجتهدُ: صفة (نعت) مرفوعة.

1.4 Parts of Speech in Arabic

Words in Arabic are divided into three main categories [4]:

Noun (اسم) A noun denotes a meaning in itself without being associated with a specific time, such as "كتاب" "طالب" "علم" [4].

Verb (فعل) A verb denotes an action associated with tense and is divided into past, present, and imperative forms [4].

Particle (حرف) A particle has no complete meaning unless associated with another word, such as "في" "على" "من" [4].

1.5 Types of Grammatical Analysis and Their Markers

Grammatical analysis is divided into four basic types, each associated with a primary grammatical marker [4]:

Nominative Case (Al-Raf‘ / الرفع) The Nominative Case (Al-Raf‘ / الرفع) is one of the principal grammatical cases in Arabic. Its primary marker is the Dammah (الضمة), which appears on words functioning as the subject, predicate, or agent in a sentence. In certain grammatical forms, secondary markers replace the Dammah; for example, the Alif (الألف) is used in the dual form, while the Waw (الواو) is used in the masculine sound plural[4].

Accusative Case (Al-Naṣb / النصب) The Accusative Case (Al-Naṣb / النصب) is marked primarily by the Fatha (الفتحة). It commonly appears on objects, circumstantial accusatives, and specifications. In some cases, secondary markers are used instead of the Fatha, such as the Ya’ (الياء) in the dual form and the masculine sound plural[4].

Genitive Case (Al-Jarr / الجر) The Genitive Case (Al-Jarr / الجر) is specific to nouns and is primarily marked by the Kasrah (الكسرة). It occurs in genitive constructions, including nouns governed by prepositions and possessive structures. In the dual form and the masculine sound plural, the Ya’ (الياء) serves as a secondary marker replacing the Kasrah[4].

Jussive Case (Al-Jazm / الجزم) The Jussive Case (Al-Jazm / الجزم) applies exclusively to present tense verbs. Its primary marker is the Sukoon (السكون), which indicates the jussive state of the verb. Unlike the other grammatical cases, it generally does not have commonly used secondary markers[4].

1.6 Grammatical Factors (العوامل النحوية)

A grammatical factor is the reason that causes a change in the grammatical state of a word, and it is divided into [4]:

- **Explicit Factors (العوامل الصريحة) :** Such as verbs and particles [4].
- **Implicit Factors (العوامل الضمنية) :** Such as initiation in the nominal subject [4].

1.7 Importance of Grammatical Analysis (الإعراب)

The importance of grammatical analysis lies in [4]:

- Determining the precise meaning of the sentence.
- Removing ambiguity and confusion.
- Understanding religious and literary texts.
- Improving reading and writing skills.

The grammarian Al-Zajjaj (الزجّاج) emphasized the close relationship between grammatical analysis (الإعراب) and meaning, stating that case marking was introduced into speech to distinguish between the subject and the object, the possessor and the possessed, the genitive construction and other syntactic relations that affect nouns according to their position in the structure. Therefore, grammatical analysis (الإعراب) serves as an essential mechanism for preserving meaning and preventing misinterpretation[8].

Despite its importance in understanding Arabic sentence structure, grammatical analysis (i'rab) faces several challenges, including the similarity of grammatical markers, the possibility of multiple grammatical interpretations depending on context, the absence of diacritics in written texts, and the complexity of long and compound sentences. These factors make syntactic analysis and meaning interpretation more difficult [4], [9].

1.8 Conclusion

This chapter presented an overview of the fundamental linguistic characteristics of the Arabic language, with a focus on its morphological richness, derivational capacity, the importance of the diacritization system, and the flexibility of Arabic sentence structure. It also addressed grammatical analysis (i'rab) as a central element in the Arabic grammatical system, explaining its role in identifying the syntactic functions of words and highlighting the structural and semantic relationships between them, which makes it an essential tool for understanding and accurately interpreting Arabic texts. In addition, the chapter discussed the parts of speech in Arabic, the different types of grammatical analysis and

their primary and secondary markers, as well as the grammatical factors that influence word inflection. Finally, it emphasized the importance of i'rab in preserving meaning and removing ambiguity, while outlining the main challenges faced in syntactic analysis. These concepts and theoretical foundations constitute the necessary background for understanding Arabic grammatical analysis and studying modern computational approaches designed for its automatic processing and evaluation.

Chapter 2

Large Language Models and Arabic NLP

2.1 Introduction

This chapter presents the theoretical framework of our project, starting from the main idea of the research, which focuses on evaluating large language models in the task of grammatical analysis (I'rab). From this perspective, it first addresses Natural Language Processing (NLP) and its role in enabling computers to understand and process human language.

It then explains the relationship between these technologies and the Arabic language, which is characterized by a complex syntactic and morphological structure that makes its automatic processing a significant challenge in the field of artificial intelligence. After that, large language models (LLMs) are introduced as the most prominent development in this domain, highlighting their importance in language understanding and generation.

Finally, the chapter presents a set of general-purpose and Arabic-specific language models, in addition to grammatical analysis tools, as a theoretical foundation for the experimental study that will be presented in the following chapters.

2.2 Definition of Natural Language Processing (NLP)

Natural Language Processing (NLP) is a dynamic and expanding field within Artificial Intelligence (AI) and computer science that aims to bridge the gap between human lan-

guage and machine understanding [10][11]. It involves computational linguistics, artificial intelligence, and computer language processing [10]. NLP combines methodologies from linguistics, machine learning, and deep learning to enable systems to process, analyze, understand, and generate human language in a meaningful way [12][10][11]. This interdisciplinary field draws on computer science, linguistics, cognitive science, psychology, and mathematics for its research and development [13].

2.3 Definition of Large Language Models (LLMs)

Large Language Models (LLMs) are advanced AI systems that leverage deep learning, particularly transformer architecture, to comprehend and generate human-like language [14]. These models are characterized by their massive neural network architectures and extensive training datasets, enabling them to process and create diverse content types, including text, images, audio, and video [14]. Key components such as self-attention mechanisms, embedding layers, and dense feed-forward layers allow LLMs to effectively capture syntactic and semantic information from vast amounts of textual data [14]. They learn through self-supervised methods like masked language modeling, predicting missing text segments without human labels, which establishes a deep understanding of language structure and reasoning [14].

This pre-training is followed by fine-tuning for specific Natural Language Processing (NLP) applications like translation, summarization, question answering, and code generation [14].

2.4 The Relationship Between Grammatical Analysis (الإعراب) and Arabic NLP

Grammatical analysis (الإعراب) is considered a fundamental linguistic resource for Arabic Natural Language Processing (NLP), as it provides explicit information about syntactic structure, word functions, and sentence relations. In Arabic, where word order is relatively flexible and meaning is often determined through case endings and morphological markers, الإعراب plays a key role in resolving ambiguity and identifying grammatical roles such

as subject, object, and predicate. This makes it highly relevant for computational tasks such as parsing, machine translation, and information extraction.

In modern NLP systems, especially those based on Large Language Models (LLMs), grammatical analysis serves both as a training signal and as an evaluation benchmark. Models that are capable of correctly performing الإعراب demonstrate a stronger understanding of Arabic syntax and morphology. Consequently, integrating grammatical knowledge into NLP pipelines improves the quality of Arabic text processing and supports more accurate syntactic parsing, which is essential for applications such as Arabic educational tools, language learning systems, and advanced linguistic research.

2.5 The Relationship Between Grammatical Analysis (الإعراب) and Large Language Models (LLMs)

Grammatical analysis (I'rab) is closely related to Large Language Models (LLMs), as understanding syntactic structures and grammatical markers is essential for processing the Arabic language. I'rab tasks are commonly used to evaluate the ability of these models to analyze sentences and accurately identify the grammatical functions of words, LLMs can also contribute to automating grammatical analysis by providing syntactic interpretations of Arabic sentences. Therefore, I'rab represents an important application in Arabic Natural Language Processing and serves as an effective benchmark for assessing the linguistic competence of these models.

2.5.1 LLMs as Tools for Grammatical Analysis (الإعراب)

LLMs can also assist in Arabic grammatical studies through:

- **Morphological analysis and diacritic prediction,** enabling root-pattern decomposition and context-based case prediction [15].
- **Metalinguistic explanation and rule extraction,** where LLMs can describe grammatical rules or extract patterns from corpora, benefiting both researchers and learners [16].

- **Corpus-grounded grammatical research,** allowing large-scale analysis of annotated Arabic datasets to study grammatical analysis (الإعراب) phenomena systematically [17].

2.5.2 LLMs as Subjects of Grammatical Analysis (الإعراب) Evaluation

Arabic grammatical analysis (الإعراب) poses challenges for LLMs because of:

- **Morphological richness:** Correct case endings require identifying grammatical roles, handling agreement (gender, number, case), and often understanding roots and patterns [15]. Research shows that LLMs may struggle with fine-grained linguistic tasks such as precise POS tagging and complex syntactic annotation [15].
- **Syntactic structure:** Phenomena like verb–subject agreement and flexible word order are closely tied to grammatical analysis (الإعراب). While LLMs can capture surface-level grammatical patterns, they may struggle with deeper hierarchical or long-distance dependencies [18].
- **Diacritic omission and ambiguity:** The absence of diacritics in written Arabic creates lexical and grammatical ambiguity, requiring strong contextual disambiguation abilities from LLMs [15].

2.6 General LLMs and Their Relationship to Arabic and Grammatical Analysis (الإعراب)

This section provides a definition of Large Language Models (LLMs) and their relationship to the Arabic language and grammatical analysis (الإعراب).

2.6.1 ChatGPT

ChatGPT, developed by the company *OpenAI* [1], is a multilingual large language model designed to understand and generate natural language across a wide range of languages, including Arabic.

In this study, through practical experimentation, we observed that ChatGPT demonstrates a good ability to process Arabic text and provide grammatical explanations, including basic forms of (الإعراب). It can analyze sentence structure at both the word and sentence levels and generate coherent linguistic descriptions.

2.6.2 Claude

Claude, developed by Anthropic, is a multilingual large language model capable of processing Arabic text. It can provide grammatical explanations and structured sentence analysis. Its strength in long-context understanding may support complex Arabic sentence parsing; however, like other general LLMs, it is not specifically trained for traditional grammatical analysis (الإعراب), making systematic evaluation necessary for this task.[2]

2.6.3 Gemini

Gemini, developed by Google DeepMind, a subsidiary of Google, supports multilingual language processing, including Arabic. It can analyze Arabic sentences, generate grammatical explanations, and assist in educational content. However, our experiments have also shown that Gemini is capable of performing *i'rab* (Arabic grammatical parsing and syntax analysis).[3]

2.6.4 Qwen

Qwen, developed by Alibaba Cloud, is designed as a multilingual LLM capable of handling various NLP tasks. While it supports Arabic text processing, its performance in detailed

[1] OpenAI ChatGPT: https://openai.com/chatgpt
[2] Anthropic Claude: https://www.anthropic.com/claude
[3] Google Gemini: https://deepmind.google/technologies/gemini/

syntactic tasks such as grammatical analysis (الإعراب) requires benchmarking, especially given the morphological richness and inflectional complexity of Arabic grammar.[4]

2.6.5 DeepSeek

DeepSeek, developed by DeepSeek, focuses on reasoning and coding-oriented language tasks. Although it supports multilingual input, including Arabic, its specialization is not primarily in Arabic linguistic analysis. Therefore, its ability to perform accurate grammatical analysis (الإعراب) and detailed morphological parsing must be empirically evaluated within an Arabic NLP benchmark framework.[5]

2.6.6 Mistral

Mistral models, developed by Mistral AI, are efficient open-weight LLMs capable of multilingual processing. Arabic is supported as part of their general training data. However, like other general-purpose LLMs, they are not explicitly specialized in traditional Arabic grammar. Their effectiveness in performing accurate grammatical analysis (الإعراب) depends on prompt engineering, fine-tuning, and structured evaluation against annotated Arabic datasets.[6]

2.7 Arabic LLMs and Their Relationship to Arabic and Grammatical Analysis (الإعراب)

This section presents an overview of the most prominent Arabic Language Models (Arabic LLMs) and their relationship with the Arabic language and I'rab (الإعراب)

2.7.1 Fanar

Fanar is an Arabic large language model developed by the Qatar Computing Research Institute. It aims to support Arabic-based applications through models trained specifically

[4] Alibaba Qwen: https://qwenlm.ai/
[5] DeepSeek: https://www.deepseek.com/
[6] Mistral AI: https://mistral.ai/

on Arabic linguistic data.[7] [19]

As a model designed specifically for Arabic, Fanar demonstrates stronger capability in understanding Arabic linguistic structures compared to general multilingual models. It can be applied to tasks such as syntactic analysis, extraction of morphological features, and identification of structural relationships within sentences. However, its performance in detailed traditional grammatical analysis (الإعراب) (according to classical Arabic grammar) requires careful empirical evaluation, particularly regarding case endings, inflectional markers, and complex syntactic constructions.

2.7.2 Araby.ai

Araby.ai is an Arabic artificial intelligence platform developed by Araby.ai. It aims to provide AI-powered tools that support Arabic content creation, including text generation, writing enhancement, and content production.[8]

Araby.ai primarily focuses on assisting Arabic users in content generation and language refinement. However, it is not a specialized grammatical or syntactic analysis model. While it may provide general language corrections and stylistic improvements, its capability to produce detailed traditional grammatical analysis (الإعراب) according to classical Arabic grammar rules may be limited. Therefore, its performance in grammatical analysis (الإعراب) tasks should be evaluated within a specific experimental or educational framework.

2.7.3 Jais

Jais is an Arabic large language model developed by G42 in collaboration with Mohamed bin Zayed University of Artificial Intelligence. It is considered one of the first large-scale models specifically designed to support Arabic alongside English.[9]

As a model specifically oriented toward Arabic language support, Jais demonstrates enhanced capability in understanding Arabic contexts and syntactic structures compared to general-purpose multilingual models. It can be used for sentence analysis and grammat-

[7] Fanar (QCRI): https://www.qcri.org/
[8] Araby.ai: https://araby.ai/
[9] Jais (G42 Inception): https://www.g42.ai/

ical explanation. However, its accuracy in performing detailed traditional grammatical analysis (الإعراب) depends on whether explicitly annotated Arabic grammatical data were included in its training process. Therefore, systematic evaluation is necessary within a comparative benchmarking framework.

2.7.4 Falcon

Falcon is a large language model developed by the Technology Innovation Institute in the United Arab Emirates. It is an open-weight model trained on large-scale multilingual datasets and is designed to perform various natural language processing tasks such as text generation, summarization, question answering, and content analysis.[20]

Falcon supports Arabic as part of its multilingual capabilities and can understand and generate Arabic text. However, experimental testing conducted within the framework of this study showed that the model does not perform traditional grammatical analysis (الإعراب) when explicitly requested. Instead of producing a structured syntactic breakdown specifying grammatical roles and case endings for each word, the model provided general or explanatory responses.[10]

This limitation can be attributed to the fact that Falcon is a general-purpose language model and was not specifically designed or fine-tuned for detailed Arabic grammatical analysis. Moreover, it was not explicitly trained on annotated datasets dedicated to traditional grammatical analysis (الإعراب) according to classical Arabic grammar. Consequently, its performance in this task is considered limited compared to specialized Arabic morphological and syntactic systems.

2.7.5 Others

At this stage, several Arabic language models were mentioned, including CAMeL Lab (CAMeL-BERT), MARBERT, AraGPT, AraBERT, and Arabic-LLaMA.

These models were presented for theoretical review purposes only, as they are pretrained language models rather than ready-to-use tools with direct user interfaces.

[10]Falcon https://falcon-lm.github.io/

Evaluating these models requires a dedicated programming environment, data preprocessing pipelines, and the development of custom evaluation scripts. Therefore, their practical performance on the task of Arabic grammatical analysis (الإعراب) has not been experimentally tested at this stage. However, they may be incorporated into a future experimental framework if sufficient technical resources become available.

2.8 Specialized Models for Grammatical Analysis (الإعراب) and Grammatical Analysis Tools

This section aims to present some tools and platforms specialized in Arabic grammatical analysis (الاعراب).

2.8.1 Almaqalah

Almaqalah is an Arabic online platform that provides linguistic and educational content. Among its services are tools and specialized articles dedicated to analyzing Arabic sentences and performing grammatical analysis (الإعراب).[21]

Almaqalah provides explanatory content on grammatical analysis (الإعراب) and presents applied examples of Arabic sentence analysis, making it a supportive educational tool for understanding grammatical rules. However, it is not a large language model (LLM). Rather, it is a content-based platform that relies on prepared explanations or specific analysis tools. Therefore, it can be classified as a supportive or educational system in the field of Arabic grammatical analysis, rather than a general generative AI model.[11]

2.8.2 I3rbly

I3rbly is an online platform specialized in providing automatic grammatical analysis (إعراب) (grammatical parsing) services for Arabic sentences, and it is known as "I3rbly اعرب لي". The tool offers syntactic analysis of words within a sentence, identifying the grammatical function of each word and specifying its case ending.[22]

[11] Al maqalah(plus 90) https://www.almaqalah.com/ara-aipars

I3rbly is specifically designed for the task of grammatical analysis (الإعراب), focusing on detailed syntactic analysis according to traditional Arabic grammatical rules. It attempts to identify grammatical roles such as the subject (mubtadaᵓ مبتدأ), predicate (khabar خبر), subject of the verb (fāᶜil فاعل), object (mafᶜūl bihi مفعول به), as well as nominative, accusative, genitive, and jussive cases. This makes it closer to a specialized grammatical system compared to general-purpose large language models (LLMs).[12]

2.8.3 Ajroum

Ajroum is an AI-powered platform that helps analyze Arabic texts syntactically and perform grammatical analysis الإعراب of sentences and full texts.[23]

Ajroum focuses on Arabic grammatical analysis by providing structured syntactic breakdowns of sentences according to traditional Arabic grammar rules. It aims to identify grammatical roles and case endings, supporting learners and researchers in understanding classical Arabic grammatical analysis الإعراب. As such, it functions as a specialized tool for Arabic syntactic analysis rather than a general-purpose large language model.[13]

> **Note:** Based on our preliminary experiments, we observed that all the evaluated models support the Arabic language and are capable of performing the *I'rab* task. However, the Falcon model was an exception, as it was unable to perform the *I'rab* task.

[12] I3ribly https://www.i3rbly.com/
[13] Ajroum https://ajroum.vercel.app/

Model	Developer	Arabic Support	I'rab Capability
General Large Language Models			
GPT-4 / GPT-4o / GPT-5[1]	OpenAI	Yes	Yes
Claude 3 / 3.5[2]	Anthropic	Yes	Yes
Gemini 1.5 / 2[3]	Google	Yes	Yes
Mistral / Mixtral[4]	Mistral AI	Yes	Yes
Qwen[5]	Alibaba	Yes	Yes
DeepSeek[6]	DeepSeek AI	Yes	Yes
Arabic-Oriented Language Models			
Fanar[7]	QCRI	Yes	Yes
Falcon	TII (UAE)	Yes	No
Jais[8]	G42 / MBZUAI	Yes	Yes
Araby.ai[9]	Araby.ai	Yes	Yes
I'rab-Specific Systems			
Ajroum[10]	Haithem ben halima	Yes	Yes
I3rbly[11]	I3rbly	Yes	Yes
Al Maqalah (Plus 90)[12]	—	Yes	Yes

[1] OpenAI ChatGPT: https://openai.com/chatgpt

[2] Anthropic Claude: https://www.anthropic.com/claude

[3] Google Gemini: https://deepmind.google/technologies/gemini/

[4] Mistral AI: https://mistral.ai/

[5] Alibaba Qwen: https://qwenlm.ai/

[6] DeepSeek: https://www.deepseek.com/

[7] Fanar (QCRI): https://www.qcri.org/

[8] Jais (G42 Inception): https://www.g42.ai/

[9] Araby.ai: https://araby.ai/

[10] Ajroum: https://ajroum.vercel.app/

[11] I3rbly: https://www.i3rbly.com/

[12] Al Maqalah (Plus 90): https://www.almaqalah.com/ara-aipars

Table 2.1: Categorization of the Models and Systems by Arabic Language Support and I'rab Capability

2.9 Conclusion

This chapter presented the theoretical background of the study by introducing Natural Language Processing (NLP), Large Language Models (LLMs), and their relationship with Arabic grammatical analysis (الإعراب). It highlighted the importance of الإعراب in Arabic NLP and discussed how it can be used to evaluate the linguistic capabilities of LLMs.

The chapter also reviewed a number of general-purpose LLMs, Arabic-oriented language models, and specialized grammatical analysis tools. This overview provides the theoretical foundation for the experimental evaluation presented in the following chapters.

Chapter 3

Related works

3.1 Introduction

Natural Language Processing (NLP) and Large Language Models (LLMs) have witnessed significant progress in analyzing and understanding Arabic text. However, Arabic grammatical analysis (الإعراب) remains one of the most challenging tasks due to the complexity of Arabic grammar and the richness of the language from both morphological and syntactic perspectives. Therefore, many researchers have focused on developing transformer-based models and syntactic analysis systems to improve the accuracy of Arabic linguistic processing.

This chapter reviews previous studies related to Arabic NLP, syntactic parsing, diacritization, and the evaluation of large language models such as ChatGPT and Gemini in Arabic grammatical tasks. It also highlights the relationship between these studies and the current research, which aims to evaluate the performance of large language models in Arabic grammatical analysis.

3.2 Related works

Sharefah and Al [24] examined the effectiveness of fine-tuning BERT-based pre-trained models for Arabic dependency parsing. They compared nine Arabic pre-trained models using different fine-tuning strategies and encoding techniques across three Arabic treebanks.

Their results showed that AraBERTv2 achieved the best overall performance. They further demonstrated that fine-tuning the upper transformer layers significantly improves parsing accuracy, while adding extra neural network layers can negatively impact performance. Their error analysis highlighted several challenges, including tokenization errors, annotation inconsistencies, limitations in contextual embeddings, and issues related to parse tree post-processing[24].

Maulidiya and al. [25] investigated the performance of ChatGPT and Gemini in analyzing iᶜrāb الاعراب, particularly in relation to Nominative nouns(مرفوعات الاسماء), which concern identifying syntactic roles within Arabic sentences. The study examined eleven examples selected from Summary of Arabic Grammar Rules (ملخص قواعد اللغة العربية) by Fuad Niᶜmah [26], (فؤاد نعمة)covering grammatical elements such as Subject(مبتدأ), Predicate (خبر), Subject of "kana"(اسم كان), The predicate of ⁿInna(خبر ان), Subject(فاعل), Passive subject (نائب الفاعل), Adjective(نعت), Coordination(عطف), Emphasis(توكيد), and Apposition(بدل).

To evaluate performance, the authors employed the Mann–Whitney statistical test to assess significance and the Cosine Similarity Index (CSI) to measure semantic similarity between model outputs. The results showed that ChatGPT statistically outperformed Gemini, with a significant value of 0.019. A CSI score of 0.800 indicated a high degree of semantic similarity between the two models' responses. ChatGPT demonstrated stronger performance in producing detailed and accurate grammatical analyses, whereas Gemini was more concise but occasionally less precise[25].

The study concluded that although both models show strong potential for supporting Arabic grammar analysis, human review remains necessary to ensure accuracy and pedagogical reliability in technology-assisted learning contexts [25].

Almatham et al. [27] introduced BALSAM, a comprehensive benchmark designed to evaluate Arabic Large Language Models. They emphasized that although LLMs have achieved remarkable progress in English, their performance in Arabic remains limited due to data scarcity, morphological complexity, and dialectal variation.

To address these challenges, the authors developed a community-driven benchmark covering 78 NLP tasks across 14 categories, with approximately 52,000 evaluation examples. The platform incorporates a centralized blind evaluation system to ensure fair and

contamination-free assessment. Their work contributes to establishing standardized evaluation practices for Arabic LLMs and provides a structured framework for measuring progress in Arabic NLP tasks[27].

Alyafeai et al. [28] evaluated the performance of ChatGPT models (GPT-3.5 and GPT-4) across seven Arabic NLP tasks. Their findings indicated that GPT-4 outperformed GPT-3.5 in most tasks, including part-of-speech tagging and diacritization, both of which are closely related to syntactic analysis.

This study is particularly relevant to our work as it provides empirical evidence of the effectiveness of large language models on Arabic linguistic tasks without task-specific fine-tuning. However, while their evaluation focused on general NLP benchmarks, our research extends the analysis to a deeper syntactic level through Arabic iᶜrāb.[28]

Kharsa et al. [29] proposed a BERT-based approach for Arabic diacritization that achieved state-of-the-art results on widely used datasets. By leveraging contextualized embeddings, their model significantly reduced Diacritic Error Rate (DER) and Word Error Rate (WER), demonstrating the effectiveness of transformer-based architectures in capturing Arabic morphological and syntactic patterns.

Given that diacritization is closely related to grammatical case marking and syntactic structure in Arabic, this work provides strong evidence that transformer models can effectively model linguistic features essential for tasks such as Arabic iᶜrāb. However, while their study focuses on surface-level diacritic restoration, our research extends the evaluation to deeper syntactic interpretation and comprehensive grammatical analysis.

Abdul-Mageed et al. [30] provided a comprehensive evaluation of ChatGPT on Arabic NLP tasks, showing that general-purpose LLMs do not always outperform smaller, Arabic-specialized models. Their findings revealed consistent performance gaps in dialectal Arabic and highlighted limitations in deep linguistic understanding. This is particularly relevant to our research on Arabic $i^{c}r\bar{a}b$, as it suggests that advanced generative models may still struggle with complex morpho-syntactic structures. While their study evaluates a broad range of NLP tasks, our work focuses specifically on grammatical analysis and syntactic interpretation, offering deeper insights into LLM performance on structured linguistic tasks.

Antoun et al. [31] introduced AraBERT, a transformer-based model specifically pre-

trained for Arabic language understanding. By leveraging large-scale Arabic corpora, AraBERT significantly improved performance across various NLP tasks compared to multilingual models. This work established a foundation for subsequent research in Arabic syntactic and morphological analysis, including dependency parsing and diacritization systems that rely on contextualized embeddings.

As AraBERT captures deep contextual and morphological features of Arabic, it represents an important step toward evaluating how transformer-based architectures model complex linguistic phenomena such as Arabic *iʿrāb*. Building on this progression, our research assesses the capabilities of more advanced large language models in performing fine-grained grammatical analysis.

Saadiyeh et al. [32] presented a comprehensive comparative analysis of Arabic syntactic parsing approaches. Their study examined rule-based, statistical, machine learning, hybrid, neural, and transformer-based methods, highlighting their respective strengths, limitations, and performance trade-offs. Considering the morphological richness and syntactic variability of Arabic, the authors emphasized the inherent challenges in accurately modeling syntactic relationships. The paper also provided an extensive overview of Arabic treebanks and annotated corpora, underscoring their essential role in advancing syntactic parsing research.

While their work offers a broad evaluation of Arabic syntactic parsers and linguistic resources, our study shifts the focus toward assessing modern Large Language Models (LLMs) on the task of Arabic *iʿrāb*, which requires deeper grammatical reasoning beyond conventional parsing outputs.

Green and Manning [33] provided an in-depth analysis of Arabic constituency parsing, highlighting the impact of syntactic ambiguity, morphological segmentation, and annotation inconsistencies on parser performance. Their findings demonstrated that parsing accuracy is strongly influenced by segmentation quality and treebank design, both of which are essential for reliable grammatical analysis.

Given that Arabic *iʿrāb* depends on precise syntactic structure and morphological interpretation, these findings are directly relevant to our research. While their work focuses on traditional constituency parsers, our study investigates whether modern large language models can overcome some of these structural limitations in performing fine-grained gram-

matical analysis.

Table 3.1: Classification of Related Work in Arabic NLP and I^{c}rāb Evaluation

Category	**Study**	**Short Summary**	**Relation to Our Research**
LLM Research	Abdul-Mageed et al. (2023)	Evaluated ChatGPT on Arabic NLP tasks, showing that general LLMs do not always outperform Arabic-specialized models.	Highlights limitations of general LLMs in morpho-syntactic understanding, supporting focused evaluation on Arabic *i^{c}rāb*.
	Alyafeai et al. (2023)	Compared GPT-3.5 and GPT-4 on Arabic NLP tasks; GPT-4 performed better in POS tagging and diacritization.	Extends prior evaluation by analyzing deeper syntactic reasoning.
	Almatham et al. (2025)	Introduced BALSAM benchmark covering 78 Arabic NLP tasks.	Provides standardized evaluation background; our work focuses specifically on syntax-level analysis.
	Maulidiya et al. (2024)	Compared ChatGPT and Gemini for Arabic *i^{c}rāb*.	Closely related but limited in scope; our study expands syntactic coverage.
	Antoun et al. (2020)	Introduced AraBERT pre-trained on Arabic corpora.	Serves as transformer baseline for Arabic modeling.
Grammar Parsing Research	Saadiyeh et al. (2025)	Comparative study of Arabic parsing approaches.	Provides background for evaluating LLM reasoning beyond traditional parsers.

Category	Study	Short Summary	Relation to Our Research
	Green and Manning (2010)	Studied Arabic constituency parsing challenges.	Highlights structural issues relevant to syntax evaluation.
	Sharefah et al. (2023)	Evaluated BERT-based dependency parsing models.	Shows effectiveness of fine-tuned transformers compared to generative LLMs.
I^{c}rāb Morphology Research	Kharsa et al. (2024)	Proposed BERT-based Arabic diacritization model.	Supports transformer capacity for modeling morpho-syntax needed in $i^{c}rāb$.
	Maulidiya et al. (2024)	Focused on Arabic $i^{c}rāb$ of nominative nouns.	Directly aligned but narrower; our study broadens evaluation.

3.3 Conclusion

Previous studies demonstrate significant progress in the field of Arabic Natural Language Processing, particularly with the adoption of transformer-based architectures and large language models. Research has achieved promising results in tasks such as syntactic parsing, diacritization, and linguistic analysis, despite the persistent challenges posed by the morphological and syntactic structure of the Arabic language.

These studies provide an important foundation for the current research, which focuses on evaluating the ability of modern language models to perform accurate Arabic grammatical analysis and deeper syntactic understanding.

Chapter 4

Methodology

4.1 Introduction

Arabic is considered one of the most complex languages in terms of syntax and morphology, especially in the task of *I'rab* (Arabic grammatical parsing), which requires a precise understanding of sentence structure and grammatical relationships between words. I'rab does not depend only on the meaning of a word, but is also influenced by its position within the sentence, the overall context, and the grammatical factors affecting it.

The difficulty of this task increases due to several factors, including:

- The rich morphological structure of the Arabic language and its numerous derivations.
- The presence of diacritics, which may change both meaning and grammatical analysis.
- The diversity of syntactic structures, including nominal sentences, verbal sentences, and various grammatical styles.
- The large number of grammatical rules and their exceptions.

Despite the significant progress achieved by Large Language Models (LLMs) in tasks such as text generation, translation, and question answering, their ability to accurately apply strict grammatical rules related to Arabic I'rab remains uncertain. This task requires high linguistic precision rather than general language understanding alone.

Therefore, the main problem addressed in this project is evaluating the capability of current LLMs and AI tools to perform accurate Arabic I'rab and identifying their strengths and limitations when dealing with different syntactic structures.

4.2 Requirements Analysis

4.2.1 Functional Requirements

The proposed evaluation system must provide a set of functional requirements to ensure the proper assessment of Large Language Models in Arabic I'rab tasks. The system should be capable of handling the complete evaluation workflow, from receiving input sentences to generating final performance reports.

The main functional requirements are as follows:

- Accept Arabic sentences as input from the prepared dataset.
- Send the input sentences to the selected Large Language Models or AI engines for grammatical analysis.
- Receive the generated I'rab output from each evaluated model.
- Normalize and preprocess the generated outputs to ensure consistency in formatting and terminology.
- Compare the generated outputs with the gold standard annotations.
- Calculate evaluation metrics such as accuracy, similarity score, and error rate.
- Store evaluation results for further analysis.
- Generate reports and visualizations to present model performance comparisons.

These functional requirements define the core operations of the proposed system and ensure an organized evaluation process for Arabic syntactic analysis.

4.2.2 Non-functional Requirements

In addition to functional requirements, the proposed system must satisfy a set of non-functional requirements that ensure performance quality and the effectiveness of the evaluation process. These requirements do not describe specific system functions, but rather the characteristics that the system must exhibit during operation.

The main non-functional requirements are as follows:

- **Accuracy:** The system must provide precise and reliable evaluation results when comparing model outputs with the gold standard annotations.
- **Scalability:** The system should be capable of handling a large number of Arabic sentences and allow the integration of new models in the future without affecting performance.
- **Response Time:** The process of sending sentences and receiving outputs from the language models should be performed within a reasonable time to ensure efficient evaluation.
- **Reproducibility:** The system must allow the same experiments to be re-executed using the same data and configurations in order to obtain comparable results.
- **Usability:** The system should be well-organized and easy to use, facilitating the testing process and the analysis of results.
- **Reliability:** The system must operate in a stable manner without data loss or errors that could affect the evaluation results.

These non-functional requirements contribute to building an effective and reliable evaluation system for analyzing the performance of Large Language Models in Arabic I'rab tasks.

4.3 Evaluation Architecture

The proposed system is designed as an experimental evaluation framework aimed at assessing the performance of Large Language Models (LLMs) in Arabic I'rab (syntactic

analysis). Unlike traditional systems that focus on developing standalone applications, this framework follows a structured pipeline architecture to ensure consistency, reproducibility, and reliability of the evaluation process.

The overall workflow of the system can be summarized as follows:

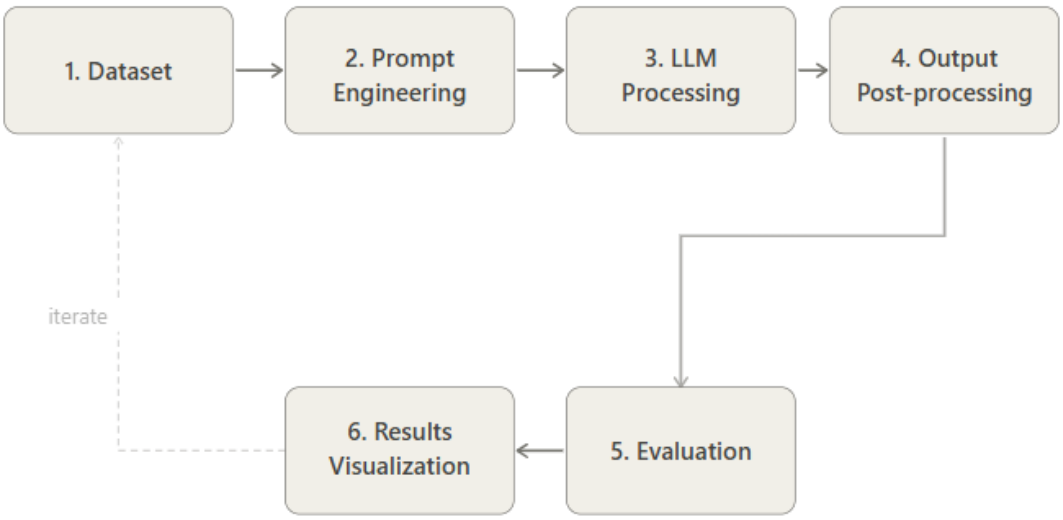


Figure 4.1: Evaluation architecture pipeline

4.3.1 Dataset Module

To perform the evaluation process, we will require a dataset consisting of Arabic sentences accompanied by their correct grammatical analysis (Gold Standard), which will serve as the reference for comparing the outputs generated by the evaluated language models.

The dataset will include different types of Arabic sentences, such as:

- Nominal sentences.
- Verbal sentences.
- Interrogative sentences.
- Sentences with both simple and complex syntactic structures.

Each dataset instance will contain:

- The original Arabic sentence.

- The reference grammatical analysis (Gold Standard).
- A unique identifier to facilitate tracking and evaluation.

An example of a dataset entry is shown below:

Sentence	ذَهَبَ الطَّالِبُ إِلَى الْمَدْرَسَةِ.
Reference I'rab	ذَهَبَ: فعل ماضٍ، الطَّالِبُ: فاعل مرفوع، إِلَى: حرف جر، الْمَدْرَسَةِ: اسم مجرور.

Table 4.1: Example of a Gold Standard Dataset Entry

4.3.2 Prompt Engineering Module

This module is responsible for designing the prompts used to query the LLMs. Since model outputs are highly sensitive to input formulation, careful prompt engineering is applied to:

- enforce complete grammatical analysis;
- constrain the output format (e.g., JSON);
- reduce ambiguity in responses.

This step is essential for ensuring consistency and fairness across different models, as illustrated in Figure 4.2.

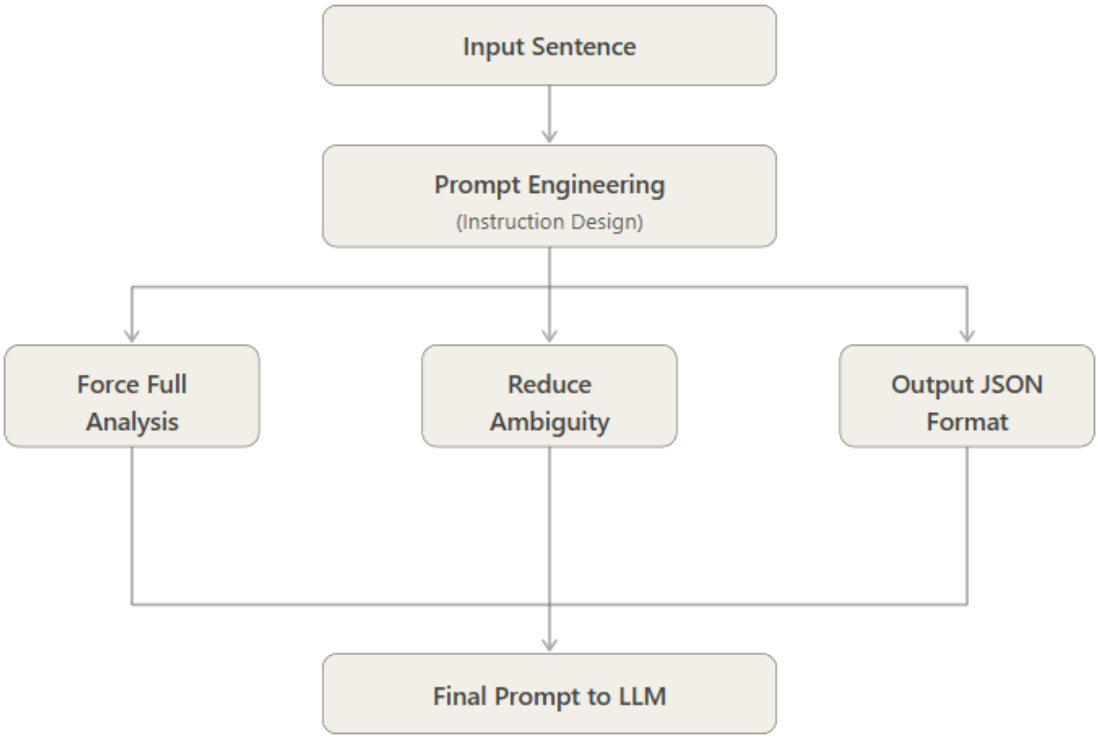


Figure 4.2: Prompt Engineering Module Workflow for I'rab Evaluation

4.3.3 LLM Processing Module

In this stage, Arabic sentences are submitted to selected Large Language Models for syntactic analysis. Multiple models are evaluated in parallel to enable benchmarking and comparison.

Each model generates I'rab outputs based on the provided prompts, allowing the system to assess their capability in handling Arabic syntax and morphology.

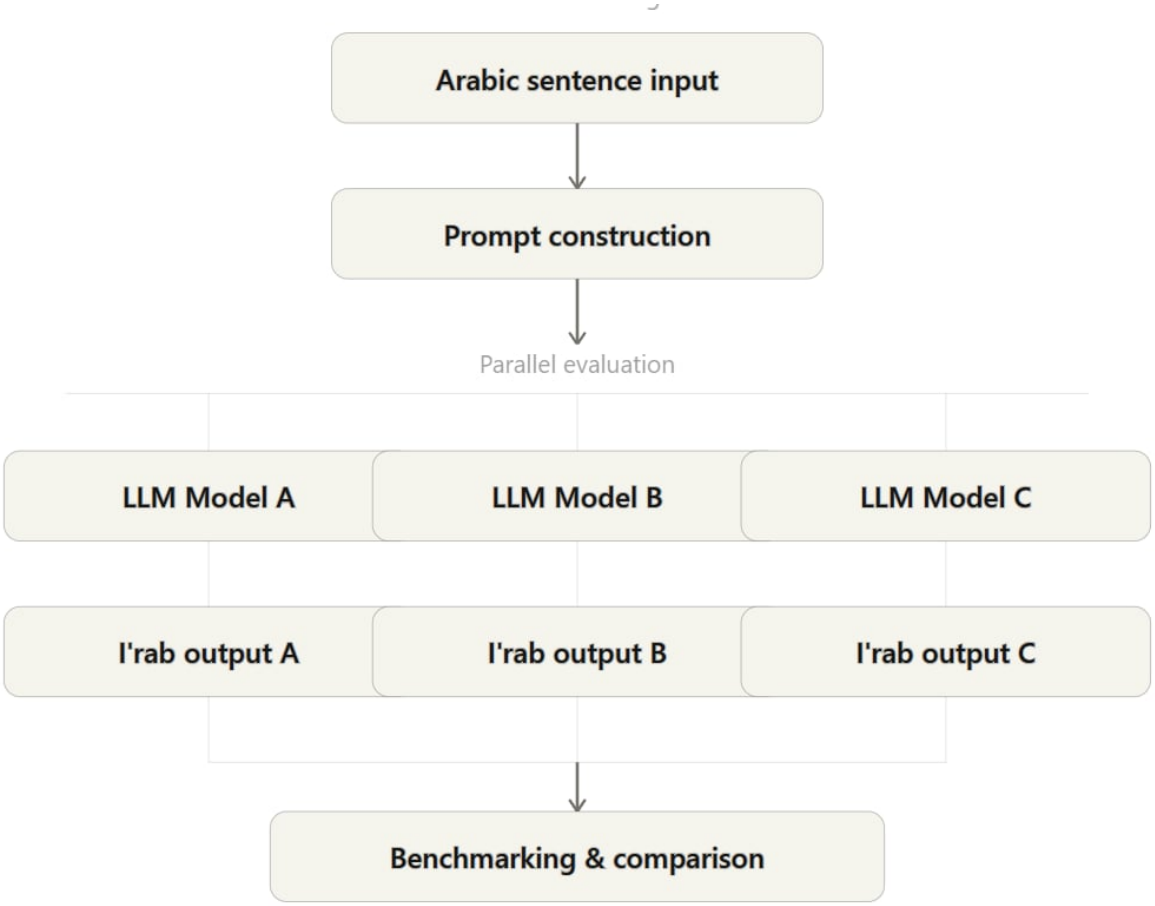


Figure 4.3: LLM Processing Module Workflow

4.3.4 Output Post-processing Module

The outputs generated by LLMs often vary in format and terminology. This module standardizes and normalizes these outputs to make them comparable.

The process includes:

- cleaning inconsistent or malformed outputs;
- converting results into a unified structured format;
- applying normalization rules to grammatical labels.

This step ensures that evaluation is not affected by formatting differences.

4.3.5 Evaluation Module

This module represents the core of the system. It compares model-generated outputs with the gold standard annotations.

The evaluation process includes:

- token-level comparison;
- grammatical label matching;
- computation of metrics such as accuracy and error rate;
- analysis of error patterns.

This allows an objective assessment of each model's performance in Arabic I'rab tasks.

4.3.6 Results Visualization Module

This final module is responsible for presenting the evaluation results in a clear and interpretable manner. The outputs include:

- comparison tables showing the performance of different models according to the adopted evaluation metrics;
- performance charts illustrating results in terms of Accuracy, Recall, Precision, and F1-score;
- analytical reports summarizing the strengths, weaknesses, and common error patterns of each model.

For example, a comparison table may indicate that GPT-4 achieved an overall Accuracy of 92%, while another model achieved 85%. Bar charts can also be used to visualize performance differences among models across various types of Arabic sentences.

An example of the expected results table is shown below:

Table 4.2: Example of Expected Evaluation Results

Model	**Accuracy**
GPT-4	92%
Gemini	88%
Claude	85%

4.4 Dataset Design

The dataset represents the core component of the proposed evaluation framework, serving as the reference benchmark for comparing and analyzing the outputs of Large Language Models in the Arabic I'rab task.

Its design aims to provide diverse and reliable Arabic sentences that ensure a fair and comprehensive evaluation of model performance during the subsequent stages of the study.

4.4.1 Data Sources

The evaluation dataset will be built using reliable Arabic linguistic resources, including the Quranic Arabic Corpus, Universal Dependencies Arabic Treebanks, CAMeL Lab resources, and Tashkeela Corpus. These resources provide Arabic sentences with various levels of morphological and syntactic annotation. The collected data are expected to include Arabic sentences, grammatical structures, morphological information, and syntactic analyses that can be used to construct a validated Gold Standard dataset for evaluating the selected Large Language Models.

4.4.2 Dataset size

The dataset will consist of a sufficiently large and diverse collection of Arabic sentences to ensure a representative and reliable evaluation of the selected Large Language Models. The number of sentences will be carefully selected to cover a wide range of grammatical rules, syntactic structures, and linguistic phenomena in Arabic. In addition, the dataset will be systematically categorized according to multiple dimensions to ensure comprehen-

sive coverage of linguistic variability. First, sentences will be classified based on syntactic type, including nominal sentences (الجمل الاسمية) and verbal sentences (الجمل الفعلية). Second, they will be grouped according to structural complexity, ranging from simple sentences (الجمل البسيطة) to more complex and compound constructions, in order to assess the models' ability to handle increasing levels of syntactic difficulty. Third, the dataset will include classification according to linguistic register and source, covering Modern Standard Arabic (العربية الفصحى الحديثة), poetic sentences (الجمل الشعرية), and Quranic sentences (الجمل القرآنية), which represent classical and highly structured linguistic forms. This multi-dimensional categorization is essential to evaluate model robustness across different linguistic contexts, since each category presents distinct challenges in terms of morphology, syntax, ambiguity resolution, and contextual interpretation. The inclusion of modern, literary, and religious texts is motivated by the need to test the models on both everyday language usage and highly formal or stylistically complex Arabic, thereby providing a comprehensive benchmark for Arabic syntactic analysis (I'rāb).

4.5 Data Preprocessing

Before using the dataset in the evaluation process, preprocessing steps will be applied to address issues such as inconsistent formatting, duplicate entries, and noisy data. These problems may affect the fairness and accuracy of the evaluation. Therefore, the data will be cleaned, standardized, and organized into a consistent format to ensure reliable and accurate results.

The preprocessing stage will include the following steps:

- **Data Cleaning:** Irrelevant, incomplete, or noisy sentences will be removed or corrected, including spelling errors and unnecessary symbols.
- **Data Structuring:** The dataset will be converted into structured formats such as JSON or Excel in order to facilitate automated processing and enable the integration of model outputs with the reference data during the evaluation process.
- **Token Alignment:** Sentences will be tokenized into words, and each token will be aligned with its corresponding grammatical annotation to ensure correct mapping

between words and their I'rab labels.

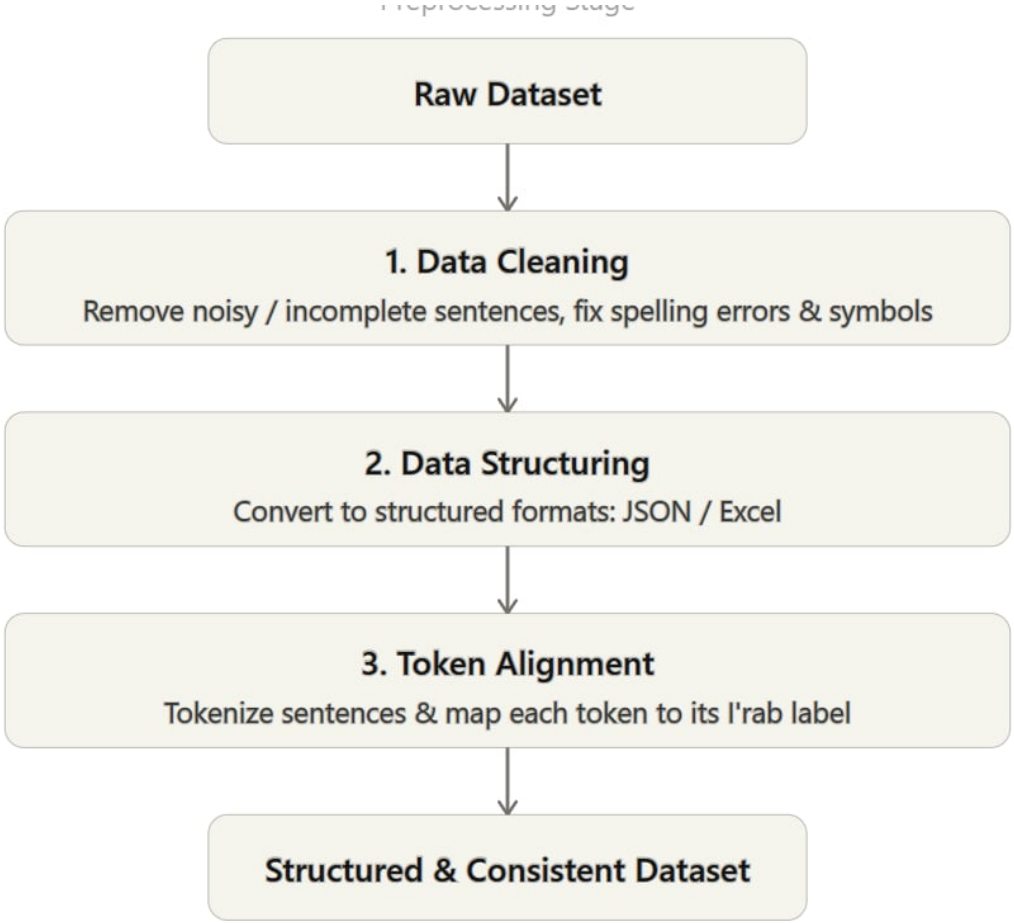


Figure 4.4: Data Preprocessing pipline

These steps ensure that the dataset is properly structured and consistent, enabling a fair and reliable evaluation of Large Language Models in Arabic I'rab tasks.

4.6 Comparison Methodology

The comparison methodology will be designed to ensure a fair and systematic evaluation of different Large Language Models on Arabic I'rab tasks. All models will be assessed under identical conditions in order to guarantee consistency and reproducibility of the results.

The comparison process will include the following steps:

- **Prompt Design:** We will design standardized prompts that clearly instruct the models to perform full Arabic grammatical analysis in a consistent output format.
- **Using the Same Input for All Models:** We will provide the same set of Arabic sentences to all evaluated models to ensure a fair comparison.
- **Extracting Model Outputs:** We will collect and structure the outputs generated by each model for further processing.
- **Comparing Outputs with Gold Labels:** We will compare the predicted outputs with the gold standard annotations to measure correctness.
- **Performing Error Analysis:** We will analyze the types of errors made by each model in order to identify weaknesses and performance gaps.

4.7 Conclusion

This chapter presented the proposed framework for evaluating Large Language Models in Arabic I'rab tasks. It described the system requirements, evaluation architecture, dataset design, data preprocessing procedures, and comparison methodology.

The proposed framework aims to ensure a fair, reliable, and reproducible evaluation process through standardized datasets, unified prompts, and objective evaluation metrics. The implementation of this framework will help assess the strengths and limitations of different LLMs in Arabic syntactic analysis and provide valuable insights for future Arabic NLP research.

Chapter 5

Evaluation and Results

5.1 Introduction

This chapter presents the experimental analysis of the study, which represents the practical phase in which the performance of Large Language Models (LLMs) is evaluated in the Arabic I'rab task. It follows the theoretical and methodological chapters, where the previously introduced concepts and procedures are translated into a practical implementation based on real data and model outputs.

This chapter focuses on describing the adopted evaluation methodology, including dataset preparation, model selection, and the definition of the performance metrics used for evaluation. It also presents the experimental results and analyzes them in a systematic manner to better understand the behavior of the models when dealing with different Arabic linguistic structures.

In addition, this chapter aims to interpret and discuss the results in order to highlight the effectiveness of the models in performing the I'rab task and to identify their strengths and weaknesses. This contributes to providing a clear perspective on the ability of these models to process Arabic from a syntactic standpoint.

5.2 Work Environment and Tools Used

To implement the proposed system and evaluate the performance of Large Language Models (LLMs) on the task of Arabic I'rab, a development environment based on Python was adopted due to its powerful capabilities in Natural Language Processing (NLP) and data analysis.

5.2.1 Development Environment

The different stages of the project were implemented using the Python programming language within the Jupyter Notebook environment, which facilitated interactive code development, testing, and result analysis. In addition, Microsoft Excel files and Google Sheets were used to organize data and store model outputs, helping with data sharing, collaborative organization, and basic verification and editing tasks.

The experiments relied on the web-accessible versions of the language models without using APIs, since most APIs were paid services.

5.2.2 Used Language Models

The study included a set of modern language models in order to evaluate their performance on the task of Arabic I'rab, namely:

GPT, Gemini, Qwen, Jais, Fanar, Claude, Mistral, DeepSeek, Al-maqalah, I3ribly.

These models were selected due to differences in their architectures and the diversity of their training data, allowing for a comprehensive comparison in Arabic language processing.

In addition, several other systems and models were initially considered for evaluation but were excluded for technical and methodological reasons, as explained below:

- The Araby AI platform was excluded because of the limitations of its free usage plan, which only allows a single conversation session. This was insufficient for conducting multiple experiments on the dataset.

- The Falcon model was also tested; however, it was found unsuitable for the Arabic I'rab task. Its interface was relatively complex, and the model often failed to follow the required task, generating translations or interpretations instead of syntactic analysis.

- The study also references several Arabic-focused language models, such as AraBERT, AraGPT, MARBERT, CAMeL-BERT, and Arabic-LLaMA. However, these models were not experimentally evaluated because they are primarily research-oriented models that require specialized environments and technical setup rather than being directly accessible end-user tools. Therefore, they were included only within the theoretical background, with the possibility of future experimentation.

- Some Arabic AI-based parsing platforms were also excluded from the full experimental evaluation, such as I3rbly, Ajroum, and Almaqalah. Although these systems rely on artificial intelligence techniques, they were not fully compatible with the evaluation methodology adopted in this study. Unlike general-purpose LLMs, these platforms do not allow flexible prompting interaction, since the prompting mechanism is internally predefined within the system itself.

- In addition, these platforms impose practical limitations on input size, as they generally accept only one or two sentences at a time. This made large-scale evaluation on the complete dataset difficult. Therefore, only a limited preliminary test involving approximately 10 sentences was conducted on Almaqalah and I3rbly in order to obtain an initial assessment of their performance.

- Another limitation concerned data preparation and result extraction, as these systems do not provide outputs in structured machine-readable formats such as JSON, which complicated the automation of evaluation and comparison processes.

- As for Ajroum, it could not be effectively evaluated because the platform did not return parsing results when sentences were submitted. Furthermore, attempts to access some of its features through account registration were unsuccessful, preventing reliable experimentation. [1]

[1] All experiments were conducted between February and April 2026. The evaluation reflects the avail-

5.2.3 Techniques and Software Libraries

Several software libraries were used for data processing and result analysis, including:

- **Pandas** for data processing and organization.
- **NumPy** for numerical and statistical operations.
- **Matplotlib** and **Seaborn** for data visualization and graph generation.
- The **re** library for text processing and data cleaning.
- The dataset used in this study was developed within the framework of the "Yu'rab" يعرب project, proposed and supervised by Professor Dr. Taha Zerrouki. To ensure the quality and reliability of the data, all collected examples underwent manual review and verification by an expert in Arabic grammar and syntactic analysis (I'rab).

These tools contributed to building an experimental system capable of analyzing and comparing the performance of language models in a structured manner.

5.3 Data Restructuring

In the first stage, the dataset underwent a restructuring and cleaning process in order to facilitate processing and evaluation. The original representation contained the complete I'rab of each sentence within a single field, where each word was followed by its grammatical annotation separated by a colon (:), as illustrated in figure 5.1.

ability and accessibility of the tested platforms during that period. Ajroum and Araby AI could not be assessed due to service unavailability or access limitations and were therefore excluded from the experimental comparison.

Sentence	Complete I‘rab
نَامَ عُمَرُ بَاكِرًا	نَامَ: فعلٌ ماضٍ مبنيٌّ على الفتحِ الظاهرِ على آخِرِهِ. عُمَرُ: فاعلٌ مرفوعٌ، وعلامةُ رفعهِ الضمةُ الظاهرةُ على آخرِهِ. بَاكِرًا: مفعولٌ فيهِ، ظرفُ زمانٍ منصوبٌ، وعلامةُ نصبهِ الفتحةُ الظاهرةُ على آخرِهِ.

Figure 5.1: Example of Complete I‘rab

Therefore, the colon separator was used to split each sentence into its fundamental components and transform the dataset into a more structured format based on tokenization. In this representation, each token is linked to its corresponding I‘rab annotation (Gold Label), as shown in Table 5.1.

Table 5.1: Token-Level Dataset Representation

Sentence	Token	Gold Label
نَامَ عُمَرُ بَاكِرًا	نَامَ	فعلٌ ماضٍ مبنيٌّ على الفتحِ الظاهرِ على آخِرِهِ.
نَامَ عُمَرُ بَاكِرًا	عُمَرُ	فاعلٌ مرفوعٌ، وعلامةُ رفعهِ الضمةُ الظاهرةُ علَى آخرِهِ.
نَامَ عُمَرُ بَاكِرًا	بَاكِرًا	مفعولٌ فيهِ، ظرفُ زمانٍ منصوبٌ، وعلامةُ نصبهِ الفتحةُ الظاهرةُ علَى آخرِهِ.

Afterward, several preprocessing and cleaning operations were performed to ensure data quality and evaluation stability. These operations included:

- Correcting minor spelling inconsistencies.
- Verifying the alignment between each sentence and its corresponding reference I‘rab.
- Ensuring sentence clarity and removing errors that could negatively affect model performance during syntactic analysis.

This restructuring process enabled evaluation at the token level rather than at the full sentence level, resulting in more precise analysis of model outputs.

5.3.1 Sentence Classification

To enable a more comprehensive analysis, the sentences in the dataset were classified according to several criteria aimed at studying the impact of different linguistic properties on model performance in the I'rab task.

By Sentence Type Sentences were divided into:

- Nominal sentences.
- Verbal sentences.

Examples:

Sentence	**Type**
أُحِبُّ وَطَنِي	Verbal
الكَرِيمُ مَحْبُوبٌ	Nominal

By Complexity Sentences were classified as:

- Simple sentences.
- Complex sentences.

Examples:

Sentence	**Class**
سَافَرَ مُصْطَفَى لَيْلاً	Simple
جَاءَ اللَّذَانِ نَجَحَا	Complex

By Source Sentences were also categorized according to their source:

- Qur'anic sentences.
- Poetic sentences.

- Standard Modern Arabic sentences.

Examples:

Sentence	Category
يُسَبِّحُ لِلَّهِ مَنْ فِي الأَرْضِ	Qur'anic
أَمُرُّ عَلَى الدِّيَارِ دِيَارَ لَيْلَى أُقَبِّلُ ذَا الْجِدَارَ وَذَا الْجِدَارِ	Poetic
دَخَلَ اللاَّعِبُونَ الْمَلْعَبَ	Simple

This classification helped analyze the effect of sentence type, complexity, and source on the performance of language models in the I'rab task, enabling a more detailed and in-depth interpretation of the results.

As a result, a clean and well-structured dataset was obtained, forming a reliable foundation for evaluating and analyzing the performance of different language models.

5.4 Processing Pipeline

The proposed system relies on a structured processing pipeline composed of several sequential stages, starting from sentence selection from the dataset and ending with result analysis and comparison against the reference I'rab. This workflow helped standardize the evaluation process across all models under the same conditions, ensuring fairness and consistency.

Initially, a group of sentences (approximately 20 sentences) is selected from the preprocessed dataset and sent to the language model along with a prompt instructing it to perform word-by-word I'rab analysis. The model outputs are then returned in a structured JSON format containing the sentence and the grammatical analysis of each token separately.

For example, the output structure is as follows:

```
[ {   "sentence": "العب بالكرة",
  "i3rab": [
    {
      "token": "العب",
      "i3rab": "فعل أمر مبني على السكون، والفاعل ضمير مستتر تقديره أنت"
    },      {
      "token": "بالكرة",
      "i3rab": "الباء حرف جر مبني على الكسر لا محل له من الإعراب، والكرة اسم مجرور بالباء
وعلامة جره الكسرة الظاهرة"
    }   ] } ]
```

Afterward, the JSON outputs are extracted and transformed into structured tables to facilitate further processing and analysis. Cleaning and normalization operations are then applied in order to standardize the outputs of different models, since each model may present I'rab in a different format or style.

Finally, the generated I'rab is compared with the reference annotations (Gold Labels) stored in the dataset. Similarity scores are computed, followed by the calculation of model accuracy. The final results are then stored for subsequent analysis and comparison between models, and are visualized using charts and graphical representations that clearly illustrate the performance of each model.

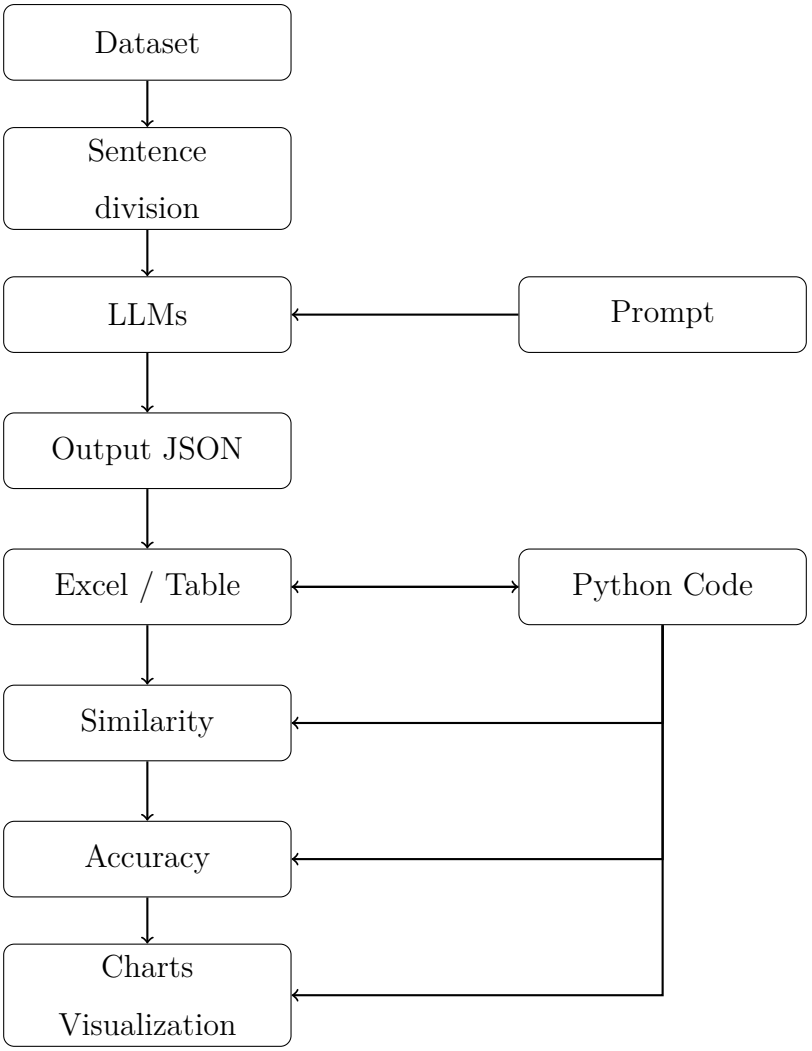


Figure 5.2: Overall workflow of the proposed evaluation system

5.4.1 Sending Sentences to the Models

The Arabic sentences were manually submitted to the language models through their available web interfaces, since paid APIs were not used in this study.

To ensure maximum consistency across results, a unified prompt was adopted. The prompt instructed the models to perform I'rab analysis and return the outputs in a structured JSON format, where each element contains the token and its corresponding I'rab annotation. The same prompt structure and instructions were maintained across all evaluated models to ensure fairness in the comparison process.

The following prompt was used during the experiments:

أعرب الجملة العربية إعراباً كاملاً.

القواعد:

1- ضع نجمة * قبل إعراب الجملة
(مثل: *الجملة الفعلية في محل رفع خبر)
ولا تضعها في token وحدها.

2- أعط الإعراب في JSON فقط.

3- كل عنصر يجب أن يحتوي:
token
i3rab

4- لا تعرب الكلمتين "قال تعالى"
أينما وجدت في بداية الجملة.

الجملة:

The results remain consistent even when the prompt is written in English, indicating that the models are robust to input language variation and produce equivalent outputs for both Arabic and English prompts. This standardized format significantly facilitated output processing and their later transformation into structured tables suitable for analysis and comparison.

5.4.2 Extracting I'rab Results

After obtaining the outputs generated by the language models, the I'rab corresponding to each token was extracted and transformed into a structured format suitable for comparison with the Gold Labels stored in the dataset. This process was implemented using Python code that converts JSON outputs into organized Excel tables containing the sentence, token, and corresponding grammatical annotation.

For example, some models returned outputs in the following format:

Claude Output Example


```
{
"يَحْكُمُ الْقَاضِي بِالْعَدْلِ," "sentence":
[ "i3rab":
{
"يَحْكُمُ," "token":
"فعل مضارع مرفوع وعلامة رفعه الضمة الظاهرة" "i3rab":
},
{
"الْقَاضِي," "token":
"فاعل مرفوع وعلامة رفعه الضمة المقدرة على الياء للثقل" "i3rab":
},
{
"بِالْعَدْلِ," "token":
"الباء حرف جر مبني على الكسر لا محل له من الإعراب، والعدل اسم مجرور بالباء" "i3rab":
وعلامة جره الكسرة، *الجار والمجرور متعلقان بالفعل يحكم"
}
]
}
```


After the initial transformation, the data becomes organized as follows:

Sentence	Token	Label
يَحْكُمُ الْقَاضِي بِالْعَدْلِ	يَحْكُمُ	فعل مضارع مرفوع وعلامة رفعه الضمة الظاهرة
يَحْكُمُ الْقَاضِي بِالْعَدْلِ	الْقَاضِي	فاعل مرفوع وعلامة رفعه الضمة المقدرة على الياء للثقل
يَحْكُمُ الْقَاضِي بِالْعَدْلِ	بِالْعَدْلِ	الباء حرف جر مبني على الكسر لا محل له من الإعراب، والعدل اسم مجرور بالباء وعلامة جره الكسرة، *الجار والمجرور متعلقان بالفعل يحكم

Table 5.2: Claude structured I'rab data after JSON transformation

This stage presented several challenges due to differences in how models generated I'rab outputs. Some models produced additional explanations, while others merged multiple words into a single element, making the outputs incompatible with the reference format

used in evaluation.

To address this issue, several preprocessing and cleaning operations were performed using Google Sheets in order to standardize the formatting and make the outputs closer to the Gold Labels. For example, some models produced outputs in the following form:

بالعدلِ → الباء حرف جر والعدل اسم مجرور وعلامة جره الكسرة الظاهرة على آخره

gold_token	jais_token	gold_label	jais_label
في	فِي الْبُكُورِ	حرف جر مبني.	*جار ومجرور في محل رفع خبر.
البكور	NOT_FOUND	اسم مجرور وعلامة جره الكسرة الظاهرة.	
الكتاب	الْكِتَابُ	مبتدأ مرفوع وعلامة رفعه الضمة الظاهرة.	مبتدأ مرفوع وعلامة رفعه الضمة الظاهرة على آخره.
في	فِي الْحَقِيبَةِ	حرف جر مبني لا محل له من الإعراب.	*جار ومجرور في محل رفع خبر.
الحقيبة	NOT_FOUND	اسم مجرور وعلامة جره الكسرة الظاهرة.	
البصر	الْبَصَرُ	مبتدأ مرفوع وعلامة رفعه الضمة الظاهرة.	مبتدأ مرفوع وعلامة رفعه الضمة الظاهرة على آخره.
من	مِن أَعْظَمِ النَّعَمِ	حرف جر مبني لا محل له من الإعراب.	جار ومجرور ومضاف إليه في محل رفع خبر.
أعظم	NOT_FOUND	اسم مجرور وعلامة جره الكسرة الظاهرة.	
النعم	NOT_FOUND	مضاف إليه مجرور وعلامة جره الكسرة الظاهرة.	

Table 5.3: Examples of token merging and missing outputs in Jais model

These outputs were then separated and reorganized into the following format:

بـ → حرف جر

العدلِ → اسم مجرور بالباء وعلامة جره الكسرة الظاهرة على آخره

gold_token	jais_token	gold_label	jais_label
في	في	حرف جر مبني.	حرف جر مبني على السكون لا محل له من الإعراب.
البكور	البكور	اسم مجرور وعلامة جره الكسرة الظاهرة.	اسم مجرور بـ "في" وعلامة جره الكسرة الظاهرة على آخره.
الكتاب	الْكِتَابُ	مبتدأ مرفوع وعلامة رفعه الضمة الظاهرة.	مبتدأ مرفوع وعلامة رفعه الضمة الظاهرة على آخره.
في	في	حرف جر مبني لا محل له من الإعراب.	حرف جر مبني على السكون لا محل له من الإعراب.
الحقيبة	الحقيبة	اسم مجرور وعلامة جره الكسرة الظاهرة.	اسم مجرور بـ "في" وعلامة جره الكسرة الظاهرة على آخره.
البصر	الْبَصَرُ	مبتدأ مرفوع وعلامة رفعه الضمة الظاهرة.	مبتدأ مرفوع وعلامة رفعه الضمة الظاهرة على آخره.
من	من	حرف جر مبني لا محل له من الإعراب.	حرف جر مبني على السكون لا محل له من الإعراب.
أعظم	أعظم	اسم مجرور وعلامة جره الكسرة الظاهرة.	اسم مجرور بـ "مِن" وعلامة جره الكسرة الظاهرة على آخره، وهو مضاف.
النعم	النعم	مضاف إليه مجرور وعلامة جره الكسرة الظاهرة.	مضاف إليه مجرور وعلامة جره الكسرة الظاهرة على آخره.

Table 5.4: Comparison between Gold Labels and Jais outputs

This preprocessing step contributed significantly to standardizing the dataset format and facilitating automated comparison between model outputs and the reference I'rab during the evaluation phase.

5.4.3 Storing and Analyzing Results

After completing the extraction and cleaning of model outputs, the results were stored in Microsoft Excel files and Google Sheets in a structured manner. Each record contained the original sentence, the token, the reference annotation (Gold Label), and the outputs generated by each language model. This process was implemented using Python code that merges the reference annotations with the outputs of each model, producing a sep-

arate Excel file for every evaluated model containing all the information required for the evaluation process.

In addition, sentence-level I'rab was separated from token-level I'rab into an independent column. This separation relied on the asterisk symbol (*) that the models were instructed to place before sentence-level grammatical analysis in the prompt. For example, some outputs were returned in the following format:

label
فعل ماضٍ مبني على الفتح المقدر على الألف، وألف الاثنين ضمير متصل في محل رفع فاعل، *جملة الصلة لا محل لها من الإعراب

Figure 5.3: Separation between token-level and sentence-level I'rab

After preprocessing, the data became organized as follows:

label	**sentence_i3rab**
فعل ماضٍ مبني على الفتح المقدر على الألف، وألف الاثنين ضمير متصل في محل رفع فاعل	جملة الصلة لا محل لها من الإعراب

Figure 5.4: Separation between token-level and sentence-level I'rab

This separation facilitated the evaluation process, especially when comparing token-level grammatical analysis independently from sentence-level or structural grammatical analysis.

As a result, the final table for each model contained the sentence, token, reference annotation, model-generated annotation, and sentence-level I'rab when available. This organization simplified the computation of similarity scores between model outputs and the reference annotations.

An example of the final table is shown below:

Sentence	Token	Gold Label	Model Label	Sentence I‘rab
نَامَ عُمَرُ بَاكِرًا	نَامَ	فعلٌ ماضٍ مبنيٌّ على الفتح الظاهرِ على آخرِهِ	فعل ماضٍ مبني على الفتح الظاهر	
نَامَ عُمَرُ بَاكِرًا	عُمَرُ	فاعلٌ مرفوعٌ، وعلامةُ رفعهِ الضمةُ الظاهرةُ على آخرِهِ	فاعل مرفوع وعلامة رفعه الضمة الظاهرة	
نَامَ عُمَرُ بَاكِرًا	بَاكِرًا	مفعولٌ فيهِ، ظرفُ زمانٍ منصوبٌ، وعلامةُ نصبهِ الفتحةُ الظاهرةُ على آخرِهِ	ظرف زمان منصوب وعلامة نصبه الفتحة الظاهرة	الظرف في محل نصب مفعول فيه

Table 5.5: Example of the final structured evaluation table

Afterward, Python was used to analyze the results and compute the evaluation metrics. Libraries such as **Pandas** and **NumPy** were employed for data processing and organization, while **Matplotlib** and **Seaborn** were used to generate charts and statistical visualizations for comparing model performance.

The analysis process included comparing model outputs with the reference annotations, computing similarity scores and accuracy for each model, and studying the impact of sentence type, complexity, and source on the final results. This stage provided clear indicators regarding the strengths and weaknesses of each model in the task of Arabic I‘rab.

5.5 Evaluation Methodology

After extracting and organizing the outputs generated by the different language models, a hybrid evaluation methodology was developed to measure the degree of correspondence between the generated I‘rab and the reference annotations (Gold Labels). This methodology combines Natural Language Processing (NLP) techniques, linguistic normalization, semantic similarity comparison, and rule-based grammatical validation inspired by expert systems, in order to provide a more accurate and realistic evaluation of Arabic I‘rab outputs.

The importance of this methodology lies in the fact that language models do not follow

a unified style when generating grammatical analysis. A single grammatical meaning may be expressed through multiple linguistically correct formulations, which makes exact textual matching insufficient for evaluating output quality.

For example, language models may generate several syntactically correct formulations for the same token, as illustrated in the following examples:

Sentence	**Token**	**Gold Label**	**Model Output**	**Model**	**Observation**
اِسْتَعِينُوا بِالصَّبْرِ وَالصَّلَاةِ	ب	حرف جر مبني على الكسر لا محل له من الإعراب	الباء حرف جر مبني على الكسر لا محل له من الإعراب	Qwen	Minor difference in wording only
اِسْتَعِينُوا بِالصَّبْرِ وَالصَّلَاةِ	ب	حرف جر مبني على الكسر لا محل له من الإعراب	جار ومجرور متعلق بالفعل، الباء حرف جر	GPT	Additional grammatical explanation
اِسْتَعِينُوا بِالصَّبْرِ وَالصَّلَاةِ	ب	حرف جر مبني على الكسر لا محل له من الإعراب	الباء حرف جر	Fanar	Shortened grammatical analysis
وَكَانَ رَسُولًا نَّبِيًّا	و	حرف استئناف مبني على الفتح لا محل له من الإعراب	الواو حرف استئناف	DeepSeek	Some grammatical details omitted
حَضَرَ الْأُسْتَاذُ	حضر	فعل ماض مبني على الفتح الظاهر على آخره	فعل ماض مبني على الفتح	Mistral	Same meaning with reduced detail
حَضَرَ الْأُسْتَاذُ	الأستاذ	فاعل مرفوع وعلامة رفعه الضمة الظاهرة على آخره	فاعل مرفوع وعلامة رفعه الضمة	GPT	Difference in level of explanation

Table 5.6: Examples of variation in I'rab outputs between language models

These examples demonstrate that differences between model outputs do not necessarily indicate grammatical errors, but may simply reflect variations in phrasing style or level of detail.

5.5.1 Preprocessing and Text Normalization

To handle formulation differences between language models, a preprocessing system for Arabic text was developed based on cleaning and standardizing grammatical expressions before computing similarity scores.

A) Arabic Text Cleaning

A dedicated function was implemented to clean Arabic text from superficial orthographic variations that do not affect grammatical meaning. This process aimed to improve the accuracy of comparisons between the reference annotations and the outputs generated by language models. The normalization process included several linguistic preprocessing steps, the most important of which are the following:

- **Removing diacritics**, in order to avoid treating differences in vocalization as grammatical differences, since some models return fully diacritized text while others generate plain text without diacritics.

  Example:

  مرفوعٌ → مرفوع

- **Normalizing Hamza forms**, in order to reduce inconsistencies caused by the multiple orthographic forms of Hamza. Some language models confuse these forms even though they represent spelling variations rather than grammatical differences.

  Example:

  إعراب ، أعراب ، آعراب → اعراب

- **Converting Tā' Marbūṭa (ة) into Hā' (ه)**, in order to avoid the influence of minor spelling inconsistencies during textual comparison. Some models occasionally replace ة with ه even though the grammatical meaning remains unchanged.

  Example:

Sentence	Token	Gold Label	Gemini Label
أَيُّ طَالِبٍ جَاءَ؟	طَالِبٍ	مضاف إليه مجرور وعلامة جره الكسرة الظاهرة على آخره	مضاف إليه مجرور وعلامة جرة الكسرة

Figure 5.5: Example of Tā' Marbūṭa Normalization

In this example, the variation between **علامة** and **علامه**, as well as the omission of some grammatical details, did not affect the overall grammatical meaning. Therefore, normalization was applied to reduce the influence of such orthographic variations during similarity computation.

- **Converting Alif Maqṣūra (ى) into Yā' (ي)**, because some models may write ى instead of ي even though the grammatical meaning remains identical. During experiments, it was observed that several models used alternative spellings for the same grammatical term, which required normalization.

  Example:

Sentence	Token	Gold Label	Jais Label
مَا حَضَرَ إِلَّا زَيْدٌ	مَا	حرف نفي مبني على السكون	حرف نفى

Figure 5.6: Example of Alif Maqṣūra Normalization

In this example, the variation between **نفى** and **نفي** did not affect the grammatical meaning. Therefore, both forms were treated as equivalent during similarity computation.

- **Removing punctuation marks**, because punctuation does not provide significant grammatical information for the evaluation task and may introduce errors during text processing or tokenization. This issue negatively affects similarity computation, especially when punctuation marks are attached directly to words.

  For example, some models such as Fanar occasionally merged punctuation marks with grammatical descriptions inside the same token.

Sentence	Token	Gold Label	Fanar Label
ذَهَبْتُ فِي رِحْلَةٍ بَحْرِيَّةٍ بِرِفْقَةِ أَصْدِقَائِي	ذهبت	فعل ماض مبني على السكون لاتصاله بتاء الرفع المتحركة والتاء ضمير متصل مبني في محل رفع فاعل	فعل ماضٍ مبني على السكون لاتصاله بتاء الفاعل، والتاء ضمير متصل في محل رفع فاعل.

Figure 5.7: Example of Punctuation Inconsistencies

A similar issue appeared in cases where punctuation marks were directly attached to words, as in the following example:

Sentence	Token	Gold Label	Jais Label
ذَهَبْتُ فِي رِحْلَةٍ بَحْرِيَّةٍ بِرِفْقَةِ أَصْدِقَائِي	ب	الباء: حرف جر	الباء حرف جر

Figure 5.8: Example of Punctuation Attached to Tokens

In such cases, expressions such as:

الفاعل.
الجر.

may be treated differently from:

الفاعل
الجر

despite having identical grammatical meaning. Therefore, punctuation marks were removed during preprocessing to eliminate such superficial differences.

The normalization directions described above were selected based on practical observations during experimentation. It was noticed that language models frequently tend to replace ة with ه, use ي instead of ى, and confuse different Hamza forms. Consequently, the objective of this stage was not full orthographic correction, but rather the standardization of the most common output variations in order to reduce the impact of spelling inconsistencies on textual similarity computation.

B) Grammatical Normalization

The system relies on a custom grammatical normalization dictionary (`NORMALIZATION_MAP`) to unify different linguistic expressions that share the same grammatical meaning.

This normalization process covers the following mappings:

Original Term	**Normalized Form**
الضمة / الضم	ضمه
الفتحة / الفتح	فتحه
الكسرة / الكسر	كسره
ماضٍ / ماضي	ماض
الصفة / صفة	نعت
خبر المبتدأ	خبر
مضاف إليه	مضاف_اليه
حرف جر	جر
الواو عاطفة	عطف
الظاهرة على آخره	ظاهره
مبني على الفتح	مبني فتحه

Table 5.7: Grammatical normalization mapping

For instance, a language model may produce the following output:

فعل مضارع مرفوع بالضمة

while the gold reference (Gold Label) is:

فعل مضارع مرفوع، وعلامة رفعه الضمة الظاهرة على آخره

Therefore, normalization was applied to unify such differences.

Examples from Model Outputs

Example 1 (Jais)

Sentence	Token	Gold Label	Jais Label	Normalization Notes
يُسَبِّحُ لِلَّهِ مَنْ فِي الأَرْضِ	يسبح	فعل مضارع مرفوع، وعلامة رفعه الضمة الظاهرة على آخره	فعل مضارع مرفوع بالضمة	The terms "بالضمة" and "الضمة" were normalized to "ضمه", and the expression "الظاهرة على آخره" was normalized to "ظاهره".

Example 2 (Gemini)

Sentence	Token	Gold Label	Gemini Label	Normalization Notes
أَيُّ طَالِبٍ جَاءَ؟	طالب	مضاف إليه مجرور وعلامة جره الكسرة الظاهرة على آخره	مضاف إليه مجرور وعلامة جره الكسرة	The term "الكسرة" was normalized to "كسره", and the expression "الظاهرة على آخره" was normalized to "ظاهره".

Example 3 (Claude)

Sentence	Token	Gold Label	Claude Label	Normalization Notes
تَعْمَلُ المُمَرِّضَاتُ فِي المُسْتَشْفَى	في	حرف جر مبني	في حرف جر مبني لا محل له من الإعراب	The expression "حرف جر" was normalized to "جر", and the phrase "لا محل له من الإعراب" was removed because it does not affect the evaluation.

Example 4 (Mistral)

Sentence	Token	Gold Label	Mistral Label	Normalization Notes
لَعِبَ مَازِنٌ وَأَحْمَدُ لُعْبَةَ الشِّطْرَنْجِ	لعب	فعل ماض مبني على الفتحة الظاهرة على آخره	فعل ماض مبني على الفتح	The term "ماضٍ" was normalized to "ماض", the expression "مبني على الفتح" was normalized to "مبني فتحه", and the expression "الظاهرة على آخره" was normalized to "ظاهره".

Example 5 (DeepSeek)

Sentence	Token	Gold Label	DeepSeek Label	Normalization Notes
الْمُعَلِّمُ الْمُخْلِصُ يَحْتَرِمُهُ طُلَّابُهُ	المخلص	صفة مرفوعة وعلامتها الضمة	نعت للمعلم مرفوع وعلامة رفعه الضمة	The term "صفة" was normalized to "نعت", and the term "الضمة" was normalized to "ضمه".

Effect of Normalization on Similarity

For example, the gold label:

دَخَلَ : فعل ماض مبني على الفتح الظاهر على آخره

and the model output:

دَخَلَ : فعل ماضٍ مبني على الفتح

would not match exactly due to missing phrases such as "الظاهر على آخره" and minor morphological variations.

After normalization, both become:

Before Normalization	After Normalization
فعل ماض مبني على الفتح الظاهر على آخره	فعل ماض مبني فتحه ظاهره
فعل ماضٍ مبني على الفتح	فعل ماض مبني فتحه

After applying similarity metrics such as:

- Token Overlap
- TF-IDF Similarity
- Key Terms Coverage

the similarity score significantly increases from approximately:

```
0.65
```

to:

$$0.9318 \approx 93.18\%$$

This demonstrates that the model's output is grammatically correct, even when it is less detailed than the reference annotation.

5.5.2 Processing Complex Expressions

Since some grammatical expressions consist of multiple words, a dedicated mechanism was developed to handle compound phrases before tokenizing the text.

For example, the phrase:

مضاف إليه

is transformed into:

مضاف_اليه

This transformation preserves the semantic integrity of multi-word grammatical terms and prevents their incorrect splitting during text processing.

The approach also relies on ordering expressions from longest to shortest to avoid partial or incorrect replacements during preprocessing.

5.5.3 Removal of Non-Informative Words (Stop Words Removal)

A custom list of non-essential words that do not significantly contribute to grammatical meaning during comparison was created. These include function words and repetitive morphological markers such as:

على في من إلى وعلامة وعلامته رفعه نصبه جره آخره

For example, the sentence:

فاعل مرفوع وعلامة رفعه الضمة الظاهرة على آخره

is transformed into:

فاعل مرفوع ضمة ظاهرة

This reduction helps eliminate textual noise and focuses the comparison process on the core grammatical information.

5.5.4 Detecting Critical Grammatical Conflicts

To prevent the system from considering incorrect answers as correct due to high textual similarity, a dedicated mechanism was developed to detect **critical grammatical conflicts** (`CRITICAL_TERMS`).

Table 5.8 presents some examples of these decisive conflicts:

Term	Conflicting Terms
مرفوع	منصوب، مجرور
منصوب	مرفوع، مجرور
مجرور	مرفوع، منصوب
فاعل	مفعول، مبتدأ، خبر
مبتدأ	فاعل، خبر، مفعول
خبر	مبتدأ، فاعل
ماض	مضارع، أمر
مضارع	ماض، أمر
ظاهرة	مقدرة
مقدرة	ظاهرة

Table 5.8: Examples of critical grammatical conflicts used in `CRITICAL_TERMS`

Practical Examples of Conflict Detection

Sentence	Token	Gold Label	GPT Label
يَحْكُمُ الْقَاضِي بِالْعَدْلِ	الْقَاضِي	فاعل مرفوع وعلامة رفعه الضمة المقدرة على الياء	فاعل مرفوع وعلامة رفعه الضمة الظاهرة

Figure 5.9: Example of a critical conflict between **مقدرة** and **ظاهرة**

Example 1 Explanation: Although both outputs are highly similar, a critical grammatical contradiction exists between the terms:

ظاهرة ↔ مقدرة

According to the conflict dictionary (`CRITICAL_TERMS`), this difference is considered unacceptable because it changes the nature of the grammatical case marker.

Sentence	Token	Gold Label	GPT Label	Similarity
خَرَجَ مَا فِي الحَظِيرَةِ	مَا	اسم موصول مبني على السكون في محل رفع فاعل	اسم موصول في محل نصب مفعول به أول	0.0

Figure 5.10: Example of a critical conflict between **فاعل** and **مفعول**

Example 2 Explanation: In this case, a critical grammatical contradiction was detected in the syntactic role:

مفعول ↔ فاعل

Even if other parts of the analysis appear similar, this difference is considered a major conflict because it completely changes the grammatical function of the token within the sentence.

Therefore, the system directly assigns a similarity score equal to:

```
0.0
```

This prevents the evaluation mechanism from accepting the output as correct based solely on textual similarity.

5.6 Semantic and Statistical Similarity Computation

After completing normalization and conflict checking, the system computes the similarity score between texts using multiple evaluation metrics.

5.6.1 Token Overlap Similarity

This metric is based on comparing shared words between the reference annotation (Gold Label) and the model-generated output after applying preprocessing and normalization steps. It aims to measure how well the model covers the essential grammatical elements.

This approach relies on two main metrics:

- **Precision**: measures the proportion of correct words in the model output compared to all generated words.
- **Recall**: measures the proportion of reference words successfully covered by the model.

The harmonic mean between Precision and Recall is computed using the F1-score formula:

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

Example Gold Label:

فعل مضارع مرفوع وعلامة رفعه الضمة الظاهرة على آخره

GPT Output:

فعل مضارع مرفوع وعلامة رفعه الضمة والفاعل ضمير مستتر تقديره هو

1. Extracting Tokens After Normalization

Gold Label Tokens:

فعل، مضارع، مرفوع، علامة، رفعه، ضمة، ظاهرة، آخره

Total:

8 tokens

GPT Tokens:

فعل، مضارع، مرفوع، علامة، رفعه، ضمة، فاعل، ضمير، مستتر، تقديره، هو

Total:

11 tokens

2. Token Overlap

Common tokens:

فعل، مضارع، مرفوع، علامة، رفعه، ضمة

Shared tokens:

6 tokens

3. Precision Calculation

$$Precision = \frac{6}{11} = 0.54$$

4. Recall Calculation

$$Recall = \frac{6}{8} = 0.75$$

5. F1-score Calculation

$$F1 = \frac{2 \times 0.54 \times 0.75}{0.54 + 0.75} \approx 0.63$$

Final Result

- Precision = 0.54
- Recall = 0.75
- F1-score = 0.63

Error Interpretation

The model introduced additional incorrect information:

الفاعل ضمير مستتر تقديره هو

These tokens (فاعل، ضمير، مستتر، تقديره، هو) do not exist in the reference annotation, which reduced Precision.

This shows that while the model successfully captured the main grammatical structure, it introduced unnecessary additions, which negatively affected the final similarity score.

This evaluation method is important because it does not only reward correct matches, but also penalizes irrelevant or hallucinated grammatical elements.

5.6.2 Statistical Similarity Using TF-IDF

In addition to lexical overlap-based evaluation, a statistical approach was adopted to measure textual similarity between the reference annotation (Gold Label) and the model-generated output using the TF-IDF technique combined with Cosine Similarity.

The TF-IDF method transforms text into numerical vector representations, where each word or character sequence is represented according to its importance within the document. This allows similarity to be computed mathematically rather than relying on exact string matching.

In this study, instead of relying solely on full words, character-level n-grams were used to improve robustness against minor spelling or formatting variations.

The following configuration was adopted:

```
analyzer = 'char_wb'
ngram_range = (2, 3)
```

This means the system uses:

- 2-character sequences (2-grams)
- 3-character sequences (3-grams)

How TF-IDF Works Step by Step

Step 1: Splitting Words into Character n-grams

The system first decomposes words into smaller character sequences of length two or three.

For example, the word:

مرفوع

is transformed into:

2-grams:

مر، رف، فو، وع

3-grams:

مرف، رفو، فوع

This representation helps capture similarity even when slight variations in spelling or formatting exist between the reference and model output.

Step 2: Computing TF-IDF Weights

After extracting character n-grams, the system computes a weight for each n-gram using the TF-IDF technique.

Step 3: Converting Texts into Numerical Vectors

Once the weights are computed, each text is transformed into a numerical vector representing the importance of different character n-grams.

For example:

```
s1 = [0.3, 0.0, 0.7, 0.2, ...]
```

represents the reference annotation (Gold Label), while:

```
s2 = [0.3, 0.5, 0.7, 0.0, ...]
```

represents the model-generated output.

These values reflect the importance of each character n-gram within the text representation.

Step 4: Computing Cosine Similarity

After converting both texts into numerical vectors, the similarity between them is computed using the Cosine Similarity metric:

$$\text{Similarity} = \frac{s1 \cdot s2}{\|s1\| \times \|s2\|}$$

where:

- A value of **1.0** indicates that the vectors are nearly identical (angle = 0°).
- A value of **0.0** indicates no similarity between the texts (angle = 90°).

Why TF-IDF Is Useful Despite Its Small Weight

Although TF-IDF contributes only about **15%** to the final evaluation score, it plays an important role in detecting fine-grained lexical and morphological similarities that other metrics may miss.

For example:

مرفوع
مرفوعا

or:

نافية
ناقية

In such cases, character-level n-grams remain highly similar, resulting in a relatively high similarity score.

In contrast, token-based methods such as Token Overlap may consider these words different due to a single character variation.

Therefore, TF-IDF acts as a complementary mechanism that reduces the impact of minor orthographic variations and improves the robustness of the evaluation system, especially when dealing with outputs from language models that may differ in writing style while preserving the same grammatical meaning.

5.6.3 Key Terms Coverage

This metric aims to measure the ability of the language model to capture the essential grammatical elements present in the reference annotation (Gold Label), by focusing specifically on key grammatical terms that carry core syntactic meaning.

Unlike other similarity metrics, this measure does not give significant importance to additional explanatory words or secondary details. Instead, it focuses primarily on whether the essential grammatical concepts are present in the model output.

The metric is computed using the following formula:

$$\text{Key Coverage} = \frac{\text{Number of shared key terms}}{\text{Number of key terms in the reference annotation}}$$

For example, suppose the reference annotation is:

فاعل مرفوع

while the model output is:

فاعل مرفوع بالضمة

The system identifies that all essential grammatical terms from the reference annotation are present in the model output, namely:

فاعل
مرفوع

The additional term:

بالضمة

does not negatively affect the coverage score because it represents an additional detail rather than a core grammatical concept.

Therefore, the coverage score becomes:

$$\frac{2}{2} = 1.00$$

which corresponds to:

100%

This mechanism is particularly important in evaluating Arabic I'rab outputs because some language models tend to produce concise answers, while others generate more detailed

explanations. Despite these stylistic differences, the core grammatical meaning may still be correct in both cases.

Moreover, this metric helps focus the evaluation process on the correctness of essential grammatical roles rather than being overly influenced by stylistic variation or excessive explanatory content in the model output.

5.6.4 Comparison Between Similarity Metrics

The following table summarizes the differences between the three similarity metrics used in the evaluation system, including their computation methods, the type of similarity measured by each metric, and their role in evaluating I'rab outputs.

Metric	Computation Method	What Does It Measure?	Does It Penalize Extra Words?	Main Advantage
Token Overlap (F1-Score)	Based on Precision and Recall, followed by F1-score computation	Overall lexical similarity while balancing accuracy and coverage	Yes	Balances reference coverage while penalizing incorrect additional words
Key Terms Coverage	Shared key terms divided by reference key terms	Coverage of essential grammatical terms present in the reference annotation	Almost no	Focuses mainly on the presence of important grammatical elements
TF-IDF + Cosine Similarity	Transforms texts into numerical vectors, then computes Cosine Similarity	Lexical and orthographic similarity between texts	Partially	Detects similarity even with minor spelling variations such as: مرفوع and مرفوعٌ

Table 5.9: Comparison between similarity metrics used for evaluating I'rab outputs

5.6.5 Final Score Computation

After computing all previous metrics, they are combined into a weighted formula to produce the final similarity score:

$$\text{score} = (key\_cov \times 0.5 + token \times 0.35 + tfidf \times 0.15)$$

5.6.6 Explanation of Weight Selection

1. Key Terms Coverage (50%) This metric was assigned the highest weight because it represents the **core essence of I'rab analysis**.

- Arabic grammatical analysis primarily depends on correctly identifying grammatical functions such as: فاعل، مفعول، مبتدأ .
- If these core grammatical roles are correctly identified, the annotation is generally considered grammatically correct even when secondary details differ.

Therefore, this metric represents the most important criterion in determining the correctness of I'rab outputs.

2. Token Overlap Similarity (35%) This metric was assigned the second highest weight because it evaluates:

- The precision of the generated answer
- The absence of incorrect or unnecessary additional words

In other words, it is not sufficient for the generated words to be partially correct; the model should also avoid introducing grammatical elements that are not present in the reference annotation.

For this reason, the metric was given a moderate weight since it balances between:

- Precision
- Recall

3. Statistical Similarity Using TF-IDF (15%) This metric received the smallest weight because:

- It does not directly evaluate grammatical meaning
- Instead, it measures lexical and orthographic similarity between texts

Nevertheless, it remains important because:

- It improves the robustness of the evaluation system against minor spelling variations and writing inconsistencies

5.6.7 Acceptance Threshold

After computing the final similarity score between the reference annotation (Gold Label) and the model-generated output, an acceptance threshold mechanism was adopted to determine whether an answer should be classified as correct or incorrect.

Initially, several threshold values were experimentally tested in order to study their impact on the evaluation results. The tested values were as follows:

```
THRESHOLD = 0.85
THRESHOLD = 0.75
THRESHOLD = 0.65
THRESHOLD = 0.55
```

After conducting these experiments, a manual review and validation process was performed on a large number of classified outputs in order to compare the system decisions with human judgment.

The manual analysis showed that the threshold:

```
THRESHOLD = 0.55
```

was the most suitable value for this project. The vast majority of outputs exceeding this threshold were found to be grammatically and semantically correct, while incorrect classifications were relatively rare.

It was also observed that using higher thresholds such as:

```
0.85
```

leads to the rejection of a significant number of correct answers, especially in cases where the model's grammatical analysis differs in formulation from the reference (gold standard),

despite both sharing the same syntactic meaning. It is also observed that some responses are marked as incorrect solely due to being more concise than the gold label, even though the grammatical content is correct in essence.

For example, if the model's answer is: نام: فعل ماضي, while the reference annotation is: نام: فعلٌ ماضٍ مبنيٌّ على الفتح الظاهرِ على آخِرِهِ, then considering the model's answer as incorrect is not fully accurate, since both expressions represent the same grammatical judgment, differing only in the level of detail and linguistic elaboration.

Accordingly, the following rule was adopted within the evaluation system:

- If:

  ```
  Similarity Score >= 0.55
  ```

  the answer is considered correct.

- If:

  ```
  Similarity Score < 0.55
  ```

  the answer is considered incorrect or incomplete.

The choice of this threshold can be explained by the fact that the evaluation system relies on semantic similarity rather than exact textual matching. Consequently, medium similarity scores may still represent correct grammatical interpretations even when minor stylistic or linguistic differences exist between the two annotations.

5.6.8 Extraction of Statistical Evaluation Metrics

After evaluating all language model outputs using the semantic and statistical similarity methodology, a set of standard statistical evaluation metrics was extracted in order to provide a comprehensive and objective assessment of model performance.

The extracted metrics include:

- **Accuracy**
- **Precision**
- **Recall**
- **F1-Score**

Precision

Precision measures how accurate the model is in the answers it generates, that is, the proportion of correct answers among all generated answers.

It is computed using the following formula:

$$Precision = \frac{TP}{TP + FP}$$

where:

- **TP**: Number of correct answers (True Positives)
- **FP**: Number of incorrect answers (False Positives)

In this study, language models always generate an output even when they are uncertain about the correct grammatical analysis. Consequently, there are almost no cases of missing predictions or unanswered instances, meaning that the classification process is essentially reduced to determining whether the generated answer is correct or incorrect.

Recall

Recall measures the ability of the model to identify all correct answers present in the reference dataset.

It is computed as follows:

$$Recall = \frac{TP}{TP + FN}$$

where:

- **FN**: Cases that were not correctly identified or received no valid answer (False Negatives)

Since language models rarely produce empty outputs, the value of FN becomes nearly negligible:

$$FN \approx 0$$

Therefore, Recall becomes approximately:

$$Recall \approx 1 \quad (100\%)$$

This phenomenon is mainly caused by the tendency of language models to always generate outputs, even under uncertainty, a behavior commonly referred to as hallucination.

Accuracy

Accuracy measures the proportion of correct answers among all evaluated cases.

It is computed as follows:

$$Accuracy = \frac{TP}{TP + FP + FN}$$

Since:

$$FN \approx 0$$

the formula becomes approximately:

$$Accuracy \approx \frac{TP}{TP + FP}$$

which makes Accuracy numerically very close to Precision in this evaluation setting.

Methodological Note

Due to the nature of the evaluated language models, where empty outputs are almost nonexistent, Recall values became consistently high and nearly constant across experiments, while Accuracy and Precision produced very similar results in most cases.

Nevertheless, all evaluation metrics (Accuracy, Precision, Recall, and F1-Score) were retained because they provide complementary perspectives on model performance and remain consistent with standard evaluation practices in Natural Language Processing (NLP) research.

5.7 General Model Results

At this stage, the overall performance of the language models and Arabic parsing platforms was evaluated on the complete dataset after testing several different acceptance threshold values. The objective was to study the impact of threshold selection on the evaluation results.

The following statistical metrics were computed for each model:

- Accuracy
- Precision
- Recall
- F1-Score

It is important to note that Recall values were consistently very high across all models. This is mainly because generative language models and Arabic parsing systems almost always return an output, even when uncertain, which increases the number of generated responses and leads to some hallucinated or grammatically incorrect analyses.

In contrast, Precision and F1-Score were more representative of the actual quality of the generated I'rab outputs, since they evaluate how accurate the generated analyses are compared with the reference annotation.

5.7.1 Model Results at Threshold 0.55

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
GPT	65.35	66.88	96.62	79.04
Gemini	76.22	76.46	99.58	86.51
Qwen	75.66	75.83	99.72	86.15
Mistral	73.96	74.87	98.39	85.03
Fanar	69.00	69.22	99.54	81.66
Jais	74.26	74.33	99.86	85.23
DeepSeek	73.94	74.01	99.86	85.02
Claude	**77.13**	**77.21**	**99.86**	**87.09**

Table 5.10: Performance of Large Language Models at threshold 0.55

Large Language Models (LLMs)

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
I3rbly	**81.82**	**81.82**	100.00	**90.00**
plus90	56.82	56.82	100.00	72.46

Table 5.11: Performance of specialized Arabic parsing platforms at threshold 0.55

Specialized Arabic Parsing Platforms The results show that the **I3rbly** platform achieved the best overall performance among all evaluated systems, outperforming even most large language models. This finding highlights the effectiveness of specialized systems in the task of Arabic grammatical analysis (الاعراب).

5.7.2 Model Results at Threshold 0.65

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
GPT	58.48	59.85	96.23	73.80
Gemini	67.46	67.68	99.53	80.57
Qwen	**68.65**	**68.80**	99.69	**81.41**
Mistral	66.70	67.52	98.22	80.03
Fanar	59.29	59.48	99.47	74.44
Jais	58.48	59.85	96.23	73.80
DeepSeek	63.78	63.84	99.83	77.88
Claude	63.83	63.90	**99.83**	77.92

Table 5.12: Performance of Large Language Models at threshold 0.65

Large Language Models (LLMs)

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
I3rbly	**68.18**	**68.18**	100.00	**81.08**
plus90	50.00	50.00	100.00	66.67

Table 5.13: Performance of specialized Arabic parsing platforms at threshold 0.65

Specialized Arabic Parsing Platforms When the threshold was increased to:

```
0.65
```

the performance of most systems decreased due to the stricter evaluation criteria.

5.7.3 Model Results at Threshold 0.75

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
GPT	46.72	47.82	95.33	63.69
Gemini	55.47	55.64	99.43	71.36
Qwen	**60.68**	**60.81**	99.65	**75.53**
Mistral	57.02	57.72	97.92	72.63
Fanar	49.42	49.57	99.36	66.15
Jais	57.13	57.19	**99.81**	72.71
DeepSeek	51.49	51.54	99.79	67.98
Claude	50.80	50.85	99.79	67.37

Table 5.14: Performance of Large Language Models at threshold 0.75

Large Language Models (LLMs)

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
I3rbly	**59.09**	**59.09**	100.00	**74.29**
plus90	43.18	43.18	100.00	60.32

Table 5.15: Performance of specialized Arabic parsing platforms at threshold 0.75

Specialized Arabic Parsing Platforms

5.7.4 Model Results at Threshold 0.85

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
GPT	32.88	33.65	93.49	49.49
Gemini	38.00	38.13	99.17	55.08
Qwen	43.25	43.34	99.51	60.39
Mistral	**44.50**	**45.05**	97.35	**61.59**
Fanar	37.37	37.49	99.15	54.40
Jais	43.30	43.34	**99.75**	60.43
DeepSeek	37.82	37.86	99.72	54.88
Claude	37.87	37.91	99.72	54.94

Table 5.16: Performance of Large Language Models at threshold 0.85

Large Language Models (LLMs)

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
I3rbly	**38.64**	**38.64**	100.00	**55.74**
plus90	31.82	31.82	100.00	48.28

Table 5.17: Performance of specialized Arabic parsing platforms at threshold 0.85

Specialized Arabic Parsing Platforms The overall results demonstrate that system performance decreases progressively as the acceptance threshold increases. This decline is mainly caused by the stricter evaluation criteria, which reduce the number of accepted answers.

Manual inspection also revealed that many outputs rejected under higher thresholds were actually grammatically correct, differing only in wording style or level of detail. Therefore, the following threshold was adopted as the final evaluation threshold:

```
THRESHOLD = 0.55
```

This value provided the best balance between flexibility and precision, allowing the system to accept most genuinely correct answers while keeping the rate of incorrect acceptances very low.

It is important to note that the results obtained for the **I3rbly** and **plus90 (Almaqalah)** platforms cannot be directly compared with the large language models evaluated in this study. The evaluation of these specialized systems was conducted on a limited dataset consisting of only 10 sentences.

This limitation was mainly due to platform access restrictions and usage constraints encountered during the experiments.

Consequently, the obtained results provide only a preliminary indication of the performance of these systems in the task of Arabic grammatical analysis, rather than a comprehensive evaluation equivalent to the experiments performed on the larger and more diverse datasets used for the language models.

Nevertheless, the **I3rbly** platform demonstrated relatively strong performance even under this limited experimental setting, achieving the best results at the acceptance threshold:

```
0.55
```

This finding highlights the effectiveness of specialized Arabic grammatical systems in handling I'rab tasks compared with some general-purpose language models.

5.7.5 Analysis by Sentence Type

The sentences in the dataset were divided into two main categories: nominal sentences and verbal sentences, with the aim of studying the impact of syntactic structure on the performance of language models in the Arabic I'rab task. The results across different acceptance thresholds revealed a clear influence of sentence type on the achieved accuracy.

At the lower acceptance threshold (0.55), most models achieved better performance on verbal sentences compared to nominal sentences. The Claude model recorded the highest accuracy for verbal sentences, reaching **78.95%**, while it also achieved the best performance on nominal sentences with **74.24%**. This trend continued at the 0.65 threshold, where verbal sentences maintained a relative advantage for most models, with the Qwen model achieving the highest accuracy of **70.08%** for this sentence type.

When the threshold was increased to 0.75, the difference between the two sentence types became more pronounced. The Qwen model managed to maintain relatively strong perfor-

mance on verbal sentences compared to the other models, achieving **62.64%**. In contrast, some models such as Gemini, GPT, and DeepSeek began to show closer results or slight superiority for nominal sentences, reflecting the sensitivity of stricter evaluation thresholds to subtle differences in grammatical phrasing.

At the strictest threshold (0.85), accuracy generally decreased across all models due to the requirement for outputs to closely match the reference annotation. Nevertheless, some models such as Qwen, Jais, and Mistral maintained relatively balanced performance between the two sentence types. The highest accuracy at this threshold was achieved by Mistral on nominal sentences with **45.18%**.

The superiority of verbal sentences in most cases can be explained by their generally simpler and more direct syntactic structure. Verbal sentences often rely on a clear sequence consisting of a verb, subject, and object, making grammatical relations easier for language models to identify and process. In contrast, nominal sentences tend to exhibit greater structural complexity due to the variety of grammatical relationships they may contain, including subject and predicate constructions, embedded clauses functioning as predicates, adjectives, appositions, conjunctions, and multiple sentence components. Some nominal sentences may also involve ellipsis or long and compound structures, increasing the likelihood of confusion between grammatical functions.

Overall, the results confirm that sentence type is an influential factor affecting the quality of generated I'rab. However, the degree of this impact varies from one model to another depending on each model's ability to understand Arabic syntactic structures and process complex grammatical relationships.

Model	Sentence Type	55%	65%	75%	85%
Gemini	اسمية	72.81%	65.76%	56.83%	42.16%
Gemini	فعلية	78.61%	68.81%	54.95%	35.76%
Qwen	اسمية	73.38%	**66.62%**	**57.70%**	42.73%
Qwen	فعلية	77.26%	**70.08%**	**62.64%**	43.70%
Mistral	اسمية	71.51%	65.32%	55.11%	**45.18%**
Mistral	فعلية	76.84%	68.81%	59.26%	**44.97%**
GPT	اسمية	66.33%	59.14%	49.50%	38.13%
GPT	فعلية	67.20%	60.27%	46.83%	31.02%
DeepSeek	اسمية	72.81%	64.89%	54.24%	43.17%
DeepSeek	فعلية	74.73%	63.23%	49.96%	34.74%
Claude	اسمية	**74.24%**	60.29%	47.63%	36.83%
Claude	فعلية	**78.95%**	66.02%	52.75%	38.55%
Fanar	اسمية	68.06%	58.71%	49.64%	40.43%
Fanar	فعلية	69.91%	59.93%	49.54%	35.76%
Jais	اسمية	71.08%	65.04%	54.53%	**43.88%**
Jais	فعلية	76.25%	68.22%	55.37%	43.03%

Table 5.18: Model accuracy according to sentence type across different acceptance thresholds

5.7.6 Results of Arabic I'rab Platforms by Sentence Type

An additional analysis was conducted on the Arabic I'rab platforms **I3rbly** and **plus90** in order to study the impact of sentence type on the quality of the generated grammatical analysis. This experiment was performed using a small dataset consisting of only ten sentences; therefore, the obtained results should be considered preliminary indicators rather than definitive judgments regarding system performance.

The results showed the superiority of the **I3rbly** platform across most acceptance thresholds, particularly for nominal sentences, where it achieved the highest accuracy of **86.36%** at the 0.55 threshold. The platform also maintained relatively stable performance as the acceptance thresholds increased. In contrast, the **plus90** platform achieved relatively better results on verbal sentences at lower thresholds, but its performance decreased noticeably as the evaluation became more strict, especially at the 0.85 threshold.

Overall, these findings suggest that specialized Arabic grammatical analysis systems can achieve strong performance in certain I'rab scenarios. However, accurately evaluating such systems requires testing on larger and more diverse datasets in order to obtain more reliable conclusions.

Model	Sentence Type	55%	65%	75%	85%
I3rbly	Nominal	**86.36%**	**72.73%**	**63.64%**	40.91%
I3rbly	Verbal	77.27%	63.64%	54.55%	36.36%
plus90	Nominal	50.00%	50.00%	50.00%	**40.91%**
plus90	Verbal	63.64%	50.00%	36.36%	22.73%

Table 5.19: Accuracy of Arabic I'rab platforms according to sentence type across different acceptance thresholds

5.7.7 Analysis by Degree of Complexity

The sentences in the dataset were classified into two categories: simple sentences and compound (or complex) sentences, with the aim of studying the impact of syntactic complexity on the performance of language models in the Arabic I'rab task. The results across all evaluation thresholds showed a clear and consistent difference between the two categories, with models achieving significantly higher performance on simple sentences compared to complex ones.

At the 0.55 threshold, most models achieved high accuracy on simple sentences, exceeding **80%** in some cases, particularly for Gemini and Claude. In contrast, performance on complex sentences was noticeably lower. This gap persisted at the 0.65 and 0.75 thresholds, along with a gradual overall decline in performance. At the strictest threshold (0.85), the difference became even more pronounced, as accuracy for complex sentences dropped significantly, reaching approximately 30% in some cases.

The best performance on simple sentences at the 0.55 threshold was achieved by Gemini and Claude, both reaching **81.73%**, while Mistral recorded the highest accuracy for simple sentences at the 0.85 threshold with **48.74%**. For complex sentences, Qwen achieved the best relative performance at intermediate thresholds, despite the general decline observed across all models as evaluation strictness increased.

Model	Sentence Class	55%	65%	75%	85%
Gemini	بسيطة	**81.73%**	73.91%	60.77%	42.68%
Gemini	مركبة	67.39%	56.96%	46.81%	30.29%
Qwen	بسيطة	81.31%	**74.24%**	**66.58%**	47.64%
Qwen	مركبة	66.38%	59.42%	50.87%	35.94%
Mistral	بسيطة	79.97%	73.15%	62.12%	**48.74%**
Mistral	مركبة	66.09%	57.83%	50.14%	38.70%
GPT	بسيطة	70.29%	63.55%	51.52%	36.87%
GPT	مركبة	61.01%	53.48%	41.45%	28.12%
DeepSeek	بسيطة	77.86%	68.01%	55.22%	41.25%
DeepSeek	مركبة	67.39%	56.67%	45.22%	32.03%
Claude	بسيطة	**81.73%**	67.51%	54.04%	41.25%
Claude	مركبة	69.42%	57.68%	45.36%	32.17%
Fanar	بسيطة	73.15%	63.38%	53.45%	41.50%
Fanar	مركبة	62.46%	52.75%	42.90%	30.58%
Jais	بسيطة	78.11%	70.96%	59.18%	48.48%
Jais	مركبة	67.83%	60.29%	47.97%	34.49%

Table 5.20: Model accuracy according to sentence complexity across different acceptance thresholds

The general decline across all models can be explained by the nature of complex sentences, which include:

- Subordinate clauses (الجمل التابعة)
- Embedded syntactic structures (الجمل الواقعة في محل إعرابي)
- Coordinating and subordinating conjunctions (أدوات الربط والعطف)
- Multiple overlapping syntactic relations within a single sentence

Such structures require the model to track long-distance dependencies between words, which increases the difficulty of correctly assigning grammatical roles.

In contrast, simple sentences typically exhibit more direct syntactic structures and clearer grammatical relations, making them easier for models to analyze accurately.

Overall, the results confirm that syntactic complexity is a key factor affecting Arabic parsing performance, and that model accuracy decreases as sentence complexity increases.

5.7.8 Results of Arabic I'rab Platforms by Sentence Complexity

An additional analysis was conducted on the performance of the **I3rbly** and **plus90** platforms according to sentence complexity (بسيطة / مركبة). The results generally show that simple sentences are easier to process than complex sentences for both systems.

The **I3rbly** platform demonstrates clear superiority and greater stability across all thresholds, achieving its highest performance on simple sentences (**90.91%** at 0.55). Although its performance decreases gradually as the evaluation becomes more strict, it consistently outperforms plus90 in all cases. In contrast, the **plus90** platform shows moderate performance with greater degradation on complex sentences, particularly at higher thresholds, where accuracy drops to **27.27%** at 0.85.

Overall, the results confirm that syntactic complexity has a direct impact on the performance of Arabic I'rab systems, with I3rbly demonstrating superior stability and accuracy compared to plus90.

Model	Sentence Class	55%	65%	75%	85%
I3rbly	بسيطة	**90.91%**	**90.91%**	**72.73%**	**45.45%**
I3rbly	مركبة	**78.79%**	**60.61%**	**54.55%**	**36.36%**
plus90	بسيطة	54.55%	54.55%	54.55%	45.45%
plus90	مركبة	57.58%	48.48%	39.39%	**27.27%**

Table 5.21: Accuracy of Arabic I'rab platforms according to sentence complexity across different acceptance thresholds

5.7.9 Analysis by Text Source

The performance of the models was analyzed according to the source of the text (poetry, standard Arabic, and Quranic text). The results show that standard Arabic sentences are the easiest to process across all models, while poetic and Quranic texts are significantly more challenging.

Overall, all models achieved their best performance on standard Arabic texts, particularly

at the 0.55 threshold, with a gradual decline in accuracy as the evaluation threshold increased. In contrast, Quranic texts consistently produced the lowest accuracy across all settings due to their higher syntactic and rhetorical complexity.

These findings confirm that the source of the text has a direct impact on I'rab quality, where standard Arabic provides clearer and more stable structures compared to poetry and Quranic discourse.

Model	**Sentence Category**	**55%**	**65%**	**75%**	**85%**
Gemini	شعرية	64.83%	52.41%	42.76%	23.45%
Gemini	عادية	80.80%	72.35%	60.75%	43.34%
Gemini	قرآنية	63.50%	54.90%	40.06%	22.85%
Qwen	شعرية	65.52%	57.24%	53.10%	35.86%
Qwen	عادية	82.31%	75.86%	67.91%	49.21%
Qwen	قرآنية	53.41%	44.51%	34.72%	22.26%
GPT	شعرية	60.00%	50.34%	34.48%	17.93%
GPT	عادية	69.99%	63.54%	52.51%	38.68%
GPT	قرآنية	56.97%	48.66%	34.12%	19.58%
DeepSeek	شعرية	73.10%	56.55%	42.07%	31.03%
DeepSeek	عادية	78.51%	69.63%	56.81%	42.19%
DeepSeek	قرآنية	55.79%	43.03%	33.83%	22.85%
Fanar	شعرية	61.38%	49.66%	40.00%	27.59%
Fanar	عادية	73.07%	63.97%	53.51%	41.33%
Fanar	قرآنية	56.68%	45.10%	37.39%	25.82%
Jais	شعرية	68.28%	57.93%	47.59%	33.79%
Jais	عادية	79.66%	72.85%	59.96%	49.00%
Jais	قرآنية	54.90%	46.88%	37.98%	24.04%
Claude	شعرية	71.03%	57.24%	35.86%	24.14%
Claude	عادية	82.52%	69.63%	57.31%	43.48%
Claude	قرآنية	57.86%	43.03%	30.56%	20.77%
Mistral	شعرية	71.72%	59.31%	45.52%	33.10%
Mistral	عادية	80.80%	74.43%	64.76%	**51.65%**
Mistral	قرآنية	51.63%	42.43%	33.83%	22.85%

Table 5.22: Model accuracy according to text source across different acceptance thresholds

Interpretation The lower performance on Quranic and poetic texts can be explained by their richer rhetorical and syntactic structures, including inversion, ellipsis, and complex sentence embedding. Arabic poetry, in particular, often employs **poetic license**, which introduces deliberate linguistic variations such as word reordering, omission of particles, or morphological changes to preserve meter and rhyme, making automatic grammatical analysis more difficult.

Overall, these results confirm that the source of the text significantly affects I'rab accuracy. Standard Arabic texts are more structurally regular and easier for models to process, while Quranic and poetic texts introduce additional linguistic complexity that challenges syntactic parsing models.

5.7.10 Results of Arabic I'rab Platforms by Text Source

The performance of the **I3rbly** and **plus90** platforms was analyzed according to text source (poetry, standard Arabic, and Quranic text). The results show that standard Arabic sentences are the easiest to process for both systems, while poetic and Quranic texts are significantly more complex and challenging.

Overall, **I3rbly** clearly outperformed plus90, achieving very high accuracy on standard Arabic texts (reaching **100% at 0.55**), with a gradual decline as the evaluation threshold becomes stricter, while still maintaining superior performance across most settings. In contrast, **plus90** showed weaker performance, particularly on Quranic texts, which remained consistently low and dropped to **0% at the 0.85 threshold**.

These findings confirm that **text source has a direct impact on I'rab accuracy**, with standard Arabic being the easiest compared to poetic and Quranic texts.

Model	Sentence Category	55%	65%	75%	85%
I3rbly	شعرية	70.59%	47.06%	41.18%	23.53%
I3rbly	عادية	**100.00%**	**94.44%**	**83.33%**	**61.11%**
I3rbly	قرآنية	66.67%	55.56%	44.44%	22.22%
plus90	شعرية	58.82%	41.18%	41.18%	29.41%
plus90	عادية	72.22%	72.22%	55.56%	50.00%
plus90	قرآنية	22.22%	22.22%	22.22%	**0.00%**

Table 5.23: Accuracy of Arabic I'rab platforms according to text source across different acceptance thresholds

5.8 Detailed Observations on Model Behavior and Error Patterns

The experimental evaluation across different language models revealed several recurring patterns in handling Arabic syntactic analysis. Although the models differ in overall accuracy, most of them share common behaviors, such as the tendency to always produce an output even in uncertain cases. This often leads to *hallucinations*, where the model generates a linguistically plausible but syntactically incorrect analysis.

Moreover, many errors were associated with confusion between closely related grammatical functions, including:

- Confusion between predicate (الخبر) and adjective (النعت)
- Confusion between circumstantial accusative (الحال) and adjective (الصفة)
- Confusion between absolute object (المفعول المطلق) and purpose object (المفعول لأجله)
- Confusion between subject (الفاعل) and noun subject (المبتدأ)
- Confusion between apposition (البدل) and genitive construction (المضاف إليه)
- Confusion between adverb of time/place (الظرف) and circumstantial accusative (الحال)

In addition, clear differences were observed in the style of grammatical output among models. Some models produce concise I'rab outputs, while others provide highly detailed analyses that include clause-level functions and internal syntactic relations.

5.8.1 Qwen

The Qwen model demonstrated relatively strong performance compared to other models, particularly in verbal sentences. However, it exhibited recurring errors in identifying precise syntactic roles.

In the sentence:

مَتَى السَّاعَةُ؟

the model analyzed السَّاعَةُ as:

(خبر المبتدأ) subject the of predicate

while the correct analysis is:

(مبتدأ مؤخر) subject delayed

In the sentence:

الْمُعَلِّمُ الْمُخْلِصُ يَحْتَرِمُهُ طُلَّابُهُ

the model incorrectly classified الْمُعَلِّمُ as a fronted object (مفعول به مقدم), whereas it is actually a subject (مبتدأ مرفوع). It also misidentified طُلَّابُهُ as a delayed subject instead of a subject.

In poetic text, the model showed clear difficulty. In:

أَمُرُّ عَلَى الدِّيَارِ دِيَارَ لَيْلَى أُقَبِّلُ ذَا الْجِدَارَ وَذَا الْجِدَارِ

it incorrectly analyzed الجدارِ as a genitive complement (مضاف إليه), instead of a substitution structure (بدل).

It also confused temporal adverbs and circumstantial expressions. In:

مَشَيْنَا مِشْيَةَ اللَّيْثِ غَدًا وَاللَّيْثُ غَضْبَانُ

it classified غدًا as a circumstantial accusative (حال), whereas the correct interpretation is a time adverb (ظرف زمان).

Finally, Qwen often fails to explicitly handle emphatic constructions (التوكيد اللفظي) and sometimes produces partial or incomplete syntactic analyses.

5.8.2 ChatGPT

The GPT model is characterized by relatively detailed syntactic analyses, as it often provides full sentence-level I'rab, including explicit identification of syntactic positions such as:

> "the verbal sentence is in the position of a predicate (خبر المبتدأ)"

It also frequently separates verb morphology from attached pronouns, especially in conjugated forms such as:

سافروا

However, several recurring errors were observed, particularly in the treatment of adverbs and particles.

In the sentence:

جَلَسَ الْفَلَّاحُ تَحْتَ الشَّجَرَةِ يَسْتَظِلُّ بِفَيْئِهَا

the model analyzed تحت as a preposition (حرف جر), whereas the correct analysis is a locative adverb (ظرف مكان منصوب).

In:

شَتَّانَ الْجِدُّ وَالإِهْمَالُ

it classified الجدُّ as a subject (مبتدأ), while the correct syntactic role is a grammatical subject of a fixed expression (فاعل).

In:

مَشَيْنَا مَشْيَةَ اللَّيْثِ غَدًا وَاللَّيْثُ غَضْبَانُ

the model incorrectly identified غَضْبَانُ as a predicate, whereas the reference analysis considers it a noun in an independent nominal clause (مبتدأ).

Additional observed limitations include:

- Failure to consistently analyze coordination particles (واو العطف).
- Neglecting particles such as the emphatic lām in forms like لتسمعنّ.
- Omitting conjunctions such as ف at sentence beginnings.
- Confusing second subjects with genitive constructions.
- Treating explicit subjects as implicit pronouns in some cases.

5.8.3 Jais

The Jais model exhibited a relatively high frequency of errors related to ambiguity in syntactic interpretation, particularly confusion between predicate and adjective roles.

In the sentence:

الْعَالِمَانِ مُجِدَّانِ

it incorrectly analyzed مُجِدَّانِ as:

> predicate of a defective verb (خبر زال منصوب)

instead of the correct analysis:

> predicate of a nominal sentence in the nominative case (خبر مرفوع بالألف لأنه مثنى)

In:

لِتُصْبِحَ نَجْمًا

the model classified نَجْمًا as a direct object, whereas it should be analyzed as the predicate of تصبح.

Other recurring issues include:

- Occasional omission of words in sentence-level I'rab.
- Producing incomplete syntactic analyses.
- Inconsistent treatment of initial conjunctions (واو، فاء).
- Variability in output correctness when the same input is repeated.
- Frequent confusion between predicate (خبر) and adjective (نعت).

5.8.4 Mistral

The Mistral model showed relatively good performance on simple syntactic structures, but struggled with rhetorical and poetic constructions.

In the sentence:

اِسْتَحَالَتِ النَّارُ رَمَادًا

it analyzed رَمَادًا as:

circumstantial accusative (حال)

instead of the correct analysis:

predicate of (خبر استحال)

In:

نحن المسلمين موحدون

it incorrectly classified المسلمين as a predicate, whereas the correct analysis is a vocative/accusative of specification (منصوب على الاختصاص).

The model also frequently confused circumstantial accusative (الحال) with specification (التمييز). In:

الشاي ساخنًا ألذُّ منه باردًا

it analyzed باردًا as specification (تمييز), instead of circumstantial accusative (حال).

Additional observations include:

- Inconsistent analysis of conjunction particles (واو العطف), especially at sentence beginnings.
- Frequent confusion between genitive construction (المضاف إليه) and apposition (البدل).
- Difficulties with poetic structures involving inversion (التقديم والتأخير).

5.8.5 Claude

The Claude model demonstrated relatively structured I'rab output; however, it exhibited recurring errors related to predicate, adjective, and subject classification.

In:

نَامَ عُمَرُ بَاكِرًا

it analyzed بَاكِرًا as:

circumstantial accusative (حال)

instead of a temporal adverb (ظرف زمان).

In:

سَافَرَ أَخِي مُذْ طَلَعَتِ الشَّمْسُ

it classified مُذْ as a preposition (حرف جر), while the correct analysis is a temporal adverb (ظرف زمان).

Additional limitations include:

- Occasionally providing incomplete morphological classification (e.g., stating only "مبني" without specification).
- Failure to analyze emphatic repetition in constructions such as اجتهد اجتهد الطالب.

5.8.6 DeepSeek

The DeepSeek model showed noticeable variability between concise and highly detailed outputs.

In:

مَشَيْنَا مِشْيَةَ اللَّيْثِ غَدًا

it incorrectly analyzed غَدًا as a defective past verb (**فعل ماض ناقص**), instead of a temporal adverb (ظرف زمان).

In:

نفعني المعلم علمه

it incorrectly classified **علمه** as a direct object, instead of an apposition (**بدل**).

Additional issues include:

- Providing multiple alternative I'rab interpretations within a single response.
- Inconsistency between concise and detailed outputs.
- Confusion between predicate (**خبر**), adjective (**نعت**), and genitive constructions (المضاف إليه).

5.8.7 Gemini

The Gemini model demonstrated relatively balanced performance but struggled with verbal and adverbial structures.

In:

اِسْتَحَالَتِ النَّارُ رَمَادًا

it analyzed رَمَادًا as a circumstantial accusative (حال), instead of a predicate (خبر استحال).

In:

مَشَيْنَا مِشْيَةَ اللَّيْثِ غَدًا

it incorrectly analyzed غَدًا as a defective past verb, instead of a temporal adverb (ظرف زمان).

It was also observed that the model sometimes partially analyzes sentences, omitting certain words in longer structures.

Common errors include:

- Confusion between subject (مبتدأ) and predicate (خبر).
- Misclassification of adverbs as verbal elements.
- Partial sentence parsing in complex structures.

5.8.8 Fanar

The Fanar model exhibited frequent morphological and syntactic errors.

In:

تعمل الممرضات في المستشفى

it incorrectly analyzed الممرضات as a subject marked with kasra, instead of damma.

In:

خرج جميع الأطفال

it classified الأطفال as a direct object, instead of a genitive complement (مضاف إليه).

It also frequently confused predicate (خبر) with adjective (نعت), as in:

الله نعمته عظيمة

where عظيمة was incorrectly analyzed as an adjective instead of a predicate.

Additional issues include:

- Failure to analyze certain particles (e.g., الباء, الألف).
- Omission of words in sentence-level parsing.
- Insertion of external hyperlinks in Qur'anic text outputs.
- Frequent confusion between subject, predicate, and genitive structures.

5.9 Challenges and Limitations

During data collection and model evaluation, several technical and practical challenges were encountered, primarily due to variability in model behavior and platform constraints.

- **Input size limitations:** Some models restricted the number of sentences per request (e.g., GPT: 10 sentences, Fanar: 4 sentences), requiring batch processing and repeated execution.
- **Non-structured outputs:** Several models did not strictly follow the required JSON format, producing free-text outputs instead. A post-processing step using an auxiliary model was required to normalize outputs.
- **Missing or incomplete outputs:** Some models skipped words or entire sentences, especially in longer inputs, requiring manual verification.
- **Insertion of external links:** The Fanar model occasionally inserted hyperlinks in Qur'anic text outputs, requiring manual cleaning.

- **Deviation from task (explanatory outputs):** Some models shifted from syntactic analysis to explanation or interpretation, particularly on Qur'anic sentences.
- **Free-tier limitations:** Certain models (e.g., Claude) imposed daily usage limits, delaying data collection.
- **Interface instability:** Some platforms (e.g., Jais) suffered from unstable interfaces and frequent session resets.
- **Prompt forgetting:** Models such as DeepSeek, Jais, and Fanar occasionally ignored earlier instructions, requiring repeated prompt reinforcement.

Overall, these challenges demonstrate that evaluation of Arabic I'rab using LLMs depends not only on model quality but also on platform stability, output consistency, and adherence to instructions.

5.10 Conclusion

This chapter presented the experimental framework of the study, including dataset construction, preprocessing, evaluation methodology, and analysis of results.

The dataset was carefully designed to include diverse sentence types, complexities, and sources in order to evaluate model robustness under different linguistic conditions. The evaluation approach relied on similarity-based comparison rather than exact matching due to variability in syntactic expressions.

Results showed clear performance differences across models, with better performance generally observed in simple sentences compared to complex, poetic, and Qur'anic structures. Error analysis revealed recurring issues such as confusion between grammatical roles, omission of words, and inconsistent formatting.

Overall, while LLMs demonstrate promising capabilities in Arabic syntactic analysis, significant challenges remain in handling complex linguistic structures, highlighting the need for further improvements in Arabic grammatical understanding within these models.

General Conclusion

This project presents a comprehensive evaluation of Large Language Models (LLMs) and Arabic language processing systems for the task of Arabic syntactic analysis (*I'rab*), which is considered one of the most challenging linguistic tasks due to the rich morphology of Arabic, the diversity of its syntactic structures, and its strong dependence on contextual information. A dedicated dataset was developed within the *Yu'rab* (يعرب) project, proposed and supervised by Dr. Taha Zerrouki, and was used to evaluate several large language models as well as specialized Arabic parsing platforms.

To investigate the impact of acceptance thresholds on evaluation results, experiments were conducted using four similarity thresholds (55%, 65%, 75%, and 85%). The findings showed that the 55% threshold was the most appropriate, since syntactically correct answers may differ from the reference answers in wording, phrasing, or level of detail. Higher thresholds therefore tended to underestimate model performance by penalizing valid alternative formulations.

Using the 55% threshold, Claude achieved the best performance among the general-purpose language models, reaching an accuracy of 77.13% and an F1-score of 87.09%, followed by Gemini and Qwen. Jais, Mistral, and DeepSeek also produced competitive and relatively similar results, reflecting the significant progress made in Arabic syntactic analysis by modern language models.

Among the specialized Arabic parsing platforms, I3rbly achieved the best overall results, with an accuracy of 81.82% and an F1-score of 90.00%. However, it is important to note that both I3rbly and Al Maqalah (Plus90) were evaluated on a limited number of examples compared to the other systems, as only ten sentences were tested on each platform. Consequently, these results should be considered preliminary indicators of performance rather than comprehensive evaluations comparable to those conducted on the larger and more diverse benchmark dataset used for the other models.

The results also revealed that most models perform better on simple sentence structures, while their accuracy decreases on longer and more complex sentences, as well as on Qur'anic and poetic texts containing ambiguous syntactic constructions. Furthermore, the automated evaluation process exposed certain limitations, as similarity-based metrics

may penalize linguistically correct answers simply because they differ in formulation from the reference outputs.

Several practical challenges were encountered during the study, including input length limitations, occasional failure to comply with the required JSON format, difficulties in following instructions, instability of some platforms, daily usage restrictions, and the phenomenon of prompt forgetting. In addition, data collection proved to be highly time-consuming, requiring approximately three to four weeks of continuous work. Some outputs also required substantial cleaning due to the presence of irrelevant, malformed, or poorly structured content.

This work proposes a comprehensive framework for evaluating Arabic *I'rab* systems, including prompt design, output normalization, and automated evaluation procedures. Future work may extend this research by constructing larger and more diverse datasets, adopting hybrid evaluation approaches that combine automated metrics with human expert assessment, and developing specialized Arabic models specifically designed for syntactic analysis.

In conclusion, achieving human-level performance in Arabic *I'rab* remains a challenging goal that requires substantial improvements in language models, datasets, and evaluation methodologies, particularly for complex linguistic material such as Qur'anic and poetic texts.

Bibliography

[1] M. R. Rizqi, "ارتباط اللغة العربية الكلاسيكية واللغة العربية الفصحى المعاصرة وتعليمها," *Al-Fakkaar*, vol. 5, no. 2, pp. 116–128, Jul. 2024. DOI: 10.52166/alf.v5i2.7290 [Online]. Available: http://dx.doi.org/10.52166/alf.v5i2.7290

[2] u. Dr. Abdul Saboor and u. Dr. Noha Ibrahim El-Desouki Gamil, "اللغة العربية في عصر العولمة: تحديات الأدب واللغة," *Al-Qamar*, vol. 7, no. 3, pp. 1–24, Aug. 2024. DOI: 10.53762/alqamar.07.03.a01 [Online]. Available: http://dx.doi.org/10.53762/alqamar.07.03.a01

[3] A. Furoidah, "خَصَائِصُ اللُّغَةِ الْعَرَبِيَّةِ الْفُصْحَى وَمَكَانَتُهَا فِي الدِّيْنِ الْإِسْلَامِي," *Al-Fusha : Arabic Language Education Journal*, vol. 1, no. 2, pp. 45–53, Sep. 2020. DOI: 10.36835/alfusha.v1i2.348 [Online]. Available: http://dx.doi.org/10.36835/alfusha.v1i2.348

[4] u. Mohamad Al-Taher Radwan Mohamed Ismail, "Arabic language grammar," *Mesopotamian Journal of Arabic Language Studies*, no. 2024, pp. 22–34, Mar. 2024. DOI: 10.58496/mjals/2024/003 [Online]. Available: http://dx.doi.org/10.58496/mjals/2024/003

[5] N. G. Mingazova, R. A. Al-Foadi, and V. G. Subich, "The arabic root and the peculiarities of its language categorization (structure and inflection)," *International Journal of Criminology and Sociology*, vol. 9, pp. 2628–2637, Apr. 2022. DOI: 10.6000/1929-4409.2020.09.324 [Online]. Available: http://dx.doi.org/10.6000/1929-4409.2020.09.324

[6] M. El-Defrawy, Y. El-Sonbaty, and N. A. Belal, "A rule-based subject-correlated arabic stemmer," en, *Arabian Journal for Science and Engineering*, vol. 41, no. 8,

pp. 2883–2891, Feb. 2016. DOI: 10.1007/s13369-016-2029-2 [Online]. Available: http://dx.doi.org/10.1007/s13369-016-2029-2

[7] K. SHAALAN, M. MAGDY, and A. FAHMY, "Analysis and feedback of erroneous arabic verbs," en, *Natural Language Engineering*, vol. 21, no. 2, pp. 271–323, Sep. 2013. DOI: 10.1017/s1351324913000223 [Online]. Available: http://dx.doi.org/10.1017/s1351324913000223

[8] M. al-Mukhtār Lammāt, "Ahamiyyat al-iᶜrāb fī al-tafsīr," *al-Majalla al-Maghribiyya li-Nashr al-Abḥāth al-ᶜIlmiyya*, no. 13, 2026.

[9] M. OUDAH and K. SHAALAN, "Nera 2.0: Improving coverage and performance of rule-based named entity recognition for arabic," en, *Natural Language Engineering*, vol. 23, no. 3, pp. 441–472, May 2016. DOI: 10.1017/s1351324916000097 [Online]. Available: http://dx.doi.org/10.1017/s1351324916000097

[10] S. Skaria, P. K. A. Krishna, K. Anagha, S. K. James, and V. Nargeese, "Natural language processing (nlp): Advancements, applications, and challenges," *AIP Conference Proceedings*, vol. 3240, p. 20 005, Jan. 2024. DOI: 10.1063/5.0230441 [Online]. Available: http://dx.doi.org/10.1063/5.0230441

[11] M. A. A. S. Ali, "Ai-natural language processing (nlp)," *International Journal for Research in Applied Science and Engineering Technology*, vol. 9, pp. 135–140, Aug. 2021. DOI: 10.22214/ijraset.2021.37293 [Online]. Available: http://dx.doi.org/10.22214/ijraset.2021.37293

[12] A. M. Alayba, "Arabic natural language processing (nlp): A comprehensive review of challenges, techniques, and emerging trends," en, *Computers*, vol. 14, no. 11, p. 497, Nov. 2025. DOI: 10.3390/computers14110497 [Online]. Available: http://dx.doi.org/10.3390/computers14110497

[13] Y. A. Moaiad, M. Alobed, M. Alsakhnini, and A. M. Momani, "Challenges in natural arabic language processing," *Edelweiss Applied Science and Technology*, vol. 8, no. 6, pp. 4700–4705, Nov. 2024. DOI: 10.55214/25768484.v8i6.3018 [Online]. Available: http://dx.doi.org/10.55214/25768484.v8i6.3018

[14] M. Deepadharshana and S. Vijayarani, "Large language models: State-of-the-art techniques, applications and emerging trends," en, *International Journal of Artificial Intelligence and Soft Computing*, vol. 9, no. 2, pp. 85–105, Jan. 2025. DOI: 10.1504/

ijaisc.2025.149613 [Online]. Available: http://dx.doi.org/10.1504/IJAISC.2025.149613

[15] J. Cheng and H. Amiri, "Linguistic blind spots of large language models," *arXiv*, Jan. 2025. DOI: 10.48550/ARXIV.2503.19260 [Online]. Available: https://arxiv.org/abs/2503.19260

[16] M. Marmonier, R. Bawden, and B. Sagot, "Explicit learning and the llm in machine translation," Jan. 2025. DOI: 10.48550/ARXIV.2503.09454 [Online]. Available: https://arxiv.org/abs/2503.09454

[17] M. Klemen, T. Arčon, L. Terčon, M. Robnik-Šikonja, and K. Dobrovoljc, "Towards corpus-grounded agentic llms for multilingual grammatical analysis," Jan. 2025. DOI: 10.48550/ARXIV.2512.00214 [Online]. Available: https://arxiv.org/abs/2512.00214

[18] M. Alhilal, "Understanding syntax in large language models: Successes and limitations," *International Journal of Educational Sciences and Arts*, vol. 4, no. 1, pp. 10–28, Jan. 2025. DOI: 10.59992/ijesa.2025.v4n1p1 [Online]. Available: http://dx.doi.org/10.59992/ijesa.2025.v4n1p1

[19] F. Team et al., "Fanar: An arabic-centric multimodal generative ai platform," *arXiv preprint arXiv:2501.13944*, 2025.

[20] Technology Innovation Institute, *Falcon llm*, https://falconllm.tii.ae/, Accessed February 17, 2026., n.d.

[21] Almaqalah, *Almaqalah*, https://www.almaqalah.com/ara-aipars, Accessed February 17, 2026., n.d.

[22] I3rbly, *I^{c}rab li platform*, https://www.i3rbly.com/tool, Accessed February 17., n.d.

[23] Ajroum, *Ajroum arabic grammar platform*, https://ajroum.vercel.app/, Accessed February 17, 2026., n.d.

[24] S. Al-Ghamdi, H. Al-Khalifa, and A. Al-Salman, "Fine-tuning bert-based pre-trained models for arabic dependency parsing," *Applied Sciences*, vol. 13, no. 7, p. 4225, 2023.

[25] A. R. Maulidiya, M. Abdurrahman, and N. Saleh, "Exploring ai capabilities in arabic grammar: Comparative analysis of chatgpt and gemini.," *Arabiyat: Journal of Arabic Education & Arabic Studies/Jurnal Pendidikan Bahasa Arab Dan Kebahasaaraban*, vol. 11, no. 2, 2024.

[26] المكتب العلمي للتأليف .ملخص قواعد اللغة العربية و الصرف: مرجع كامل لقواعد النحو والصرف والترجمة, 1974. [Online]. Available: https://books.google.dz/books?id=JP5zMgAACAAJ

[27] R. N. Almatham et al., "Balsam: A platform for benchmarking arabic large language models," in *Proceedings of The Third Arabic Natural Language Processing Conference*, 2025, pp. 258–277.

[28] Z. Alyafeai, M. S. Alshaibani, B. AlKhamissi, H. Luqman, E. Alareqi, and A. Fadel, "Taqyim: Evaluating arabic nlp tasks using chatgpt models," *arXiv preprint arXiv:2306.16322*, 2023.

[29] R. Kharsa, A. Elnagar, and S. Yagi, "Bert-based arabic diacritization: A state-of-the-art approach for improving text accuracy and pronunciation," *Expert Systems with Applications*, vol. 248, p. 123 416, 2024, ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2024.123416 [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417424002811

[30] M. T. I. Khondaker, A. Waheed, M. Abdul-Mageed, et al., "Gptaraeval: A comprehensive evaluation of chatgpt on arabic nlp," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023, pp. 220–247.

[31] W. Antoun, F. Baly, and H. Hajj, "Arabert: Transformer-based model for arabic language understanding," in *Proceedings of the 4th workshop on open-source arabic corpora and processing tools, with a shared task on offensive language detection*, 2020, pp. 9–15.

[32] O. Saadiyeh, A. Ramadan, M. Hajjar, and G. Bernard, "A comparative study of arabic syntactic analyzers," *Frontiers in Artificial Intelligence*, vol. 8, p. 1 638 743, 2025.

[33] S. Green and C. D. Manning, "Better arabic parsing: Baselines, evaluations, and analysis," in *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, 2010, pp. 394–402.

[34] I. Ājurrūm, *Al-Ājurrūmiyyah*. Dār al-Mashāriᶜ, 1853.

[35] D. Halabi, E. Fayyoumi, and A. Awajan, "I3rab: A new arabic dependency treebank based on arabic grammatical theory," *Transactions on Asian and Low-Resource Language Information Processing*, vol. 21, no. 2, pp. 1–32, 2021.

[36] L. G. B. Johnsen, "Grammaticality judgments in humans and language models: Revisiting generative grammar with llms," Jan. 2025. DOI: 10.48550/ARXIV.2512.10453 [Online]. Available: https://arxiv.org/abs/2512.10453

[37] M. Oba et al., "Can language models induce grammatical knowledge from indirect evidence?," Jan. 2024. DOI: 10.48550/ARXIV.2410.06022 [Online]. Available: https://arxiv.org/abs/2410.06022

[38] X. Bai et al., "Constituency parsing using llms," Jan. 2023. DOI: 10.48550/ARXIV.2310.19462 [Online]. Available: https://arxiv.org/abs/2310.19462

[39] OpenAI, *Chatgpt*, https://openai.com/chatgpt, Accessed February 17, 2026., n.d.

[40] Anthropic, *Claude*, https://www.anthropic.com/claude, Accessed February 17, 2026., n.d.

[41] Google DeepMind, *Gemini: A family of highly capable multimodal models*, https://deepmind.google/technologies/gemini/, Accessed February 17, 2026., n.d.

[42] Alibaba Cloud, *Qwen large language model*, https://qwenlm.ai/, Accessed February 17, 2026., n.d.

[43] DeepSeek, *Deepseek ai models*, https://www.deepseek.com/, Accessed February 17, 2026., n.d.

[44] Mistral AI, *Mistral ai models*, https://mistral.ai/, Accessed February 17, 2026., n.d.

[45] Qatar Computing Research Institute, *Fanar arabic language model*, https://www.qcri.org/, Accessed February 17, 2026., n.d.

[46] Araby.ai, *Araby.ai platform*, https://araby.ai/, Accessed February 17, 2026., n.d.

[47] G42 and Mohamed bin Zayed University of Artificial Intelligence, *Jais large language model*, https://www.g42.ai/, Accessed February 17, 2026., n.d.

[48] M. Nasrulloh, "Sejarah kronologi bahasa arab: Semitik," *TARBAWI:Journal on Islamic Education*, vol. 1, no. 1, pp. 90–100, Apr. 2023. DOI: 10.24269/tarbawi.v1i1.2977 [Online]. Available: http://dx.doi.org/10.24269/tarbawi.v1i1.2977

[49] N. Nurbaiti, "The contribution of al-'ilm sharaf to the development of understanding classical arabic grammar at islamic educational institutions," *Jurnal Al-Fikrah*, vol. 13, no. 1, pp. 112–121, Jun. 2024. DOI: 10.54621/jiaf.v13i1.876 [Online]. Available: http://dx.doi.org/10.54621/jiaf.v13i1.876

[50] S. Al-Ghamdi, H. Al-Khalifa, and A. Al-Salman, "Fine-tuning bert-based pre-trained models for arabic dependency parsing," en, *Applied Sciences*, vol. 13, no. 7, p. 4225, Mar. 2023. DOI: 10.3390/app13074225 [Online]. Available: http://dx.doi.org/10.3390/app13074225

[51] H. Alshammari, A. El-Sayed, and K. Elleithy, "Ai-generated text detector for arabic language using encoder-based transformer architecture," en, *Big Data and Cognitive Computing*, vol. 8, no. 3, p. 32, Mar. 2024. DOI: 10.3390/bdcc8030032 [Online]. Available: http://dx.doi.org/10.3390/bdcc8030032

Annex

.1 Annex A: I‘rab Extraction and Structured Result Generation

The generated I‘rab outputs were extracted, standardized, and converted into structured tables to enable automatic comparison with the reference annotations.

Claude Output Example


```
{
",يَحْكُمُ الْقَاضِي بِالْعَدْلِ" "sentence":
[ "i3rab":
{
",يَحْكُمُ" "token":
"فعل مضارع مرفوع وعلامة رفعه الضمة الظاهرة" "i3rab":
},
{
",الْقَاضِي" "token":
"فاعل مرفوع وعلامة رفعه الضمة المقدرة على الياء للثقل" "i3rab":
},
{
",بِالْعَدْلِ" "token":
"الباء حرف جر مبني على الكسر لا محل له من الإعراب، والعدل اسم مجرور بالباء" "i3rab":
وعلامة جره الكسرة، *الجار والمجرور متعلقان بالفعل يحكم"
}
]
}
```


After the initial transformation, the data becomes organized as follows:

Sentence	**Token**	**Label**
يَحْكُمُ الْقَاضِي بِالْعَدْلِ	يَحْكُمُ	فعل مضارع مرفوع وعلامة رفعه الضمة الظاهرة
يَحْكُمُ الْقَاضِي بِالْعَدْلِ	الْقَاضِي	فاعل مرفوع وعلامة رفعه الضمة المقدرة على الياء للثقل
يَحْكُمُ الْقَاضِي بِالْعَدْلِ	بِالْعَدْلِ	الباء حرف جر مبني على الكسر لا محل له من الإعراب، والعدل اسم مجرور بالباء وعلامة جره الكسرة، *الجار والمجرور متعلقان بالفعل يحكم

Table 24: Claude structured I'rab data after JSON transformation

Gemini Output Example


```
{
  "sentence": "يَحْكُمُ الْقَاضِي بِالْعَدْلِ",
  "i3rab": [
    {
      "token": "يَحْكُمُ",
      "i3rab": "فعل مضارع مرفوع وعلامة رفعه الضمة الظاهرة"
    },
    {
      "token": "الْقَاضِي",
      "i3rab": "فاعل مرفوع وعلامة رفعه الضمة المقدرة على الياء للثقل"
    },
    {
      "token": "بِ",
      "i3rab": "حرف جر مبني على الكسر لا محل له من الإعراب"
    },
    {
      "token": "الْعَدْلِ",
      "i3rab": "اسم مجرور بالباء وعلامة جره الكسرة الظاهرة"
    }
  ]
}
```


After the initial transformation, the data becomes organized as follows:

Sentence	**Token**	**Label**
يَحْكُمُ الْقَاضِي بِالْعَدْلِ	يَحْكُمُ	فعل مضارع مرفوع وعلامة رفعه الضمة الظاهرة
يَحْكُمُ الْقَاضِي بِالْعَدْلِ	الْقَاضِي	فاعل مرفوع وعلامة رفعه الضمة المقدرة على الياء للثقل
يَحْكُمُ الْقَاضِي بِالْعَدْلِ	بِ	حرف جر مبني على الكسر لا محل له من الإعراب
يَحْكُمُ الْقَاضِي بِالْعَدْلِ	الْعَدْلِ	اسم مجرور بالباء وعلامة جره الكسرة الظاهرة

Table 25: Gemini structured I'rab data after JSON transformation

Qwen Output Example

```
{
  "sentence": "يَحْكُمُ الْقَاضِي بِالْعَدْلِ",
  "i3rab": [
    {
      "token": "يَحْكُمُ",
      "i3rab": "فعل مضارع مرفوع وعلامة رفعه الضمة الظاهرة"
    },
    {
      "token": "الْقَاضِي",
      "i3rab": "فاعل مرفوع وعلامة رفعه الضمة المقدرة على الياء للثقل"
    },
    {
      "token": "بِالْعَدْلِ",
      "i3rab": "الباء حرف جر مبني على الكسر لا محل له من الإعراب، وَالْعَدْلِ اسم مجرور بالباء
وعلامة جره الكسرة الظاهرة، *الجار والمجرور في محل نصب متعلق بالفعل"
    }
  ]
}
```

After the initial transformation, the data becomes organized as follows:

Sentence	**Token**	**Label**
يَحْكُمُ الْقَاضِي بِالْعَدْلِ	يَحْكُمُ	فعل مضارع مرفوع وعلامة رفعه الضمة الظاهرة
يَحْكُمُ الْقَاضِي بِالْعَدْلِ	الْقَاضِي	فاعل مرفوع وعلامة رفعه الضمة المقدرة على الياء للثقل
يَحْكُمُ الْقَاضِي بِالْعَدْلِ	بِالْعَدْلِ	الباء حرف جر مبني على الكسر لا محل له من الإعراب، وَالْعَدْلِ اسم مجرور بالباء وعلامة جره الكسرة الظاهرة، *الجار والمجرور في محل نصب متعلق بالفعل

Table 26: Qwen structured I'rab data after JSON transformation

Mistral Output Example

```
{
  "sentence": "يحكم القاضي بالعدل",
  "i3rab": [
    {
      "token": "يحكم",
      "i3rab": "فعل مضارع مرفوع وعلامة رفعه الضمة"
    },
    {
      "token": "القاضي",
      "i3rab": "فاعل مرفوع وعلامة رفعه الضمة المقدرة على الياء للتعذر"
    },
    {
      "token": "بالعدل",
      "i3rab": "الباء حرف جر مبني على الكسر لا محل له من الإعراب، والعدل اسم مجرور بالباء
وعلامة جره الكسرة، *الجار والمجرور في محل نصب حال"
    }
  ]
}
```

After the initial transformation, the data becomes organized as follows:

Sentence	Token	Label
يَحْكُمُ الْقَاضِي بِالْعَدْلِ	يحكم	فعل مضارع مرفوع وعلامة رفعه الضمة
يَحْكُمُ الْقَاضِي بِالْعَدْلِ	القاضي	فاعل مرفوع وعلامة رفعه الضمة المقدرة على الياء للتعذر
يَحْكُمُ الْقَاضِي بِالْعَدْلِ	بالعدل	الباء حرف جر مبني على الكسر لا محل له من الإعراب، والعدل اسم مجرور بالباء وعلامة جره الكسرة، *الجار والمجرور في محل نصب حال

Table 27: Mistral structured I'rab data after JSON transformation

DeepSeek Output Example

```
{
  "sentence": "يَحْكُمُ الْقَاضِي بِالْعَدْلِ",
  "i3rab": [
    {
      "token": "يَحْكُمُ",
      "i3rab": "فعل مضارع مرفوع وعلامة رفعه الضمة الظاهرة"
    },
    {
      "token": "الْقَاضِي",
      "i3rab": "فاعل مرفوع وعلامة رفعه الضمة المقدرة على الياء للثقل"
    },
    {
      "token": "بِالْعَدْلِ",
      "i3rab": "الباء حرف جر مبني على الكسر، العدل اسم مجرور بالباء وعلامة جره الكسرة،
*الجار والمجرور في محل نصب مفعول به (أو متعلق بالفعل)"
    }
  ]
}
```

After the initial transformation, the data becomes organized as follows:

Sentence	**Token**	**Label**
يَحْكُمُ الْقَاضِي بِالْعَدْلِ	يَحْكُمُ	فعل مضارع مرفوع وعلامة رفعه الضمة الظاهرة
يَحْكُمُ الْقَاضِي بِالْعَدْلِ	الْقَاضِي	فاعل مرفوع وعلامة رفعه الضمة المقدرة على الياء للثقل
يَحْكُمُ الْقَاضِي بِالْعَدْلِ	بِالْعَدْلِ	الباء حرف جر مبني على الكسر، العدل اسم مجرور بالباء وعلامة جره الكسرة، *الجار والمجرور في محل نصب مفعول به (أو متعلق بالفعل)

Table 28: DeepSeek structured I'rab data after JSON transformation

Fanar Output Example

```
{
  "sentence": "يحكم القاضي بالعدل",
  "i3rab": [
    {
      "token": "يحكم",
      "i3rab": "فعل مضارع مرفوع وعلامة رفعه الضمة الظاهرة"
    },
    {
      "token": "القاضي",
      "i3rab": "فاعل مرفوع وعلامة رفعه الضمة الظاهرة"
    },
    {
      "token": "بالعدل",
      "i3rab": "الباء حرف جر مبني على الكسر لا محل له من الإعراب، والعدل اسم مجرور بالباء
وعلامة جره الكسرة"
    }
  ]
}
```

After the initial transformation, the data becomes organized as follows:

Sentence	**Token**	**Label**
يَحْكُمُ الْقَاضِي بِالْعَدْلِ	يحكم	فعل مضارع مرفوع وعلامة رفعه الضمة الظاهرة
يَحْكُمُ الْقَاضِي بِالْعَدْلِ	القاضي	فاعل مرفوع وعلامة رفعه الضمة الظاهرة
يَحْكُمُ الْقَاضِي بِالْعَدْلِ	بالعدل	الباء حرف جر مبني على الكسر لا محل له من الإعراب، والعدل اسم مجرور بالباء وعلامة جره الكسرة

Table 29: Fanar structured I'rab data after JSON transformation

GPT Output Example


```
{
  "sentence": "يَحْكُمُ الْقَاضِي بِالْعَدْلِ",
  "i3rab": [
    {
      "token": "يَحْكُمُ",
      "i3rab": "فعل مضارع مرفوع وعلامة رفعه الضمة، والفاعل ضمير مستتر تقديره هو"
    },
    {
      "token": "الْقَاضِي",
      "i3rab": "فاعل مرفوع وعلامة رفعه الضمة الظاهرة"
    },
    {
      "token": "بِالْعَدْلِ",
      "i3rab": "الباء حرف جر مبني على الكسر لا محل له من الإعراب، والعدل اسم مجرور بالباء
وعلامة جره الكسرة"
    }
  ]
}
```


After the initial transformation, the data becomes organized as follows:

Sentence	**Token**	**Label**
يَحْكُمُ الْقَاضِي بِالْعَدْلِ	يَحْكُمُ	فعل مضارع مرفوع وعلامة رفعه الضمة، والفاعل ضمير مستتر تقديره هو
يَحْكُمُ الْقَاضِي بِالْعَدْلِ	الْقَاضِي	فاعل مرفوع وعلامة رفعه الضمة الظاهرة
يَحْكُمُ الْقَاضِي بِالْعَدْلِ	بِالْعَدْلِ	الباء حرف جر مبني على الكسر لا محل له من الإعراب، والعدل اسم مجرور بالباء وعلامة جره الكسرة

Table 30: GPT structured I'rab data after JSON transformation

Jais Output Example

```
{
  "sentence": "يَحْكُمُ الْقَاضِي بِالْعَدْلِ",
  "i3rab": [
    {
      "token": "يَحْكُمُ",
      "i3rab": "فعل مضارع مرفوع بالضمة."
    },
    {
      "token": "الْقَاضِي",
      "i3rab": "فاعل مرفوع بالضمة المقدرة."
    },
    {
      "token": "بِالْعَدْلِ",
      "i3rab": "الباء حرف جر، والعدل اسم مجرور بالكسرة."
    }
  ]
}
```

After the initial transformation, the data becomes organized as follows:

Sentence	**Token**	**Label**
يَحْكُمُ الْقَاضِي بِالْعَدْلِ	يَحْكُمُ	فعل مضارع مرفوع بالضمة.
يَحْكُمُ الْقَاضِي بِالْعَدْلِ	الْقَاضِي	فاعل مرفوع بالضمة المقدرة.
يَحْكُمُ الْقَاضِي بِالْعَدْلِ	بِالْعَدْلِ	الباء حرف جر، والعدل اسم مجرور بالكسرة.

Table 31: Jais structured I'rab data after JSON transformation