



Palestine Technical University (Khadooreh)

Faculty of Engineering and Technology

Department of Computer Systems Engineering

Detecting Broken Plural Words Using AI Model



Prepared By:

Haneen Khalil 202112505

Raneen Jubahi 202010062

Supervised by:

Dr. Osama Hamed

Dr. Anas Melhem

**A graduation project submitted in partial fulfilment of the requirements for the
bachelor's degree in computer systems engineering.**

Tulkarm, Palestine

June 2022

الشكر والاهداء

في رحلة هذا المشروع، لم تكن المعرفة وحدها كافية، بل كانت هناك قلوبٌ دعمت، وعقولٌ وجَّهت، وأيادٍ امتدَّت بخيرٍ وصبرٍ. لذلك، من الواجب أن أكتب هذه الكلمات بامتنانٍ عميقٍ لكلِّ مَنْ كان جزءاً من هذا الإنجاز.

إلى مَنْ كانت أوَّل مَنْ علَّمني كيف أنطقُ الكلمة وأحبُّ الحرف، إلى أُمِّي الغالية، مُعلِّمة اللغة العربية، التي علَّمتني أن للحرف روحاً، وأن اللغة العربية ليست مجردَ مادةٍ دراسية، بل هُويَّةٌ تحكى. شكراً لكِ على حبِّك الذي شكَّل أولَ جذورِ هذا الشغف، وعلى دعمكِ الهادئ الذي رافقتني في كلِّ مراحلِ هذا المشروع.

وإلى مَنْ وجَّهوني علمياً وكانوا سنداً حقيقياً في رحلتي، الدكتور أسامة حامد، والدكتور أنس ملحم، أشكرُكما إشرافكما الكريم، وملاحظاتكما الدقيقة، وحرصكما الدائم على أن يكونَ هذا العملُ علمياً متيناً ومفيداً. شكراً لتقنيتكما، ولأنكما كنتم جزءاً من هذا الطريق.

كما لا أنسى عائلتي وأصدقائي، الذين كانوا لي مصدرَ راحةٍ ودَفعةٍ أملٍ في كلِّ لحظةٍ تعب، ومن آمنوا معي أن التكنولوجيا لا تُضعف لغتنا... بل يمكن أن تحفظها.

هذا المشروع هو امتدادٌ لحُبِّ قديمٍ للغة العربية، ودليلٌ صغيرٌ على أن الذكاء الاصطناعي يمكنه أن يُصغي للعربية، وخطوةٌ صغيرةٌ في سبيل أن تكون هذه اللغة العظيمة حاضرةً بقوةٍ في مستقبل التقنية.

"وما ازداد المرءُ علماً إلا ازدادَ للغة عشقاً"

ABSTRACT

Arabic broken plurals pose one of the most complex challenges in Natural Language Processing (NLP), due to their irregular, non-concatenative morphological structure and lack of fixed pluralization rules. Unlike sound plurals, which follow predictable suffix-based patterns, broken plurals require deep morphological understanding, making them difficult to analyze using conventional AI methods.

This project presents a comprehensive AI-powered system for the detection and classification of Arabic plural forms, with a focus on broken plurals. A manually curated dataset of 1,701 entries was developed, including broken, sound masculine, and sound feminine plurals, enriched with lemma forms, diacritics, morphological patterns, and English translations.

The word-level classification model was built using character-level TF-IDF features and multiple machine learning algorithms, with Random Forest achieving the best F1-score (0.9976). Additionally, a sentence-level classifier was implemented using Support Vector Machine (SVM), integrated with Stanza for part-of-speech tagging to extract plural candidates, which were then passed to the word-level model for final confirmation.

The system was deployed through two interactive web interfaces: one using React and Flask for word-level analysis, and another using Streamlit for sentence-level analysis. Both interfaces provide real-time predictions with confidence scores and user-friendly designs tailored for educators, researchers, and developers.

This project bridges classical Arabic morphology with modern AI techniques, laying the groundwork for future NLP applications that incorporate deep learning, contextual analysis, and cross-linguistic Arabic support in intelligent systems.

ملخص

يُعدّ جمع التكسير من أكثر الظواهر الصرفية تعقيداً في اللغة العربية، وهو يشكل تحدياً حقيقياً أمام تقنيات معالجة اللغة الطبيعية (NLP) نظراً لافتقاره إلى قواعد اشتقاقية ثابتة واعتماده على أنماط صرفية غير منتظمة. وعلى خلاف الجموع السالمة التي تُبنى وفق صيغ مألوفة، يتطلب التعامل مع جمع التكسير فهماً دقيقاً للبنية الصرفية العميقة للكلمة، وهو ما يصعب على النماذج التقليدية تحقيقه بكفاءة.

استجابةً لهذا التحدي، طوّر هذا المشروع نظاماً ذكياً ومتكاملاً لتصنيف الجموع العربية، مع تركيز خاص على جمع التكسير. شُيّدت مجموعة بيانات شاملة ومُشروحة يدوياً ضمت 1701 مدخلاً، موزعة بين جموع التكسير، وجموع المذكر والمؤنث السالم، مدعّمة بمعلومات لغوية دقيقة كالجذر، والوزن الصرفي، والتشكيل، والترجمة الإنجليزية.

استُخدم تمثيل TF-IDF على مستوى الحروف لاستخلاص السمات اللغوية، وجرى تدريب أربعة نماذج تعلم آلي، كان أفضلها نموذج الغابة العشوائية (Random Forest) الذي حقق F1-score بلغ 0.9976 على مستوى الكلمات. ولاحقاً، تم تطوير نموذج إضافي على مستوى الجملة باستخدام خوارزمية SVM، مدعوماً بمحلل صرفي (Stanza) لتحديد الكلمات المحتملة ثم إعادة تصنيفها بالنموذج الأول.

تم دمج النموذجين عبر واجهتين تفاعليتين: الأولى تعتمد على React و Flask لمعالجة الكلمات، والثانية مبنية باستخدام Streamlit لمعالجة الجمل، ما أتاح تقديم نتائج فورية ودقيقة مدعومة بدرجات ثقة قابلة للتفسير.

يجمع هذا المشروع بين العمق اللغوي والدقة التقنية، ويُعدّ نقلة نوعية في مجال تحليل الجموع في اللغة العربية آلياً. كما يوفر أداة تعليمية وبحثية يمكن الاستفادة منها في مجالات تعليم اللغة، التصحيح اللغوي، والترجمة، ويشكّل أساساً راسخاً لتطوير تطبيقات مستقبلية تعتمد على التعلم العميق وتحليل السياق، بما يساهم في تعزيز حضور اللغة العربية في بيئات الذكاء الاصطناعي متعددة اللغات.

TABLE OF CONTENTS

Contents

الشكر والاهداء.....	I
ABSTRACT	II
ملخص.....	III
TABLE OF CONTENTS	1
LIST OF FIGURES	4
LIST OF TABLE.....	5
CHAPTER 1: INTRODUCTION.....	6
1.1 Background and Problem Overview:	6
1.2 Problem Statement.....	7
1.3 Objectives	7
1.4 Motivation	8
1.5 Proposed Solution	10
1. Building a Specialized Dataset:	11
2. Model Development and Selection:..	11
3. Future Directions:.....	11
1.6 Thesis structures	11
CHAPTER 2: RELATED WORK.....	12
2.1 Arabic NLP Challenges.....	13
2.2 Existing Models and Techniques	13
2.3 Research Gaps	14
2.4 Contribution of This Study	15
CHAPTER 3: DATASET BUILDING AND DEVELOPMENT	16
3.1 Dataset construction:.....	16
3.1.1: Data Collection	16
3.1.2 Word Annotation	18
3.1.3 Data Documentation	19
3.1.4: Exploring New Patterns	19
3.1.5: Analysis and Comparison	21
3.2 Data Processing: Cleaning, Organizing, and Preparing for Modeling	22

3.2.1 Preliminary Data Analysis	22
3.2.2 Data Cleaning and Normalization	23
3.2.3 Manual Lemmatization of Target Words	23
3.2.4 Morphological Weight Assignment.....	24
3.2.5 Plural Type Classification	24
3.2.6 Database Organization	25
3.3 Deriving Additional Linguistic Features	26
3.3.1 English Translation	27
3.3.2 Latin Transliteration	27
3.4 Expanding the Dataset with New Plural Types.....	29
3.4.1 Inclusion of Sound Masculine Plural.....	29
3.4.2 Inclusion of Sound Feminine Plural	31
3.5 Significance of the Additions.....	33
3.6 Relative Frequency of Morphological Weights in Broken Plurals	34
CHAPTER 4: SYSTEM ARCHITECTURE AND IMPLEMENTATION	39
4.1 Tools and Technologies Used	39
4.2 Data Preparation Pipeline.....	40
4.2.1 Data Collection	40
4.2.2 Data Cleaning and Normalization	40
4.2.3 TF-IDF Feature Extraction.....	40
4.2.4 Train-Test Split	41
4.3 Model Training and Evaluation	41
4.3.1 Models Implemented	41
4.3.2 Rationale for Selection	41
4.3.3 Hyperparameter Tuning.....	41
4.4 Model Serialization and Integration	41
4.4.1 Joblib Serialization	41
4.5 System Architecture	42
4.5.1 Use Case Overview.....	42
4.5.2 Architecture Flow.....	42
4.6 Conclusion	44
CHAPTER 5: RESULTS AND EVALUATION.....	45
5.1 Statistical Performance Metrics	45
5.2 K-Fold Cross Validation	45
5.3 Evaluation on Unseen Data.....	46
5.4 Error Analysis.....	46

5.5 Visual Comparisons.....	47
5.6 Impact of Preprocessing Enhancements	47
5.7 Summary	47
CHAPTER 6: DEPLOYMENT AND INTERFACE	48
6.1 Front-End Interface (React.js)	48
6.2 Back-End Ann and API Integration	49
6.3 Deployment Process	49
6.4 GitHub Repository Structure.....	50
6.5 Conclusion	50
CHAPTER 7: Sentence-Level Classification System for Arabic Broken Plurals	51
7.1 Introduction.....	51
7.2 Tools and Technologies Used	51
7.3 Data Preparation Pipeline.....	52
7.4 Model Training and Evaluation	53
7.4.1 K-Fold Validation.....	54
7.4.2 Serialization.....	54
7.5 System Architecture and Use Case Flow	55
7.5.1 Use Case	55
7.5.2 Data Flow	55
7.6 Deployment and Interface	55
CHAPTER 8: CONCLUSION AND FUTURE WORK.....	58
8.1 Conclusion	58
8.2 Future Work	59
8.3 Final Remarks	60

LIST OF FIGURES

Figure	Page
Figure 3.1	Sample of from the final structured dataset. 25
Figure 3.2	Sample visualization of annotated Sound Masculine Plural entries. 31
Figure 3.3	Sample visualization of annotated Sound Feminine Plural entries 33
Figure 3.4	Relative frequency distribution of broken plural patterns in (تسعة عشر)the novel 35
Figure 3.5	Sample of manually extracted and labeled broken plurals from the novel 36
Figure 3.6	Frequency distribution of broken plural morphological weights in our dataset 37
Figure 4. 1	TF-IDF Vectorizer Configuration for Character-Level N-grams 40
Figure 4.2	Use Case Diagram showing the interaction between user and system. 43
Figure 4.3	Flowchart of the complete system pipeline (from user input to prediction). 44
Figure 6.1	User Interface of the Arabic Plural Classification System Displaying Real-Time Results 48
Figure6.2	Directory Structure of the Plural Classification System 50
Figure 7.1	A line chart showing consistent accuracy values across all 5 folds of cross-validation. 54
Figure 7.2	A diagram illustrating how a sentence is processed from input to classification using SVM, Stanza, and Random Forest. 55
Figure7.3	Activity Diagram of Broken Plural Detection System 54

LIST OF TABLE

Table		Page
	showcases common examples of Arabic broken plurals, demonstrating the non-linear transformations between singular and plural forms based on morphological patterns.	24
Table 3.1		
Table 3.2	summarizes the core structure of the dataset	26
Table 3.3	Sample English translations for broken plural lemmas used in semantic enrichment and multilingual tasks.	27
Table 3.4	Sample of Latin transliteration for broken plural lemmas.	28
	Final Result represents the full structured dataset used for	29
Table 3.5	model training.	
	examples of the sound masculine plural entries added to the	31
Table 3.6	dataset.	
Table 3.7	Examples of Sound Feminine Plural	32
Table 3.8	Comparison of Morphological Weight Frequencies between Dataset and Novel	38
Table 4.1	Summary of Tools and Libraries Used in the Project	39
Table 5.1	Comparison of the performance of machine learning models based on accuracy, precision, recall, and F1-score.	45
	F1-Score Results Across 5-Fold Cross Validation	46
Table 5.2	Demonstrating Model Stability	
Table 7.1	Tools and Libraries Used in Sentence-Level Classification	52
Table 7.2	Example of sentence cleaning step	53
Table 7.3:	Comparison of Classifiers on Sentence- Level Data	54

CHAPTER 1: INTRODUCTION

1.1 Background and Problem Overview:

Arabic is one of the most prominent languages in the world, spoken in over 22 countries[26]. Despite its importance, the representation of Arabic in the field of artificial intelligence (AI) remains limited, particularly in natural language processing (NLP) applications. One of the key challenges facing AI in handling Arabic is detecting broken plurals, which involve significant and irregular structural changes when singular words are converted into their plural forms. For instance, the word “كِتَاب” (kitab, meaning "book") transforms into “كُتُب” (kutub, meaning "books"). These patterns deviate from regular rules, making them particularly difficult for computational models to analyze accurately [27].

AI applications struggle to process Arabic texts effectively due to the lack of comprehensive datasets that encompass the diverse patterns of broken plurals [28]. This deficiency hinders the performance of models in tasks such as used in machine translation, search engines, and text summarization systems.

This research aims to address these challenges by developing a specialized machine learning model tailored to the unique structure of Arabic broken plurals. The model will be trained on a diverse and robust dataset, allowing it to capture the various forms and exceptions of broken plurals. By incorporating advanced NLP techniques and deep learning algorithms, the model will not only improve the accuracy of broken plural detection but also enhance the broader capabilities of Arabic language processing. The anticipated outcome is a significant boost in the efficiency of Arabic AI applications, making them more accessible and reliable in fields such as machine translation, information retrieval, and text analytics. Ultimately, this research seeks to bridge the gap between Arabic and other languages in AI, facilitating its integration into a wide range of technological

innovations and supporting its role in global digital communication.

This study is among the first to focus specifically on computational recognition of Arabic broken plurals using a hybrid approach that combines traditional morphological patterns with machine learning classifiers trained on a richly annotated corpus

1.2 Problem Statement

This project aims to address the following problems:

1. **Resource Scarcity:** The scarcity of high-quality, labeled Arabic datasets limits the training and evaluation of AI models designed for broken plural recognition.
2. **Low Model Accuracy:** Available AI models suffer from low accuracy in distinguishing words as broken plurals or not, whether they appear individually or within a sentence, due to the lack of training on high-quality datasets.

To overcome these challenges, in this project we will

1. build a corpus of Arabic broken plural terms.
2. This corpus will be used to build an AI model capable of analyzing and identifying Arabic broken plurals accurately.

1.3 Objectives

The primary objectives of this project are as follows:

1. **Analyze Linguistic Patterns:** Study the morphological and grammatical rules governing broken plurals in Arabic to establish a foundational understanding.
2. **Develop an AI Model:** Design and train a machine learning model capable of accurately identifying and classifying broken plurals in Arabic text.
3. **Build a Comprehensive Dataset:** Create a sufficiently large and diverse dataset of Arabic text that includes a wide range

of broken plural examples to support model training and evaluation.

4. **Interactive Web Interface:** Develop a user-friendly web interface where users can input Arabic text and receive instant identification of broken plurals within the text.
5. **Enhance NLP Applications:** Improve AI-driven tools such as translation systems, search engines, and text generation software by integrating advanced broken plural recognition capabilities.

1.4 Motivation

Despite being one of the most widely spoken languages globally, Arabic remains underrepresented in AI research, particularly when compared to languages like English. This disparity is evident in the field of Natural Language Processing (NLP), where Arabic presents unique challenges due to its rich morphological structure, including the phenomenon of broken plurals. Addressing these challenges is critical for improving AI's ability to understand and process Arabic text with accuracy and nuance.

The motivation for this project arises from several key factors:

1. Unique Linguistic Challenge

Broken plurals are one of the most intricate and unique aspects of Arabic morphology [29]. They involve non-linear transformations of singular nouns into plural forms, often defying standard grammatical rules. This complexity poses a significant challenge for natural language processing (NLP) and makes broken plurals an ideal testbed for developing advanced AI models capable of handling irregular linguistic structures. Addressing this challenge not only pushes the boundaries of AI in morphologically rich languages like Arabic but also drives innovation in processing complex language patterns globally.

2. Text Generation and Enrichment

Understanding and incorporating broken plurals into text generation systems is essential for improving the

authenticity and expressiveness of Arabic text .By utilizing correct plural forms, these systems can produce smooth and natural outputs that feel closer to human language. This capability enhances linguistic diversity, supports creative writing by enabling richer and more nuanced content, and improves automated content generation, such as news, educational materials, and social media posts. Ultimately, it enriches the user experience by delivering high-quality, human-like Arabic text that aligns with native linguistic conventions.

3. Improved Error Detection and Correction

Understanding broken plurals enables AI systems to provide more accurate error correction in grammar and spelling. This improvement enhances the clarity, coherence, and professionalism of Arabic text, which is particularly valuable for educational tools, automated proofreading software, and professional communication[33].

4. Part of speech:

Introducing the corpus will strengthen part-of-speech identification, refine sentence structure analysis, and facilitate interaction with diverse texts. It will also improve tasks like translation and summarization, enabling the AI to process different types of text more naturally.

5. Promoting Arabic Language Development

Promoting the development of the Arabic language

This project contributes to the development of the Arabic language and ensuring its sustainability in the digital age and making it applicable to modern applications supported by artificial intelligence[30]The plural of brokenness is one of the major linguistic challenges in the Arabic language, and understanding it enhances the ability of digital systems to deal with Arabic texts accurately and naturally.

6. Inclusive and Culturally Relevant AI

Addressing broken plurals in NLP contributes to building AI systems that are inclusive and linguistically accurate for Arabic speakers. This effort ensures that technologies developed for Arabic are culturally relevant and meet the specific needs of their users, ultimately enhancing user satisfaction and engagement.

7. Supporting Education and Linguistics

This project provides valuable resources for educators, linguists, and developers. For educators, it helps improve the teaching of Arabic grammar, particularly complex areas like broken plurals. For linguists, it offers data and analysis that deepen the understanding of Arabic's linguistic challenges. For developers, it provides tools and techniques for creating AI-driven language applications, such as grammar correction software and smart assistants.

8. Broader Impact on AI Research

By tackling the challenges of Arabic broken plurals, this project lays the foundation for processing other morphologically rich languages such as Turkish, Hebrew, and Finnish. It advances culturally aware and linguistically accurate AI systems, fostering inclusivity across NLP research. Beyond its linguistic contribution, the project elevates the digital presence of Arabic—an underrepresented yet complex language—by integrating it into the growing landscape of intelligent multilingual technologies. In doing so, it empowers Arabic speakers, reinforces linguistic diversity, and supports the development of globally relevant and equitable AI solutions.

1.5 Proposed Solution

To develop an AI model capable of detecting broken plurals in Arabic, the following approach was adopted:

1. **Building a Specialized Dataset:** Due to the lack of publicly available resources, a custom corpus was compiled by collecting sentences from diverse Arabic sources. Identified broken plural words were extracted, analyzed, and annotated with key morphological features, including their root, pattern, and plural type (broken, sound masculine, or sound feminine) [27].

2. **Model Development and Selection:**

- Initial experimentation included classical machine learning algorithms like Logistic Regression and Naive Bayes.
- More complex models, such as Support Vector Machines (SVM) and Random Forest, were also trained and evaluated.
- The final model was selected based on its performance on a set of metrics, especially F1-score, due to its relevance in imbalanced data settings.

3. **Future Directions:**

In later phases, the project will explore deep learning models (e.g., LSTM) to capture broader contextual patterns. Efforts will also focus on training the model to handle rare or irregular broken plural forms, as well as conducting comparative studies to enhance grammatical analysis and educational tools.

"Additionally, all steps in this pipeline were documented and versioned to support reproducibility and future model comparisons."

1.6 Thesis structures

This thesis is organized into six chapters as follows:

- Chapter 1: Introduction

Provides an overview of the project, including the background, problem statement, objectives, motivation, proposed solution, and a summary of the thesis structure.

- Chapter 2: Literature Review
Surveys previous research related to Arabic natural language processing (NLP), broken plural detection, and the use of machine learning models in linguistic applications.
- Chapter 3: Dataset Development
Describes the construction of the dataset, covering data collection, annotation, preprocessing, and augmentation techniques.
- Chapter 4: System Architecture and Implementation
Details the system design, including the technical architecture, tools used, model training pipeline, and core components developed during implementation.
- Chapter 5: Results and Evaluation
Presents experimental results, evaluates model performance using relevant metrics, and includes error analysis and comparative assessments.
- Chapter 6: Deployment and Interface
Explains the deployment strategy, user interface development using React, backend integration with Flask, and the structure of the final application.

CHAPTER 2: RELATED WORK

Arabic Natural Language Processing (NLP) has evolved significantly in recent years, yet it continues to lag behind languages like English, primarily due to the rich and intricate morphological structure of Arabic [26]. Among the most complex phenomena in Arabic is broken plural formation, which poses serious challenges to computational models. Unlike sound plurals that follow predictable suffix-based rules, broken plurals involve internal and often irregular changes to word structure. These transformations hinder the performance of many AI-driven

applications such as machine translation, grammar correction, and search engines.

2.1 Arabic NLP Challenges

Arabic is a morphologically rich and highly inflected language with three main types of plural forms: sound masculine, sound feminine, and broken plurals. Broken plurals, in particular, deviate from regular rules and often require advanced morphological insight to analyze accurately. These complexities make standard NLP pipelines inadequate, especially for tasks requiring fine-grained morphological classification.

Despite advances in Arabic NLP, most research has focused on basic preprocessing tasks such as tokenization, stemming, sentiment analysis, and part-of-speech tagging. Morphological analysis—particularly for broken plurals—remains relatively underdeveloped and is often limited to rule-based or shallow approaches [41].

Several linguistic tools have been developed over the years, including:

- Buckwalter Arabic Morphological Analyzer (BAMA) 2.0 [41],
- Its updated version SAMA 3.1, and
- CAMEL Tools, an open-source Python toolkit for Arabic NLP [44].

These tools provide valuable morphological features, yet they are not specialized for broken plural detection, which limits their effectiveness in downstream applications.

2.2 Existing Models and Techniques

Earlier efforts in Arabic morphology relied heavily on rule-based analyzers such as Buckwalter’s system. While effective in specific contexts, these models are constrained by their inability to generalize and often fail to handle contextual or semantic variation.

Later, supervised machine learning models such as Logistic Regression and Naive Bayes were employed in tasks like gender prediction and named entity recognition. These models improved performance but remained dependent on large annotated datasets—which are notably scarce for broken plurals.

In recent years, deep learning and transformer-based models have emerged, offering improved contextual understanding:

- AraBERT by Antoun et al. (2020) demonstrated success in Arabic NLP tasks [40],
- Other models such as ARBERT, MARBERT, and AraT5 further pushed dialectal and classical Arabic understanding using pre-trained transformers [43].

Additionally, hybrid NLP frameworks like MADAMIRA [45] and CAMEL Tools [44] combine rule-based morphological analysis with statistical modeling. Despite these advances, none of these systems are fine-tuned specifically for broken plural recognition, leaving a noticeable gap.

2.3 Research Gaps

Although Arabic NLP has progressed, broken plural recognition remains an underexplored and challenging domain due to several factors:

- A lack of annotated datasets specifically designed for broken plural forms,
- The structural irregularity and linguistic complexity of plural patterns,
- Existing models tend to generalize pluralization without differentiating broken from sound forms,
- A scarcity of educational or visualization tools that demonstrate morphological transformations in an interactive or interpretable manner.

Moreover, even state-of-the-art deep learning models like AraBERT are rarely adapted for this niche task, resulting in low interpretability and reduced effectiveness in real-world applications [43].

These limitations underscore the urgent need for domain-specific datasets, interpretable models, and educational resources to facilitate accurate and explainable broken plural classification.

2.4 Contribution of This Study

This project addresses these research gaps by proposing an integrated system that blends traditional morphological knowledge with modern machine learning techniques. Key contributions include:

- Curating a specialized dataset focusing exclusively on broken plural forms, annotated with contextual examples, morphological weights, and plural types.
- Evaluating multiple ML algorithms (e.g., Logistic Regression, Naive Bayes, SVM, Random Forest) to identify the most effective classifier for this linguistic phenomenon.
- Augmenting the dataset with sound masculine and sound feminine plural forms to build a multi-class classification model.
- Incorporating rule-based features such as suffix analysis and morphological pattern detection to enhance interpretability.
- Developing a web interface using React.js and Flask, enabling real-time classification for linguists, educators, and researchers.

Unlike prior tools, this system is uniquely designed for broken plural analysis—offering not only accurate classification but also transparency, educational value, and ease of access.

By bridging classical Arabic linguistics with machine learning, this project establishes a novel framework for plural pattern recognition in morphologically complex languages. It sets a strong foundation for future studies and applications in Arabic computational morphology, educational platforms, and intelligent language systems.

CHAPTER 3: DATASET BUILDING AND DEVELOPMENT

3.1 Dataset construction:

To construct our Arabic broken plural Dataset, we followed a five-phase methodology outlined as follows:

3.1.1: Data Collection

We collected more than a thousand Arabic broken plural term from various books and websites. Arabic broken plurals are categorized based on the morphological patterns into 23 patterns [29], [30], [31] grouped into two main categories (قلة) and (كثرة). So we collected broken plural terms based on these patterns as follows:

1. Category of (قلة), which subcategorized into:
 - 1.1. The pattern (أَفْعُل): we collected 23 term from the web sites [10-13], and 50 terms from the books [5-9].
 - 1.2. The pattern (أَفْعَال): we collected 25 term from the web sites [20-25], and 40 terms from the books [1-3].
 - 1.3. The pattern (أَفْعِلَة): we collected 22 term from the web sites [13-14], and 42 terms from the books [6-8].
 - 1.4. The pattern (فَعْلَة): we collected 30 term from the web sites [13-17], and 10 terms from the books [6-8].

2. Category of (كثرة), which subcategorized into:
 - 2.1. The pattern (فُعْل): we collected 35 term from the web sites [12-15], and 57 terms from the books [1-9].
 - 2.2. The pattern (فُعْل): we collected 23 term from the web sites [17-19], and 20 terms from the books [1-9].
 - 2.3. The pattern (فُعْل): we collected 36 term from the web sites [19-25], and 17 terms from the books [1-9].
 - 2.4. The pattern (فُعْل): we collected 25 term from the web sites [20-21], and 19 terms from the books [1-9].
 - 2.5. The pattern (فُعْلَة): we collected 23 term from the web sites [15,17,23], and 14 terms from the books [1-9].
 - 2.6. The pattern (فُعْلَة): we collected 25 term from the web sites [17-25], and 31 terms from the books [1-9].
 - 2.7. The pattern (فُعْلَى): we collected 40 term from the web sites [10-15], and 15 terms from the books [1-9].
 - 2.8. The pattern (فُعْلَة): we collected 20 term from the web sites [21-25], and 20 terms from the books [1-9].
 - 2.9. The pattern (فُعْل): we collected 18 term from the web sites [20-23], and 36 terms from the books [1-9].
 - 2.10. The pattern (فُعَال): we collected 23 term from the web sites [16,20], and 40 terms from the books [1-9].
 - 2.11. The pattern (فُعَال): we collected 13 term from the web sites [10-12], and 22 terms from the books [1-9].
 - 2.12. The pattern (فُعُول): we collected 23 term from the web sites [23-25], and 20 terms from the books [1-9].
 - 2.13. The pattern (فُعْلَان): we collected 42 term from the web sites [19-22], and 28 terms from the books [1-9].
 - 2.14. The pattern (فُعْلَان): we collected 30 term from the web sites [13,25], and 15 terms from the books [1-9].
 - 2.15. The pattern (فُعْلَاء): we collected 17 term from the web sites [15-17], and 21 terms from the books [1-9].
 - 2.16. The pattern (أَفْعِلَاء): we collected 21 term from the web sites [15-25], and 13 terms from the books [1-9].
 - 2.17. The pattern (فَوَاعِل): we collected 28 term from the web sites [2-5], and 15 terms from the books [1-9].
 - 2.18. The pattern (فُعَائِل): we collected 20 term from the web sites [2-5], and 14 terms from the books [1-9].

- 2.19. The pattern (فَعَالِي): we collected 31 term from the web sites [13-23], and 24 terms from the books [1-9].
- 2.20. The pattern (فَعَالِي): we collected 13 term from the web sites [2-5], and 12 terms from the books [1-9].
- 2.21. The pattern (فَعَالِي): we collected 23 term from the web sites [22-23], and 13 terms from the books [1-9].
- 2.22. The pattern (فَعَالِل): we collected 27 term from the web sites [24], and 18 terms from the books [1-9].
- 2.23.1 The pattern ((شِبْه فَعَالِل مَفَاعِل)): we collected 18 term from the web sites [24-25], and 12 terms from the books [1-9].
- 2.23.2 The pattern ((شِبْه فَعَالِل فَوَاعِل)): we collected 23 term from the web sites [24-25], and 22 terms from the books [1-9].
- 2.23.3 The pattern ((شِبْه فَعَالِل فَعَاعِل)): we collected 25 term from the web sites [20-25], and 17 terms from the books [1-9].

3.1.2 Word Annotation

In the phase following data collection, we implemented a comprehensive word annotation process. Each of the 1,000+ collected Arabic broken plural terms was carefully examined and labeled with its corresponding morphological pattern and plural type.

Specifically, each word was analyzed in context and annotated with:

- the token term,
- its morphological pattern (e.g., "مفاعل", "فُعَل"),
- and its classification as either a plural of paucity (قِلَّة) or a plural of abundance (كَثْرَة).

For example, in the sentence "شَاهَدْتُ الْكُتُبَ الْجَدِيدَةَ فِي الْمَكْتَبَةِ", the word "الْكُتُبَ" was annotated with the pattern "فُعَل" and classified under the كَثْرَة category.

This detailed annotation enriches the dataset and establishes a strong foundation for analysis and AI model development by providing key linguistic features for each instance.

3.1.3 Data Documentation

In the Data Documentation phase, all annotated terms were systematically recorded in structured formats (e.g., spreadsheets or databases) with essential details such as:

- the word itself,
- its morphological pattern,
- plural type,
- and an illustrative example sentence.

This structured approach enables efficient retrieval and supports consistent analysis. Furthermore, strict quality control procedures were applied to ensure the accuracy and reliability of the data, forming a solid base for subsequent machine learning processes.

3.1.4: Exploring New Patterns

In this phase, the dataset is analyzed to identify both conventional and unconventional patterns in Arabic broken plurals. This exploration contributes to a deeper understanding of Arabic morphology and helps refine computational linguistic models.

To systematically explore and document new patterns in Arabic broken plurals, the following structured approach was adopted:

1. Reviewing the Data:
 - Each plural form was examined to determine whether it followed classical morphological patterns (e.g., “فُعُل”, “مَفَاعِل”, “أَفْعَال”) or deviated from them.
2. Identifying Conventional Patterns:
 - Common broken plural structures were validated against traditional Arabic grammar to ensure the accuracy and linguistic validity of the dataset.
3. Detecting Unconventional or Novel Patterns:

Words that did not conform to expected grammatical norms were identified and categorized as follows:

- Rare Forms: Plurals that are infrequently used in modern or classical Arabic. Examples include “غزلان” (from “غزال”), “أعقاب” (from “عقب”), and “أجراء” (from “جرو”).
- Irregular Patterns: Plural forms that deviate from standard broken plural structures or appear semantically inconsistent. Examples include “آبار” (from “بئر”), “نساء” (from “امرأة”), and “أشياء” (from “شيء”).
- Hybrid Constructions: Non-standard or dialect-influenced patterns often found in colloquial or creative language usage. Examples include “شبان” (from “شاب”) and “حيات” (from “حوت”).

Note: Forms like شبابات, عقبات, and حوات are intentionally fabricated to contrast with valid broken plurals. They are used strictly for illustrative purposes and do not exist in standard Arabic.

4. Comparative Analysis & Documentation:

The identified unconventional and novel plural forms were further examined in light of classical Arabic grammatical references to validate their linguistic authenticity and structural integrity. This comparative analysis reinforces the credibility of the dataset and supports its use in advanced computational linguistic applications.

This step was essential for bridging the gap between practical data collection and traditional Arabic morphology. By comparing plural patterns against authoritative linguistic sources, the analysis ensures that both widely recognized and less common forms are accurately documented and understood within their correct grammatical context.

****Examples of Unconventional or Novel Patterns in Arabic Broken Plurals:****

1. ****Rare Forms**** (Plurals that are infrequent in standard usage):

- (دَعَاوِي) → (دَعَوَى) *(instead of the more expected form "دَعَوَات")*
 - (وَصَايَا) → (وَصِيَّة) *(constructed example: "وَصِيَّات")*
 - (أَجْرَاء) → (جَرَوْ) *(a rare form compared to "جَرَاء")*
2. ****Irregular Patterns**** (Structures that do not fit traditional morphological models):
- (أَرْمَلَات) → (أَرْمَلَة) *(instead of expected "أَرْمَلَات")*
3. ****Hybrid Constructions**** (Mixed patterns influenced by dialects or creative usage):
- (شُبَّان) → (شِبَاب) *(constructed alternative: "شِبَابَات")*
 - (حَوْتَات) → (حَوْت) *(instead of the non-standard form "حَوْتَات")*

3.1.5: Analysis and Comparison

After completing the annotation phase, a comprehensive linguistic comparison was conducted to verify the morphological accuracy of all collected broken plural forms. This involved systematically reviewing each word against the standard patterns of Arabic plural morphology as documented in trusted linguistic references .

The goal of this step was to validate that each plural entry adheres to recognized morphological structures or, where applicable, falls within accepted linguistic exceptions. This ensured the internal consistency of the dataset and confirmed its alignment with classical and contemporary grammatical norms.

The analysis revealed a wide range of broken plural patterns—from the canonical forms documented in classical Arabic grammar to rare, irregular, and colloquially-influenced constructions . This layered understanding provides both linguistic credibility and practical insight for model training and evaluation in Arabic NLP.

3.2 Data Processing: Cleaning, Organizing, and Preparing for Modeling

Data processing is a crucial step in this research. It aims to prepare the collected Arabic sentences—each containing one or more broken plural forms—for downstream analysis and classification using machine learning algorithms.

Given the morphological complexity of Arabic broken plurals, this stage required detailed manual review and linguistic validation. The process involved the following steps:

- Cleaning the text by removing irrelevant characters and non-linguistic elements,
 - Extracting the broken plural word as it appeared in context,
 - Trimming the word from prefixes, suffixes, or the definite article (ال) to yield the base form,
 - Analyzing its morphological pattern,
 - Classifying broken plural word as either "قلة" or "كثرة",
 - Structuring the annotated data into a unified, machine-readable format suitable for training.
- In total, 1,047 sentences were processed. These were carefully selected from a wide range of sources, including literary, scientific, religious, and educational texts, to ensure linguistic richness and diversity.

3.2.1 Preliminary Data Analysis

Before diving into model development, a targeted linguistic analysis was carried out to assess the representativeness and morphological richness of the dataset. This step was crucial to ensure that the dataset includes a wide variety of broken plural patterns, which are essential for building a generalizable model. A diverse set of patterns was observed during this analysis, including well-known broken plural forms such as:

- مَنَاصِب, مَقَاعِد (e.g. مَفَاعِل)
- شُرُوط, حُرُوب (e.g. فُعُول)
- أَغْصَان, أَكْوَاب (e.g. أَفْعَال)

among others. These patterns were documented and categorized based on their frequency and structure, providing valuable insight into the morphological behavior of Arabic plurals. This analysis also helped identify underrepresented patterns, informing decisions for potential dataset augmentation[2][3].

3.2.2 Data Cleaning and Normalization

To ensure data consistency and quality, the following non-linguistic elements were removed:

- Punctuation
- Numbers
- Duplicate words
- Sentences without a correct broken plural

In addition, the diacritical variations of the morphological weights were standardized.

Each sentence was standardized using consistent formatting rules. Diacritical marks were initially applied using the **Farasa Arabic Diacritizer** and then manually corrected to ensure morphological accuracy. Diacritics were initially applied using the **Farasa toolkit**[32], and then manually corrected to ensure precision in morphological interpretation.

3.2.3 Manual Lemmatization of Target Words

Each broken plural word was **manually extracted** from its context and stripped of:

- the definite article “ال”,
- **nunation** (تنوين),
- and other inflectional affixes, to produce its root form(target)

This step was fully manual to ensure high precision, especially in cases where automated tools might fail or introduce errors.

3.2.4 Morphological Weight Assignment

After lemmatization, each word was manually assigned its correct **Arabic morphological pattern (الوزن الصرفي)** using authoritative Arabic morphology references and expert validation. The most common broken plural patterns used for classification included:

Singular (المفرد)	Broken Plural (الجمع)	Morphological Pattern (الوزن الصرفي)
كِتَاب	كُتُب	فُعْل
عُصْن	عُصُون	فُعُول
مَسْجِد	مَسَاجِد	مَفَاعِل
جَرِيدَة	جَرَائِد	فَعَائِل

Table 3.1 showcases common examples of Arabic broken plurals, demonstrating the non-linear transformations between singular and plural forms based on morphological patterns.

3.2.5 Plural Type Classification

Based on the assigned morphological pattern, each word was automatically classified as follows:

- **Broken Plural of paucity "قِلَّة"** if it matched known sound patterns such as:
 - أَفْعُل
 - أَفْعَال
 - أَفْعَلَة
 - فَعْلَة
- **Broken Plural of many "كَثْرَة"** if it did **not** conform to the above patterns and followed irregular broken plural structures.

This classification was automated after the weight assignment phase to ensure consistency and efficiency.

3.2.6 Database Organization

The processed data was structured into a unified format and stored in a **CSV file** encoded with **UTF-8** to ensure compatibility with various processing and modeling tools. The dataset included the following key columns:

- **Original Sentence:** The complete sentence from which the target word was extracted.
- **Token:** The target word as it originally appeared in the sentence.
- **Abstract target word :** The basic or abstract form of the token word (without suffixes).
- **Morphological Pattern:** The assigned Arabic morphological weight (e.g., أَفَاعِل, فُعُول).
- **Plural Type:** Classification as **Broken Plural of many** "كثرة" or **Broken Plural of paucity** "قلّة".

BP_type	morph_weight	target	token	sentence	
					1
جمع كثرة	مفاعِل	مَخَابِر	المَخَابِر	يُوجَدُ العَدِيدُ مِنَ المَخَابِرِ فِي الحَيِّ.	2
جمع قلة	أفْعَال	أَسْمَاك	الأسْمَاك	يَهْوَى بَغْضُ النَّاسِ تَرْبِيَةَ الأسْمَاكِ فِي الأخْوَاضِ .	3
جمع قلة	أفْعَال	أَوْزَان	أَوْزَان	يَقِيْسُ الرِّيَاضِيُّ أَوْزَانَ الحَدِيدِ فِي التَّمَارِينِ.	4
جمع كثرة	مفاعِل	مَنَاصِح	مَنَاصِح	يُقَدِّمُ الأسْتَاذُ مَنَاصِحَ ثَمِينَةَ لِلطُّلَابِ لِتَحْقِيقِ النِّجَاحِ .	5
جمع كثرة	مفاعِل	مَجَانِين	المَجَانِينِ	يُقَالُ إِنَّ المَجَانِينَ أحيانًا يَرَوْنَ الحَقِيقَةَ بِشَكْلِ مُخْتَلَفٍ .	6
جمع كثرة	مفاعِل	مَسَاكِن	مَسَاكِن	يَعِيشُ النَّاسُ فِي مَسَاكِنَ مُرِخَةٍ وَمُجَهَّزَةٍ بِجَمِيعِ الخِدْمَاتِ .	7
جمع كثرة	فَوَاعِل	صَوَامِع	صَوَامِع	يَعِيشُ الرُّهْبَانُ فِي صَوَامِعَ صَغِيرَةٍ لِلتَّأَمُّلِ وَالْعِبَادَةِ .	8
جمع كثرة	مفاعِل	مَنَاجِل	مَنَاجِل	يَعْمَلُ المَزَارِعُونَ بِاسْتِخْدَامِ مَنَاجِلَ حَدِيدَةٍ لِخَصْدِ المَحَاصِيلِ .	9
جمع كثرة	فُعَال	عُمَال	العُمَالِ	يَعْمَلُ العُمَالُ بِجِدِّ فِي مَوْقِعِ البِنَاءِ.	10
جمع كثرة	فِعْل	بِئَع	البِئَعِ	يَعْمَلُ العُمَالُ بِجِدِّ فِي المَصْنَعِ لِضَمَانِ إِيْتَاكِ البِئَعِ بِأَعْلَى جُودَةٍ.	11
جمع كثرة	مفاعِل	مَصَادِر	المَصَادِرِ	يَعْتَمِدُ الطُّلَابُ عَلَى المَصَادِرِ المَوْثُوقَةِ فِي أَبحَاثِهِمْ .	12

Figure 3.1 Sample of from the final structured dataset.

Additionally, Table 3.2 below summarizes the core structure of the dataset as a tabular representation of column descriptions and data roles:

تَدْرِبُ أَشْبَالُ الكَشَافَةِ عَلَى المَهَارَاتِ الجَدِيدَةِ.	أَشْبَالُ	أَشْبَالُ	أَفْعَال	جمع قلة
تَخْتَلِفُ مَقَاصِدُ النَّاسِ فِي السَّفَرِ.	مَقَاصِدُ	مَقَاصِدُ	مَفَاعِل	جمع كثرة
تَخْتَلِفُ فُصُولُ السَّنَةِ بَيْنَ الصَّيْفِ وَالشِّتَاءِ.	فُصُولُ	فُصُولُ	فُعُول	جمع كثرة
تَزَيِّنُ المَسْجِدُ بِالمَحَارِيبِ المُرْخَرَفَةِ.	بِالمَحَارِيبِ	مَحَارِيب	مَفَاعِل	جمع كثرة
تَزِيدُ الفِرَقُ الرِّيَاضِيَّةُ الأَقْمَصَةَ المُمَيَّزَةَ .	الأَقْمَصَةَ	أَقْمَصَةَ	أَفْعَلَة	جمع قلة
الكَرَمُ وَالشَّجَاعَةُ مِنْ أَمْزَجِ خِصَالِ القَادَةِ النَّاجِحِينَ.	خِصَالِ	خِصَالِ	فِعَال	جمع كثرة
عِنْدَمَا نَظَرْتُ إِلَى السَّمَاءِ ، كَانَتْ الأَعْيُنُ تُرَاقِبُ كُلَّ حَرَكَةٍ بِدِقَّةٍ وَاهْتِمَامٍ .	الأَعْيُنُ	أَعْيُن	أَفْعَل	جمع قلة

Table 3.2: summarizes the core structure of the dataset

3.3 Deriving Additional Linguistic Features

To enrich the dataset linguistically and enhance its utility for both computational modeling and educational purposes, two additional features were integrated into each broken plural entry:

meaning	BP_type	morph_weight	target	token	sentence
Courts	جمع كثرة	مَفَاعِل	مَحَاكِم	المَحَاكِمُ	تَنْظُرُ المَحَاكِمُ فِي القَضَايَا المِهْمَةِ الَّتِي تُؤَثِّرُ عَلَى المُجْتَمَعِ
constellations	جمع قلة	أَفْعَال	أَبْرَاجِ	بِأَبْرَاجِهَا	تَشْتَهَرُ دُبِّي بِأَبْرَاجِهَا العَالِيَةِ.
Beaks	جمع كثرة	مَفَاعِل	مَنَاقِيرِ	مَنَاقِيرَهَا	تَسْتَخْدِمُ الطُّيُورُ مَنَاقِيرَهَا لِجَمْعِ الطَّعَامِ وَبِنَاءِ الأعْشَاشِ.
Stars	جمع كثرة	فُعُول	نُجُومِ	النُّجُومَ	تَأْمَلْتُ النُّجُومَ اللَّامِعَةَ فِي السَّمَاءِ الصَّافِيَةِ.

Systems	جمع كثرة	فَعْل	نُظْم	النُّظْم	النَّظْمُ الْحَدِيثَةُ تُسَهِّلُ إِدَارَةَ الْمُؤَسَّسَاتِ بِشَكْلِ فَعَالٍ.
Newspapers	جمع كثرة	فَعَائِل	صَحَائِف	الصَّحَائِفُ	الْصَّحَائِفُ الْقَدِيمَةُ تَحْتَفِظُ بِتَفَاصِيلَ دَقِيقَةٍ عَنْ حَيَاةِ الْأَجْدَادِ

Table 3.3: Sample English translations for broken plural lemmas used in semantic enrichment and multilingual tasks.

3.3.1 English Translation

Each lemmatized word was manually translated into English using reputable bilingual dictionaries such as Al-Mawrid, Almaany [50][51], and verified with online lexicons including Google Translate and Wiktionary. This provides semantic clarity, supports cross-linguistic tasks, and facilitates applications in translation and language education.

3.3.2 Latin Transliteration

Latin transliteration refers to converting Arabic words into a standardized Latin-alphabet phonetic form that approximates Arabic pronunciation. This facilitates Arabic learning and makes the dataset more accessible for non-native speakers and multilingual AI systems.

The transliteration was conducted using the PyArabic library GitHub [52], which provides a programmatic interface to convert Arabic text into Latin script based on established phonetic conventions. A custom script (see Appendix A) was used to automate this process, and all outputs were manually reviewed to correct common errors and confirm alignment with Arabic phonology.

transliteration	target	token	sentence
-----------------	--------	-------	----------

<somaAk	أَسْمَاك	الْأَسْمَاك	يَهْوَى بَعْضُ النَّاسِ تَرْبِيَةَ الْأَسْمَاكِ فِي الْأَخْوَاضِ.
maSaAdir	مَصَادِر	الْمَصَادِر	يَعْتَمِدُ الطَّلَابُ عَلَى الْمَصَادِرِ الْمُوثِقَةِ فِي أبحاثِهِمْ.
wm	لُحُوم	اللُّحُوم	يَجِبُ تَخْزِينُ اللَّحُومِ فِي دَرَجَاتِ حَرَارَةٍ مُنْخَفِضَةٍ.
abariiQ	أَبَارِيقَ	الْأَبَارِيقَ	وَضَعَتِ الْوَالِدَةُ الْأَبَارِيقَ عَلَى الطَّائِلَةِ لِتَقْدِيمِ الشَّايِ لِلزُّوَارِ
khanaaziir	خَنَازِيرَ	خَنَازِيرَ	فِي الْمَرْعَةِ، كَانَ هُنَاكَ عِدَّةُ خَنَازِيرَ تَتَجَوَّلُ بِحُرِّيَّةٍ
qwArb	قَوَارِبَ	الْقَوَارِبَ	عَبَرَ الْفَلَّاحُونَ الْأَنْهَارَ بِاسْتِخْدَامِ الْقَوَارِبِ.

Table 3.4: Sample of Latin transliteration for broken plural lemmas.

These features were stored in two additional columns: Meaning and Transliteration, transforming the dataset from a purely morphological resource into a linguistically enriched corpus suitable for education, research, and cross-linguistic applications.

transliteration	meaning	BP_type	morph_weight	target	token	sentence
<agoTiyap	Covers	جمع قلة	أَفْعَلَة	أَعْطِيَة	الأَعْطِيَة	وَضَعَتْ الأُمُّ الأَعْطِيَة عَلَى الأسْرَة فِي الشِّتَاءِ .
<TobaAq	Dishes	جمع قلة	أَفْعَال	أَطْباق	الأَطْباق	وَضَعَتْ الأَطْباقُ عَلَى الطَّاولَة اسْتِعْدَادًا لِلْعَشَاءِ .
wjwh	Faces	جمع كثرة	فُعُول	وُجوه	وُجوه	كَانَتْ الصَّوْرَةُ تُزَيِّنُهَا وُجوهٌ كَأَنَّهَا وُجوهُ المَلَائِكَةِ .
<anaAmili	Fingers	جمع كثرة	أَفَاعِل	أَنَامِل	بِأَنَامِلِهِ	عَزَفَ العَازِفُ عَلَى أَلْيَانُو بِأَنَامِلِهِ بِرَاعَةٍ

Table 3.5 Final Result: represents the full structured dataset used for model training.

3.4 Expanding the Dataset with New Plural Types

To enhance the model's performance and enable more comprehensive classification of Arabic plural forms, the dataset was expanded to include two additional morphological types: Sound Masculine Plural and Sound Feminine Plural. This step aims to support multi-class classification, allowing the model to distinguish broken plurals from other plural structures that follow more regular morphological patterns.

3.4.1 Inclusion of Sound Masculine Plural

To ensure that the model could effectively differentiate between broken plurals and regular plural forms, a set of sound masculine plural entries was added. These forms typically follow a

consistent morphological pattern, making them ideal for teaching the model contrastive structures.

Sentences containing sound masculine plural forms (e.g., "مهندسون", "معلمون") were collected from literary and educational texts. For each sentence, the following steps were performed:

- The plural token was extracted from its sentence,
- *The token was normalized by removing affixes such as the definite article "ال", plural suffixes, and other morphological additions to extract its base singular form (lemma) for precise linguistic analysis.*
- The morphological weight (e.g., "فَاعِلُونَ") was assigned based on Arabic grammatical rules,
- Each entry was tagged as "جمع المذكر السالم" (Sound Masculine Plural),
- English translation and Latin transliteration were added.

The addition of this category supports model robustness and mirrors real-world applications where systems must classify between multiple types of Arabic plurals

HA'irUna	Confused people	جمع المذكر السالم	فَاعِلُونَ	حَائِرُونَ	الحائرون يَحْتَوْنَ عَنْ إجاباتٍ شافيةٍ لِمَشاكِلِهِمْ.
muEtamirUna	Pilgrims	جمع المذكر السالم	فَاعِلُونَ	مُعْتَمِرُونَ	المُعْتَمِرُونَ يُؤَدُّونَ مَناسِكَ العُمْرَةِ بِخُشُوعٍ وَتَقْوَى.
mutawakkilUna	Trusting people	جمع المذكر السالم	مُفْعِلُونَ	مُتَوَكِّلُونَ	الْمُتَوَكِّلُونَ يَعْثَمِدُونَ عَلَى اللَّهِ فِي كُلِّ أَعْمَالِهِمْ.
muEtamadUna	Trusted people	جمع المذكر السالم	مُفْعِلُونَ	مُعْتَمِدُونَ	المُعْتَمِدُونَ يَقُومُونَ بِمَهَامِهِمْ عَلَى أَكْمَلِ وَجْهِهِ وَمَسْئُولِيَّةٍ.
mujbarUna	Forced people	جمع المذكر السالم	مُفْعِلُونَ	مُجْبَرُونَ	الْمُجْبَرُونَ يَضْطَرُّونَ إِلَى اتِّخَاذِ قَرَارَاتٍ صَعْبَةٍ فِي ظُرُوفٍ قَاسِيَةٍ.
maqtUIUna	Killed people	جمع المذكر السالم	مُفْعِلُونَ	مُقْتُولُونَ	الْمُقْتُولُونَ فِي الْحُرُوبِ يَسْتَحْفِقُونَ الذِّكْرَى وَالتَّكْرِيمَ الْأَبَدِيِّينَ.
muTallaqUna	Divorced people	جمع المذكر السالم	مُفْعِلُونَ	مُطَلَّقُونَ	الْمُطَلَّقُونَ يَعِيشُونَ حَيَاةً جَدِيدَةً بَعْدَ الطَّلَاقِ بِتَقَاوُلٍ وَأَمَلٍ.
HAfiZ.unA	Guardians	جمع المذكر السالم	مُفْعِلُونَ	حَافِظُونَ	الحافظون لِلْقُرْآنِ يَتَعَلَّمُونَ آيَاتِهِ بِتَدَبُّرٍ وَتَرْتِيلٍ.

musawwirUna	Wall-builders	جمع المذكر السالم	فَاعِلُونَ	مُسَوِّرُونَ	مُسَوِّرُونَ	الْمُسَوِّرُونَ يَحْمُونَ الْمَمْلَكَاتِ مِنَ السَّرِقَةِ بِجُهْدٍ مُسْتَمِرٍّ.
mankUbUna	Calamity-stricken people	جمع المذكر السالم	مُفْعَلُونَ	مَنْكُوبُونَ	الْمَنْكُوبُونَ	الْمَنْكُوبُونَ يَتَلَقَّوْنَ الْمُسَاعَدَاتِ مِنَ الْجِهَاتِ الْإِغَائِيَّةِ بِسُرْعَةٍ.
sAjidUna	Those who prostrate	جمع المذكر السالم	مَفْعُولُونَ	سَاجِدُونَ	السَّاجِدُونَ	السَّاجِدُونَ يَلْتَزِمُونَ بِإِدَاءِ الصَّلَاةِ بِخُشُوعٍ وَصِدْقٍ.
muHammalUna	Loaded people	جمع المذكر السالم	فَاعِلُونَ	مُحْمَلُونَ	الْمُحْمَلُونَ	الْمُحْمَلُونَ يَحْمِلُونَ الْأَمْعَةَ الثَّيْلَةَ بِسُهُولَةٍ وَنُظْمٍ.
muEtaqalUna	Arrested people	جمع المذكر السالم	مُفْعَلُونَ	مُعْتَقَلُونَ	الْمُعْتَقَلُونَ	الْمُعْتَقَلُونَ يَعَانُونَ ظُرُوفًا صَعْبَةً دَاخِلَ السُّجُونِ.
muD.TaribUna	Disturbed people	جمع المذكر السالم	مُفْعَلُونَ	مُضْطَرِبُونَ	الْمُضْطَرِبُونَ	الْمُضْطَرِبُونَ يَطْلُبُونَ الدَّعْمَ النَّفْسِيَّ لِتَحْسِينِ حَالَتِهِمْ.
mu'aDh_dhinUna	Muezzins	جمع المذكر السالم	مُفْعَلُونَ	مُؤَذِّنُونَ	الْمُؤَذِّنُونَ	الْمُؤَذِّنُونَ يُؤَدُّونَ الْأَذَانَ بِدِقَّةٍ وَانْتِظَامٍ خَمْسَ مَرَّاتٍ يَوْمِيًّا.

Table 3.6 examples of the sound masculine plural entries added to the dataset.

transliteration	meaning	BP_type	morph_weight	target	token	sentences	1
mubarmijUna	Programmers	جمع المذكر السالم	مُفْعَلُونَ	مُبْرَمِجُونَ	مُبْرَمِجُونَ	مُبْرَمِجُونَ يُطَوِّرُونَ التَّطبيقاتَ الحديثةَ.	2
muhandisUna	Engineers	جمع المذكر السالم	مُفْعَلُونَ	مُهَنْدِسُونَ	مُهَنْدِسُونَ	مُهَنْدِسُونَ يَتِمَرَّنُونَ عَلَى المَتَارِيعِ المُعْكِزَةِ.	3
hirafiyUna	Artisans	جمع المذكر السالم	مُفْعَلُونَ	حِرَافِيُونَ	حِرَافِيُونَ	حِرَافِيُونَ يَصْنَعُونَ المُنْتَخَفَاتِ البَنِيَّةَ.	4
SaHafiyUna	Journalists	جمع المذكر السالم	مُفْعَلُونَ	سَهَافِيُونَ	سَهَافِيُونَ	سَهَافِيُونَ يُنَاقِشُونَ الأَخْبَارَ بِالنَّمَّةِ.	5
faliAHUna	Farmers	جمع المذكر السالم	مُفْعَلُونَ	فَلَكَّارُونَ	فَلَكَّارُونَ	فَلَكَّارُونَ يُزَرِّعُونَ الحُكُومَ بِالخَفَافِ.	6
raHHAUna	Travelers	جمع المذكر السالم	مُفْعَلُونَ	رَحَّالُونَ	رَحَّالُونَ	رَحَّالُونَ يَسَافِرُونَ إِلَى بِلَادٍ بَعِيدَةٍ.	7
muHAmUna	Lawyers	جمع المذكر السالم	مُفْعَلُونَ	مُحَاوِلُونَ	مُحَاوِلُونَ	مُحَاوِلُونَ يَنَاقِشُونَ عَنِ المُتَّهَمِينَ فِي المَحَاكِمِ.	8
mudarrisUna	Teachers	جمع المذكر السالم	مُفْعَلُونَ	مُدَرِّسُونَ	مُدَرِّسُونَ	مُدَرِّسُونَ يُعَلِّمُونَ الأَطْلَالَ فِي المَنَازِلِ.	9
mu'arikhUna	Historians	جمع المذكر السالم	مُفْعَلُونَ	مُؤَرِّخُونَ	مُؤَرِّخُونَ	مُؤَرِّخُونَ يُؤَرِّقُونَ الأَخْبَارَ القَدِيمَةَ.	10
muzArieUna	Farmers	جمع المذكر السالم	مُفْعَلُونَ	مُزَارِعُونَ	مُزَارِعُونَ	مُزَارِعُونَ يُزَوِّقُونَ المَزَارِعَ بِحِرَافَةٍ.	11
najjArUna	Carpenters	جمع المذكر السالم	مُفْعَلُونَ	نَجَّارُونَ	نَجَّارُونَ	نَجَّارُونَ يَصْنَعُونَ الأثاثَ المُتَشَدِّدَ.	12
rA'idUna	Pioneers	جمع المذكر السالم	مُفْعَلُونَ	رَافِعُونَ	رَافِعُونَ	رَافِعُونَ يَطَوِّرُونَ المَتَارِيعَ الأَدْنَى.	13
jaghrAfiyUna	Geographers	جمع المذكر السالم	مُفْعَلُونَ	جُغَرَّافِيُونَ	جُغَرَّافِيُونَ	جُغَرَّافِيُونَ يَتَرَسَّسُونَ القَتَارِيعَ وَالمَنَاقِصَ.	14
muHalilUna	Analysts	جمع المذكر السالم	مُفْعَلُونَ	مُحَلِّلُونَ	مُحَلِّلُونَ	مُحَلِّلُونَ يُفَسِّرُونَ البَيِّنَاتِ بِدِقَّةٍ.	15
masAfirUna	Travelers	جمع المذكر السالم	مُفْعَلُونَ	مَسَافِرُونَ	مَسَافِرُونَ	مَسَافِرُونَ يَسَافِرُونَ المَدَائِمَ.	16
musAEidUna	Assistants	جمع المذكر السالم	مُفْعَلُونَ	مُسَاعِدُونَ	مُسَاعِدُونَ	مُسَاعِدُونَ يُقَدِّمُونَ المَرْءَ لِأَخْرَاجِهِ.	17
mustashArUna	Consultants	جمع المذكر السالم	مُفْعَلُونَ	مُسْتَشَارُونَ	مُسْتَشَارُونَ	مُسْتَشَارُونَ يُقَدِّمُونَ النُصِيحَةَ المُهِمَّةَ.	18
lughawiyUna	Linguists	جمع المذكر السالم	مُفْعَلُونَ	لُغَوِيُونَ	لُغَوِيُونَ	لُغَوِيُونَ يُحَلِّلُونَ لُغَمَ الأُمَّتِ.	19
mudaribUna	Coaches	جمع المذكر السالم	مُفْعَلُونَ	مُدَرِّبُونَ	مُدَرِّبُونَ	مُدَرِّبُونَ يُؤَدِّبُونَ الفِرَقَ لِلتَّائِيْدِ.	20

Figure 3.2: Sample visualization of annotated Sound Masculine Plural entries.

3.4.2 Inclusion of Sound Feminine Plural

To further strengthen the model's ability to distinguish between various plural forms, **sound feminine plural** entries were also added [29], [30], [31]. These forms are typically regular and end in the suffix "-ات" (e.g., "طالبات", "مهندسات"), making them morphologically distinct from both masculine and broken plural types.

Sentences containing sound feminine plural forms were compiled from the same sources, ensuring contextual diversity. Each entry was processed as follows:

- The plural word was identified and extracted,
- Lemmatization was performed by removing the plural suffix and any affixes to retrieve the base singular form,
- The appropriate morphological weight (e.g., "فَاعِلَات") was assigned,
- The word was tagged as "جمع المؤنث السالم" (Sound Feminine Plural),
- English translation and Latin transliteration were added to support multilingual applications.

qAri}aAt	جمع مؤنث سالم	readers	فَاعِلَات	قَارِئَات	القَارِئَات	القارئات يجدن متعة في استكشاف الكتب الجديدة.
mbdiEaAt	جمع مؤنث سالم	creators, innovators	مُفْعِلَات	مَبْدِعَات	المَبْدِعَات	المبدعات يصمن أعمالاً فنية رائعة.
bAHivaAt	جمع مؤنث سالم	researchers	فَاعِلَات	بَاحِثَات	البَاحِثَات	الباحثات يقدمن دراسات علمية متميزة.
mdiyraAt	جمع مؤنث سالم	managers	فَاعِلَات	مَدِيرَات	المَدِيرَات	المديرات يشرفن على تنظيم العمل بكفاءة.
mdr~isaAt	جمع مؤنث سالم	instructors, female teachers	مُفْعِلَات	مَدْرَسَات	المَدْرَسَات	المدرسات يقدمن الشروحات بطريقة مبسطة.
AEiraAt\$	جمع مؤنث سالم	poets	فَاعِلَات	شَاعِرَات	الشَاعِرَات	الشاعرات يكتبن أبياتاً مليئة بالمشاعر.
faA}izaAt	جمع مؤنث سالم	winners	فَاعِلَات	فَائِزَات	الفَائِزَات	الفائزات حصلن على جوائز تقديرية لإنجازاتهم.
mumav~ilAt	جمع مؤنث سالم	actresses	مُفْعِلَات	مُمَثِّلَات	المُمَثِّلَات	الممثلات قدمن أدواراً مؤثرة في المسرح.
mHAmiyAt	جمع مؤنث سالم	lawyers	مُفَاعِلَات	مَحَامِيَّات	المَحَامِيَّات	المحاميات يدافعن عن حقوق المظلومين بشجاعة.
mmr~iDaAt	جمع مؤنث سالم	nurses	مُفْعِلَات	مَمْرُضَات	المَمْرُضَات	المرمرضات يسهرن على راحة المرضى ليلاً ونهاراً.
mr\$idaAt	جمع مؤنث سالم	guides, counselors	مُفْعِلَات	مَرشِدَات	المَرشِدَات	المرشدات يقدمن نصائح قيمة للطلاب الجدد.
mSm~imaAt	جمع مؤنث سالم	designers	مُفْعِلَات	مَصْمِمَات	المَصْمِمَات	المصممات يعملن على تصميم ملابس بأحدث الصيحات.

Table 3.7: Examples of Sound Feminine Plural

G	F	E	D	C	B	A	
transliteration	Blural-type	meaning	morph_weight	target	token	sentence	
TAIbAt	جمع مؤنث سالم	Female students	فَاعِلَات	طَالِبَات	الطَالِبَات	الطَالِبَات فِي الصَّفِّ الْأَوَّلِ مَشْغُوقَاتٌ جَدًّا .	1
TbybAt	جمع مؤنث سالم	Female doctors	فُعَيْلَات	طَبِيبَات	الطَبِيبَات	اخْتَصَّتْ الطَّبِيبَات فِي عِرْقَةِ الإِجْتِمَاعَات .	2
mElmAt	جمع مؤنث سالم	Female teachers	مُعَلِّمَات	مُعَلِّمَات	المُعَلِّمَات	المُعَلِّمَات يُقَدِّمْنَ دُرَرَات تَرْبِيَّةَ عِزِّ الْإِنْتَرْنِت .	3
HsnAti	جمع مؤنث سالم	Good deeds	فُعَلَات	حَسَنَات	الحَسَنَات	إِنَّ الحَسَنَات يُلْهِيْنَ السَّيِّئَات .	4
fnAnAt	جمع مؤنث سالم	Female artists	فَعَالَات	فَنَات	الفَنَات	الفَنَاتُ الْفَنَاتُ فِي مَهْرَجَانِ الْفَنِ الْخَدِيثِ .	5
m&rxAt	جمع مؤنث سالم	Female historians	مُفَعِّلَات	مُؤَرِّخَات	المُؤَرِّخَات	المُؤَرِّخَات يَدْرُسْنَ تَارِيخَ الشُّعُوبِ الْقَدِيمَةِ .	6
mhndsAt	جمع مؤنث سالم	Female engineers	مُفَعِّلَات	مُهِنْدِسَات	المُهِنْدِسَات	عَمِلَتْ المُهِنْدِسَاتُ فِي تَصْمِيمِ الْجِسْرِ الْخَدِيدِ .	7
mtrjmAt	جمع مؤنث سالم	Female translators	مُفَعِّلَات	مُتَرَجِمَات	المُتَرَجِمَات	المُتَرَجِمَاتُ يَتَرَجِمْنَ الْمَقَالَات مِنَ الْإِنْجِلِيزِيَّةِ .	8
TAHyAt	جمع مؤنث سالم	Female chefs	فَاعِلَات	طَاهِيَات	الطَاهِيَات	الطَاهِيَاتُ يُخْضِرْنَ الْمَأْكُولَات لَوَجْهِ الْعَشَاءِ .	9
ryADyAt	جمع مؤنث سالم	Female athletes	فُعَالِيَات	رِيَاضِيَات	الرِّيَاضِيَات	الرِّيَاضِيَاتُ يَتَسَلَّقْنَ فِي السُّبُوطِ الدَّوْلِيَّةِ .	10
kAtbAt	جمع مؤنث سالم	Female writers	فَاعِلَات	كَاتِبَات	الكَاتِبَات	الكَاتِبَاتُ يَكْتُبْنَ عَنْ قِصَصِ النِّجَاحِ وَالْإِلْهَامِ .	11
mdrbAt	جمع مؤنث سالم	Female coaches	مُفَعِّلَات	مُتَرَبِّيات	المُتَرَبِّياتُ	المُتَرَبِّياتُ يُلْقِمْنَ تَصَالِيحَ قِيَمَةٍ لِلْأَعْيَانِ .	12
zhrAt	جمع مؤنث سالم	Flowers	فُعَلَات	زَهْرَات	الزُّهْرَاتُ	الزُّهْرَاتُ تَمَلُّ الْخَيْطَةَ بِزَوَاجِحِهَا الْجَبِلَةِ .	13
EaAzifAt	جمع مؤنث سالم	female musicians	فَاعِلَات	عَازِفَات	العَازِفَاتُ	العَازِفَاتُ يُقَدِّمْنَ الْحَقَائِقَ سَاحِرَاتٍ فِي الْحُلِيِّ الْمَوْسِقِيِّ .	14
mSmmAt	جمع مؤنث سالم	Female designers	مُفَعِّلَات	مُصَمِّمَات	المُصَمِّمَاتُ	المُصَمِّمَاتُ يَنْتَكِرْنَ تَصَاوِيمَ عَصْرِيَّةً وَخَدِيدَةً .	15
EdA'At	جمع مؤنث سالم	Female runners	فُعَالَات	عَوَّادَات	العَوَّادَاتُ	العَوَّادَاتُ يُسَارِكْنَ فِي سِبَاقِ الْمَارَاثُونِ .	16
bAHvAt	جمع مؤنث سالم	Female researchers	فَاعِلَات	بَاحِثَات	البَاحِثَاتُ	البَاحِثَاتُ يَنْحُدْنَ فِي أَسْوَاقِ خَدَدَةٍ مُعَمَّةٍ .	17

Figure 3.3 Sample visualization of annotated Sound Feminine Plural entries.

In total, **300 sound masculine plural sentences** and **310 sound feminine plural sentences** were added to the dataset. These additions enriched the dataset with over **600 new annotated examples**, representing real-world morphological patterns and enabling multi-class classification across plural types.

3.5 Significance of the Additions

The inclusion of sound plural forms was critical in training the model to differentiate broken plurals from other plural structures in Arabic. This addition improved classification accuracy and broadened the scope of the project by extending its applicability to more diverse linguistic phenomena.

This chapter detailed the full development of the dataset: from collecting broken plural examples, applying detailed linguistic

analysis, conducting morphological classification, and enriching the dataset with new grammatical features.

The process confirmed that high-quality input data is the backbone of any intelligent language model. Special focus was placed on accurate annotation of morphological patterns and plural categories, ultimately creating a rich, structured dataset ready for both training and evaluation.

The foundation laid in this chapter paves the way for building a model that can distinguish subtle linguistic differences across Arabic plural types—pushing forward the educational and research goals of this project.

In the next chapter, we explore the training pipeline and performance evaluation of the model using this curated and linguistically diverse dataset.

3.6 Relative Frequency of Morphological Weights in Broken Plurals

To assess the distribution of broken plural patterns in authentic Arabic usage and validate the representativeness of the constructed dataset, we conducted a relative frequency analysis using a real-world literary corpus. Since no standardized global reference exists for the frequency of Arabic broken plural patterns—unlike the well-documented distribution of letters or roots—this analysis provides a unique and original contribution to Arabic NLP.

As part of this effort, broken plural forms were manually extracted from the novel "Tis'ata 'Ashar" (تسعة عشر) by Ayman Al-Atoum approximately 300-page [60] a modern literary work characterized by rich and expressive Arabic usage. Each plural form was tagged with its corresponding morphological pattern (e.g. فَعَائِل, مَفَاعِل, فَعُول) and frequencies were calculated accordingly. These frequencies were then plotted graphically.

from book

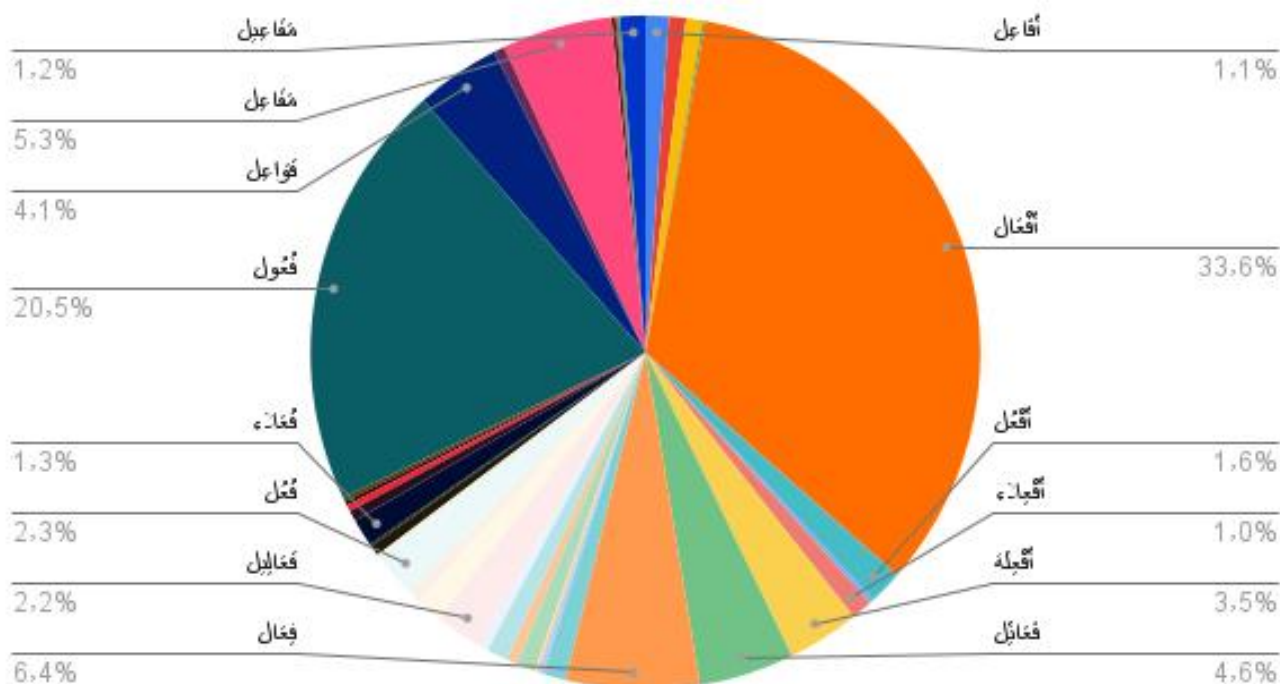


Figure 3.4: Relative frequency distribution of broken plural patterns in the novel (تسعة عشر)

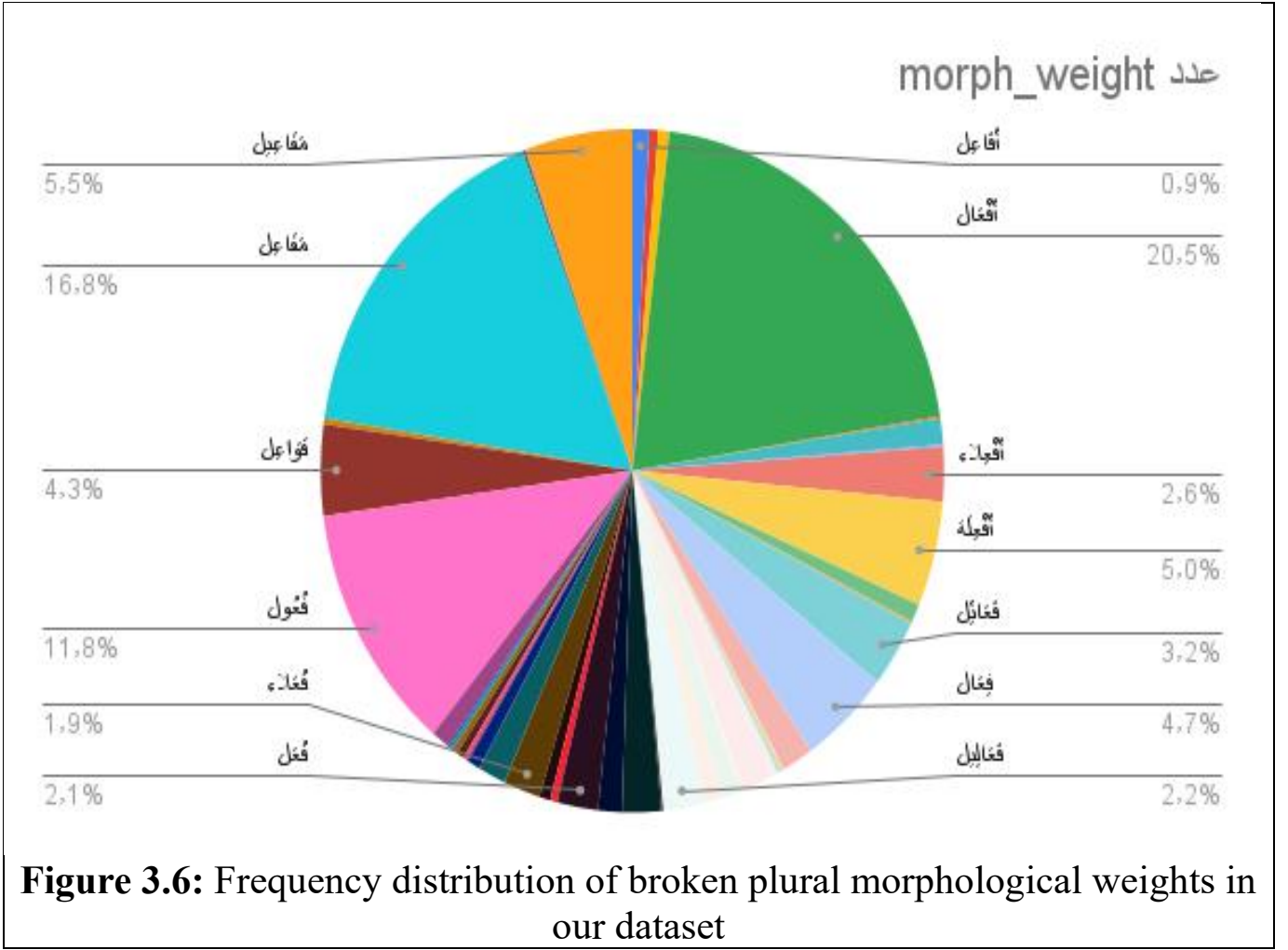
This chart reveals that "أَفْعَال" and "فُعُول" are the most dominant morphological weights, collectively covering more than half of the observed instances. This strong presence confirms their centrality in Arabic morphology and further validates their emphasis in our dataset.

To support this analysis, we include a snapshot of the manual extraction process that was used to label each word with its weight:

الكلمة	الوزن الصرفي
هياكل	فَعَائِل
هواجس	فَوَاعِل
هواجس	فَوَاعِل
نيران	فِعَال
نوافذ	فَوَاعِل
نوافذ	فَوَاعِل
نوافذ	فَوَاعِل
نوارس	فَوَاعِل
نواب	فَعَائِل
نواب	فَعَائِل
نواب	فَعَائِل
نمور	فُعُول
نمور	فُعُول
نَمِل	فَعِل
نُقَط	فُعِل
نقاط	فِعَال
نُقَاد	فُعَال
نفوس	فُعُول
نَعَم	فَعِل
نصوص	فُعُول
نصوص	فُعُول

Figure 3.5: Sample of manually extracted and labeled broken plurals from the novel

A comparative breakdown was also conducted between the literary-based frequency results and the internal dataset created in this study. The goal was to determine the extent to which our dataset mirrors natural usage frequencies.



These comparative visuals demonstrate a striking similarity between the distributions extracted from the novel and our curated dataset. Notably, patterns such as "أفعال" و"فُعول" و"مفعول" و"فُعلاء" consistently appear with high frequency across both sources, confirming that our data selection process accurately reflects actual Arabic usage patterns.

This section was intentionally placed here, following the detailed dataset construction and feature derivation steps, to serve as a bridge between raw data preparation and model training. By quantifying the real-world frequency of broken plural patterns, it reinforces the practical value of the prior sections and provides a statistical foundation for the modeling phase.

Morphological Pattern	Frequency in Dataset (%)	Frequency in Novel (%)
أفعال	20.5	33.6
فُعُول	11.8	20.5
مَفَاعِل	16.8	5.3
فَوَاعِل	4.3	4.1
فَعَال	6.4	2.2
فَعَالِل	2.2	2.3
فِعال	2.1	6.4
أَفْعِلَة	5.0	1.0
أَفْعَال	2.6	1.6
فُعَل	1.9	1.3
فُعَال	1.2	1.2
أَفَاعِل	0.9	1.1

Table 3.8: Comparison of Morphological Weight Frequencies between Dataset and Novel

Such analysis is highly beneficial in multiple ways:

- It allows us to focus model training on the most relevant and widely used morphological patterns.
- It helps avoid overfitting on rare or marginal forms.
- It provides an external benchmark to evaluate dataset completeness.

This empirical grounding reinforces the credibility of the dataset and ensures that the model developed will generalize effectively to real-world Arabic text. It also offers a baseline for future comparisons should larger corpora or genre-specific datasets be analyzed in subsequent studies.

CHAPTER 4: SYSTEM ARCHITECTURE AND IMPLEMENTATION

4.1 Tools and Technologies Used

To implement a robust system capable of analyzing Arabic broken plurals and delivering real-time predictions, a full-stack technology stack was employed. The selected tools cover the full data science pipeline: preprocessing, modeling, deployment, and user interaction.

Tool/Library	Type	Purpose / Description
Python	Programming Lang.	Core language used for all ML and NLP tasks [34]
Pandas, NumPy	Libraries	Data manipulation and array-based operations [34]
Scikit-learn	ML Library	Model training and evaluation (LogReg, SVM, RF, etc.) [34]
Matplotlib& Seaborn	Visualization	Enabled exploratory data analysis and visualization [34]
Flask	Web Framework	Exposes the model as an API [39]
React + Vite	Front-End Framework	Builds fast, modular interface [38]
Tailwind CSS	Styling Framework	Provides utility-first classes for responsive design [56]
Joblib	Serialization	Saves and loads the model and vectorizer efficiently [34]
Table 4.1: Summary of Tools and Libraries Used in the Project		

This technology stack ensured high modularity, maintainability, and the ability to deploy a fully functional and responsive NLP application.

4.2 Data Preparation Pipeline

4.2.1 Data Collection

The dataset was manually curated by the research team and includes a total of **1,702** Arabic sentences, composed of 1,047 broken plurals, 335 sound feminine plurals, and 320 sound masculine plurals. Each entry included the original sentence, extracted plural word, lemmatized form, morphological pattern, and plural type.

4.2.2 Data Cleaning and Normalization

Text was normalized by removing punctuation, numbers, diacritics, and noisy elements. Sentences lacking a valid broken plural were excluded. All words were fully diacriticized using Farasa, followed by manual review.

4.2.3 TF-IDF Feature Extraction

TF-IDF was applied at the character level using `char_wb` with n-gram ranges from 3 to 6. [52] This configuration captured internal morphological patterns effectively, outperforming word-level representations.

```
from sklearn.feature_extraction.text import TfidfVectorizer
vectorizer = TfidfVectorizer(analyzer='char_wb', ngram_range=(3, 6))
X_tfidf = vectorizer.fit_transform(corpus)
```

Figure 4.1: TF-IDF Vectorizer Configuration for Character-Level N-grams

4.2.4 Train-Test Split

The data was split into 80% training and 20% testing using stratified sampling to maintain the distribution of plural types.

4.3 Model Training and Evaluation

4.3.1 Models Implemented

Four classifiers were trained:

- Logistic Regression
- Multinomial Naive Bayes
- Support Vector Machine (SVM)
- Random Forest (final model)

4.3.2 Rationale for Selection

- These models are suited for high-dimensional, sparse data, typical of TF-IDF representations. Random Forest was ultimately chosen for its superior performance in both accuracy and interpretability [34].

4.3.3 Hyperparameter Tuning

Basic tuning was conducted:

- SVM: Explored different kernels and regularization values.
- Random Forest: Adjusted `n_estimators` and `max_depth`.

4.4 Model Serialization and Integration

4.4.1 Joblib Serialization

The final model and vectorizer were saved using Joblib for efficient reloading within the Flask environment [34]

```
import joblib
joblib.dump(model, "rf_final_model.pkl")
joblib.dump(vectorizer, "rf_final_vectorizer.pkl")
```

4.5 System Architecture

4.5.1 Use Case Overview

The system is designed for students, educators, and linguists who wish to input Arabic text and receive morphological classification in real-time.

4.5.2 Architecture Flow

- Frontend (React [56]+ Tailwind CSS [38]): Provides a user-friendly interface. Input is sent to the Flask API.
- Backend (Flask): Receives user input, applies TF-IDF transformation, and uses the Random Forest model for prediction [39].
- Model: Outputs the plural type (Broken, Sound Masculine, Sound Feminine) and confidence score.

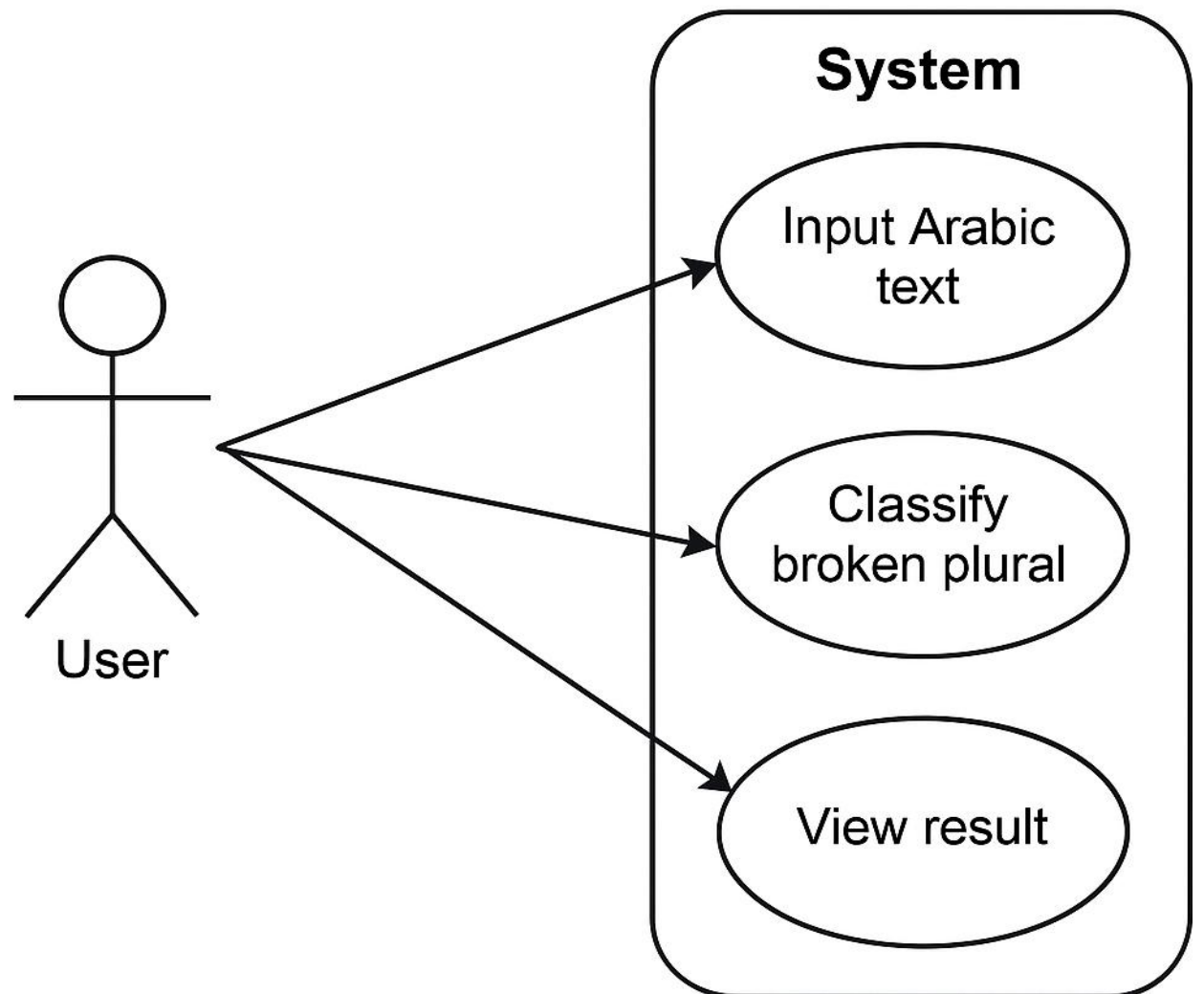
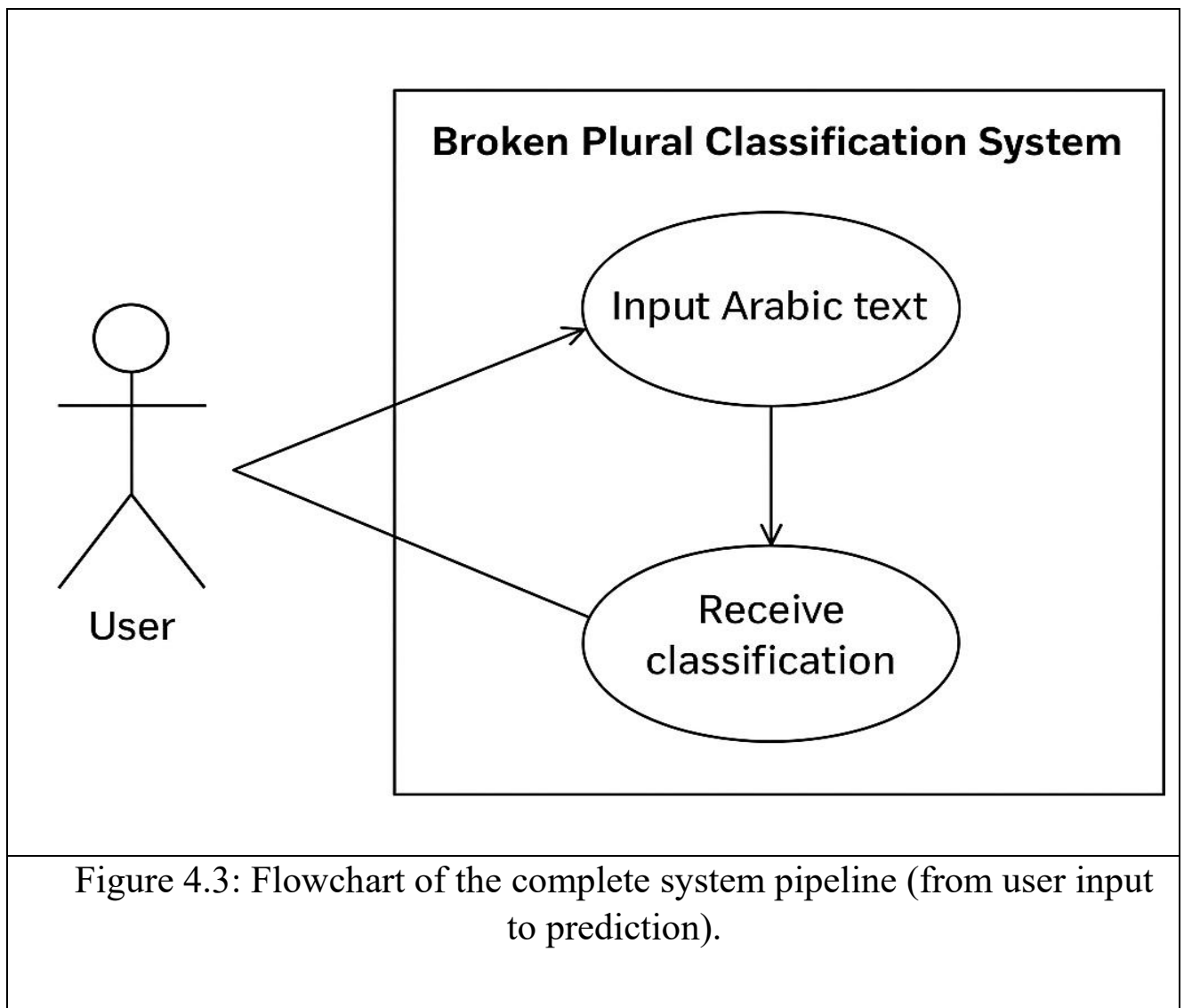


Figure 4.2: Use Case Diagram showing the interaction between user and system.



These visuals help reinforce understanding of how the data flows from interface to model and back.

4.6 Conclusion

This chapter demonstrated how a carefully selected set of tools, paired with linguistically-informed data processing and robust machine learning techniques, led to the construction of a practical and educationally valuable Arabic NLP system.

CHAPTER 5: RESULTS AND EVALUATION

5.1 Statistical Performance Metrics

To assess the performance of the implemented models, standard classification metrics were used: accuracy, precision, recall, and F1-score. The results from the test set are summarized below:

Model	Accuracy	Precision	Recall	F1-score
Logistic Regression	96.08%	96.12%	96.02%	96.06%
Naive Bayes	93.87%	94.12%	93.45%	93.78%
Support Vector Machine (SVM)	97.17%	97.18%	97.09%	97.13%
Random Forest (Final Model)	98.11%	98.21%	98.03%	98.12%

Table 5.1: *Comparison of the performance of machine learning models based on accuracy, precision, recall, and F1-score.*

These results show that Random Forest outperformed all other models, with particularly strong results in both precision and recall based on scikit-learn metrics module on the test set [34].

5.2 K-Fold Cross Validation

To ensure the model's robustness and mitigate overfitting, a 5-fold cross-validation was performed using the entire dataset [34]. The average F1 Macro score for the Random Forest model was 97.88%, indicating stable generalization across different subsets.

Fold #	F1-Score (%)
Fold 1	97.6%
Fold 2	98.2%
Fold 3	97.9%
Fold 4	97.7%
Fold 5	98.1%
Table 5.2: F1-Score Results Across 5-Fold Cross Validation Demonstrating Model Stability	

5.3 Evaluation on Unseen Data

The final model was evaluated on new, manually selected words not seen during training. It consistently predicted the correct plural types with high confidence (above 95%), matching human classification in nearly all cases.

5.4 Error Analysis

Despite the model's strong overall performance, some misclassifications were observed. Common causes included:

- **Rare or ambiguous plural forms** not well represented in the training data.
- **Morphologically irregular forms** that deviate significantly from known patterns.

These cases highlight the need for additional training data in edge cases and further integration of morphological rules[27].

While the model achieved strong overall accuracy, some words—such as "عناوين" (80%), "أساتذة" (62%), and "صلوات" (64%)—were classified correctly but with lower confidence scores. These borderline cases highlight the model's uncertainty with morphologically complex or less frequent patterns. Enhancing the

training data with additional examples of such structures could improve model calibration.

5.5 Visual Comparisons

To better understand performance differences, visual aids were created: These visuals highlight the dominance of Random Forest and the balanced variety of patterns present in the dataset.

5.6 Impact of Preprocessing Enhancements

The improvements applied during preprocessing were critical to model accuracy:

- Manual lemmatization avoided noise from automated tools.
- Normalization improved pattern recognition.
- TF-IDF character n-gram settings (3 to 6) captured morphological structures[52].
- Balancing the dataset improved generalization.

Without these steps, SVM and Naive Bayes showed signs of underfitting.

5.7 Summary

The combination of robust preprocessing, morphological insight, and algorithmic evaluation enabled the creation of a highly effective broken plural classification model. Random Forest was selected for deployment based on empirical superiority across metrics and its interpretability.

CHAPTER 6: DEPLOYMENT AND INTERFACE

6.1 Front-End Interface (React.js)

The user interface was built using React.js[54] with Vite and styled using Tailwind CSS[55]. The application offers:

- Manual input or pasted Arabic text.
- Dynamic visualization of classification results.
- Real-time display of plural types and confidence scores.

The interface is component-based, supporting scalability and reusability. All logic resides in dedicated components (e.g., PluralClassifierApp.jsx).

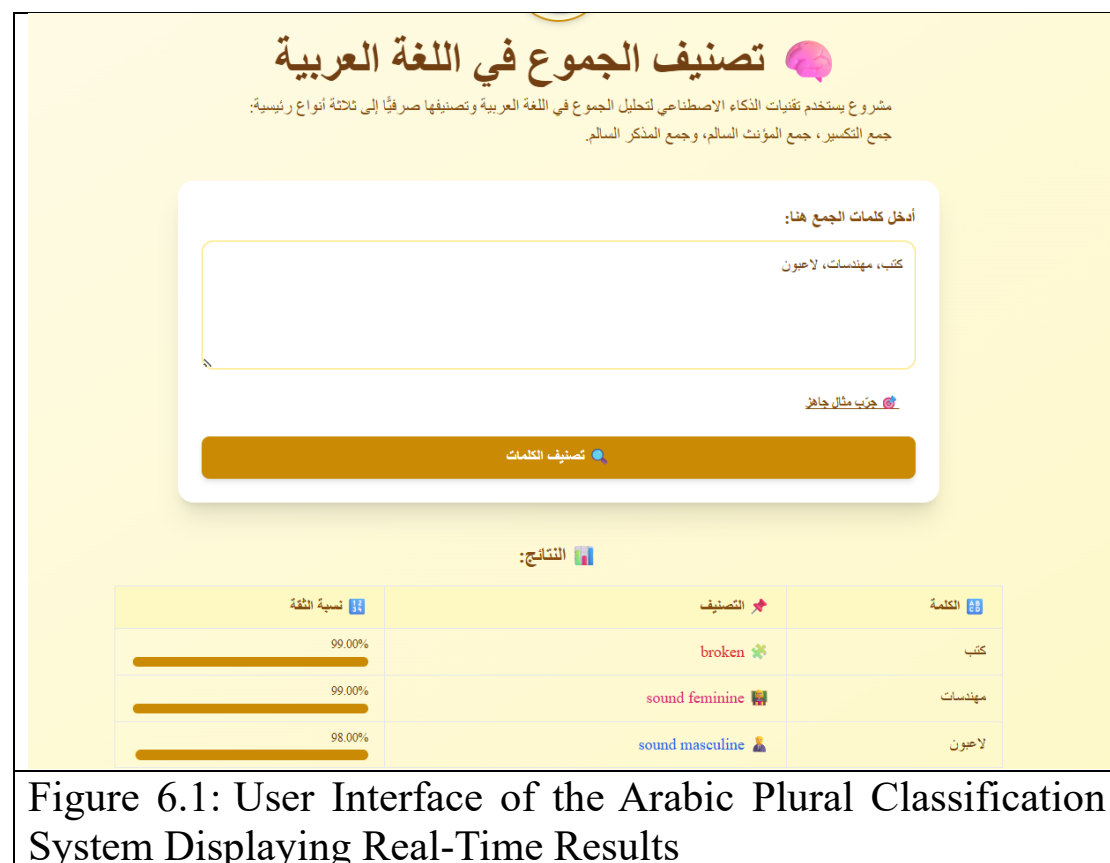


Figure 6.1: User Interface of the Arabic Plural Classification System Displaying Real-Time Results

6.2 Back-End and API Integration

A lightweight Flask server exposes a REST API[53] on the /api/classify endpoint. The API handles POST requests from the front-end, applies the saved model and TF-IDF vectorizer[34], and returns predictions in JSON format.

Data flow summary:

1. User submits Arabic text from the React form
2. Text is sent via HTTP POST to Flask backend
3. Backend transforms and classifies the input
4. Response with plural type and confidence is rendered in the React table.

6.3 Deployment Process

The system currently runs locally in a development environment, but is fully production-ready and compatible with popular cloud deployment platforms. However , it is fully prepared for production deployment on cloud platforms. The Flask back-end can be deployed using Render[57], Railway, or Heroku[56], each offering a free-tier option for lightweight applications. For the front-end, Vercel and Netlify[58] are suitable platforms due to their fast, continuous deployment from GitHub repositories.

6.4 GitHub Repository Structure

The entire project is structured and documented in a GitHub repository. Key directories include:

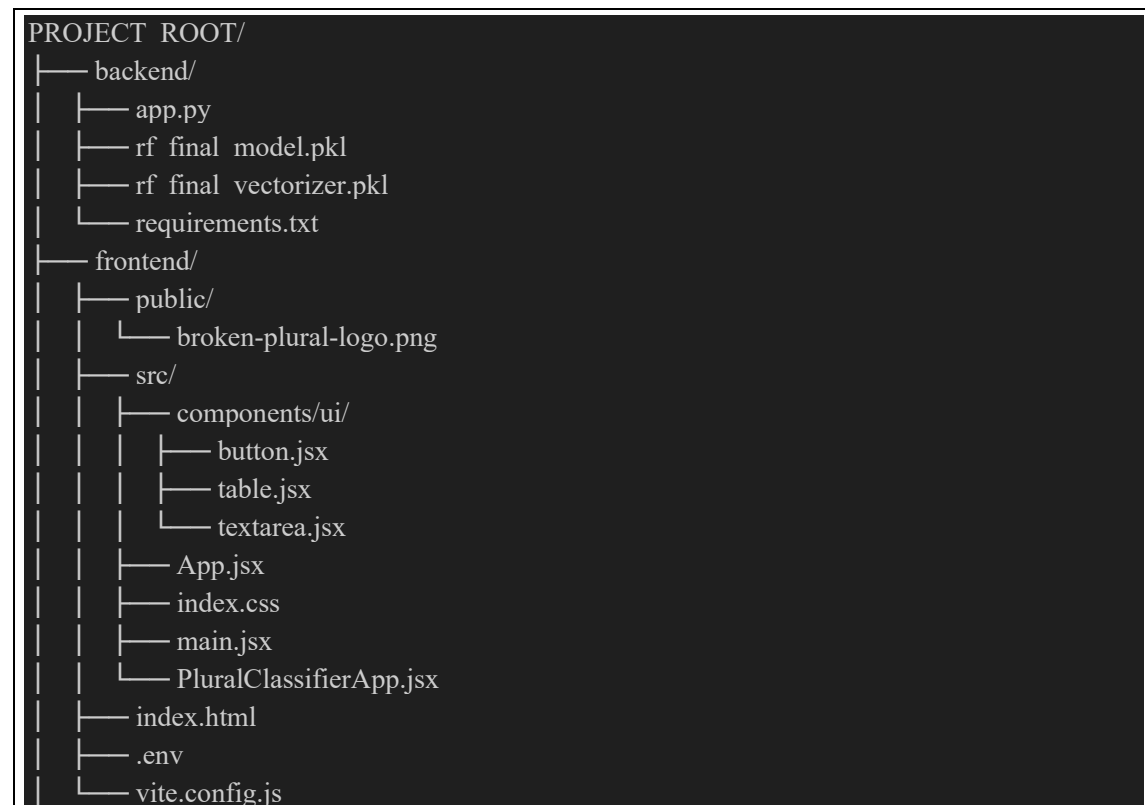


Figure6.2: Directory Structure of the Plural Classification System

A README.md file includes setup instructions, usage notes, and example screenshots.

6.5 Conclusion

This chapter detailed the deployment pipeline and interface design for the Arabic plural classification system. By combining an accessible front-end with a responsive back-end API, the system supports real-time interaction, educational utility, and future integration into larger Arabic NLP applications.

With the successful deployment of the word-level classification model, the system achieved high performance in recognizing and labeling Arabic plural forms. However, relying exclusively on

individual words introduces notable limitations in real-world text analysis, where context plays a crucial role.

To address these limitations and enhance the system’s linguistic depth, the next chapter extends the proposed solution by introducing a **sentence-level classification model**. This model analyzes full Arabic sentences to determine the presence of broken plurals in their contextual usage, thereby bridging the gap between isolated word processing and broader semantic interpretation.

CHAPTER 7: Sentence-Level Classification System for Arabic Broken Plurals

7.1 Introduction

While the word-level classification model described in earlier chapters effectively identifies Arabic broken plurals with high precision, it operates in isolation and lacks contextual awareness. This chapter presents the second part of the proposed system, focusing on sentence-level detection and classification of Arabic broken plurals. By analyzing complete sentences, the system enhances contextual understanding and captures plural forms more accurately in real-world text scenarios.

7.2 Tools and Technologies Used

To implement the sentence-level classification system, several technologies were employed across the pipeline, from data preprocessing to model deployment.

Tool/Library	Purpose
Python	Core language used throughout the project
Google Colab	Environment for training, testing, and evaluating models

Scikit-learn [60]	Model training, evaluation, and pipeline management
Joblib	Model and vectorizer serialization/deserialization
re (Regex)	Cleaning Arabic text: diacritics, articles, symbols
String Module	Punctuation removal and normalization
Pandas / NumPy	Data processing, manipulation, and analysis
Openpyxl	Reading data from Excel files
Stanza [61]	Tokenization and Part-of-Speech tagging for Arabic
NLTK	Arabic stopword removal
Streamlit [62]	Building the user interface
Matplotlib	Plotting performance and evaluation graphs
Random Forest	Word-level broken plural detection
Table 7.1: Tools and Libraries Used in Sentence-Level Classification	

7.3 Data Preparation Pipeline

The data used in this stage was extended from the corpus developed in Chapter 3.

1. **Data Collection:** Sentences were selected to include at least one plural form. Each entry included the sentence, plural type(s), and broken plural annotations.
2. **Cleaning & Normalization:** Diacritics and punctuation were removed; stopwords were filtered using NLTK. The

definite article "ال" was stripped to aid morphological pattern recognition.

Original Sentence	Cleaned Version
ذهب الطلاب إلى المدارس	ذهب طلاب مدارس

Table 7.2: Example of Sentence Cleaning

3. Labeling:

- 1: Sentence contains one or more broken plurals
- 0: No broken plurals

4. **TF-IDF Vectorization:** Combined word-level unigrams and character-level n-grams (3–5 char_wb).

5. **Train-Test Split:** Stratified sampling was used (80/20) to preserve class distribution.

7.4 Model Training and Evaluation

Four models were tested:

- Logistic Regression
- Naive Bayes
- Random Forest
- Support Vector Machine (SVM)

The SVM model with linear kernel and `class_weight='balanced'` achieved the best results and was selected.

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.74	0.79	0.75	0.71
Naive Bayes	0.76	0.79	0.75	0.71
Random Forest	0.74	0.75	0.73	0.73

Model	Accuracy	Precision	Recall	F1-Score
SVM (Selected)	0.81	0.82	0.81	0.81

Table 7.3: Comparison of Classifiers on Sentence-Level Data

7.4.1 K-Fold Validation

Stratified 5-fold cross-validation yielded an average accuracy of 90.4%.

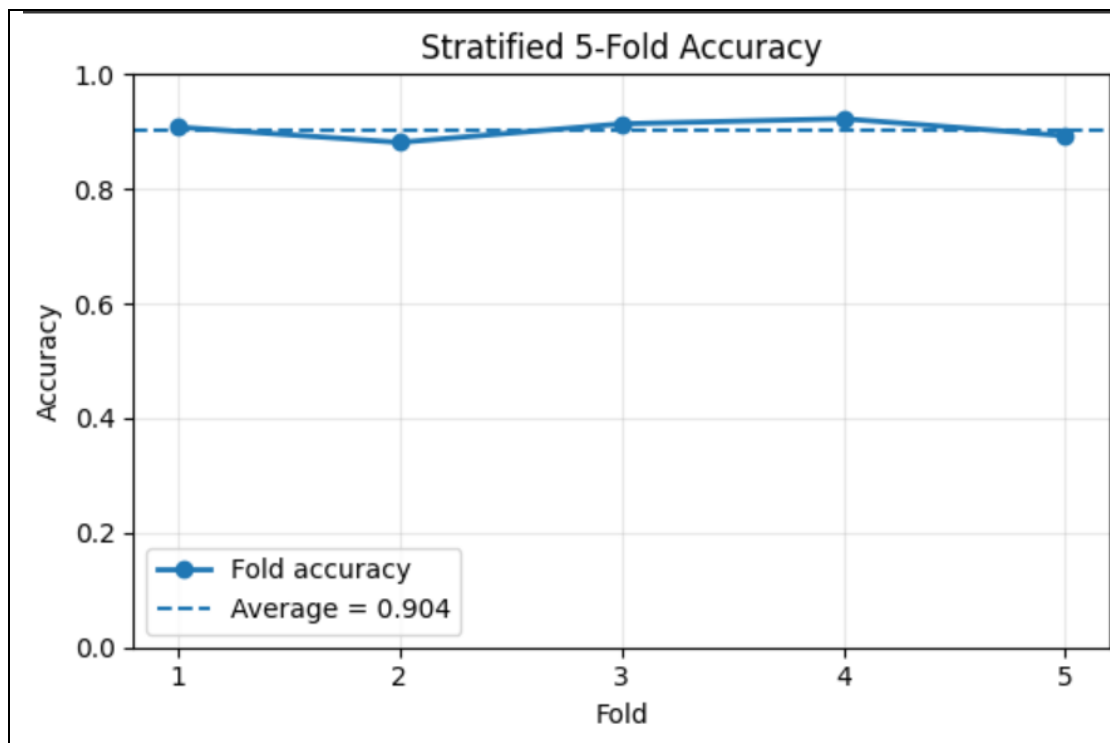


Figure 7.1: A line chart showing consistent accuracy values across all 5 folds of cross-validation.

7.4.2 Serialization

```
import joblib
joblib.dump(vectorizer, "finalmodelvectorizer.pkl")
joblib.dump(model, "finalmodel.pkl")
```

7.5 System Architecture and Use Case Flow

7.5.1 Use Case

Users such as students, teachers, or linguists input full Arabic sentences. The system identifies whether broken plurals exist and displays them interactively.

7.5.2 Data Flow

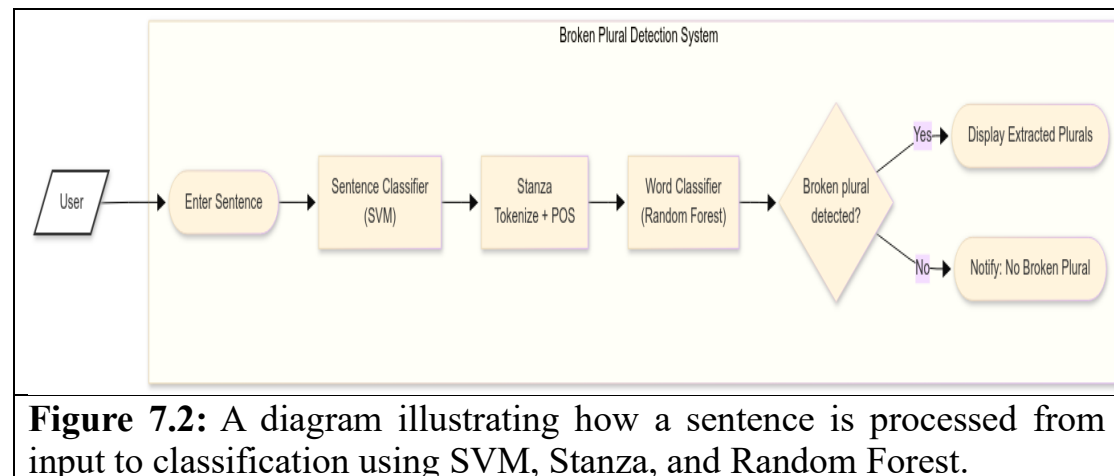


Figure 7.2: A diagram illustrating how a sentence is processed from input to classification using SVM, Stanza, and Random Forest.

Steps:

1. Input sentence via Streamlit
2. Clean and preprocess
3. Apply SVM classifier
4. Pass to Stanza for POS-tagging
5. Extract plural nouns/adjectives
6. Use Random Forest to confirm if broken
7. Display broken plurals or notify absence

7.6 Deployment and Interface

The sentence-level model was implemented in Streamlit, providing an all-in-one solution.

Front-End Features:

- Text box for sentence input
- "Auto-Try" for demo examples
- Reset button and sidebar instructions
- Styled feedback with color-coded results

Back-End Logic:

- Combined SVM (sentence) + RF (word) models
- Stanza used to filter only noun/adjective candidates
- Always runs word model for broader coverage

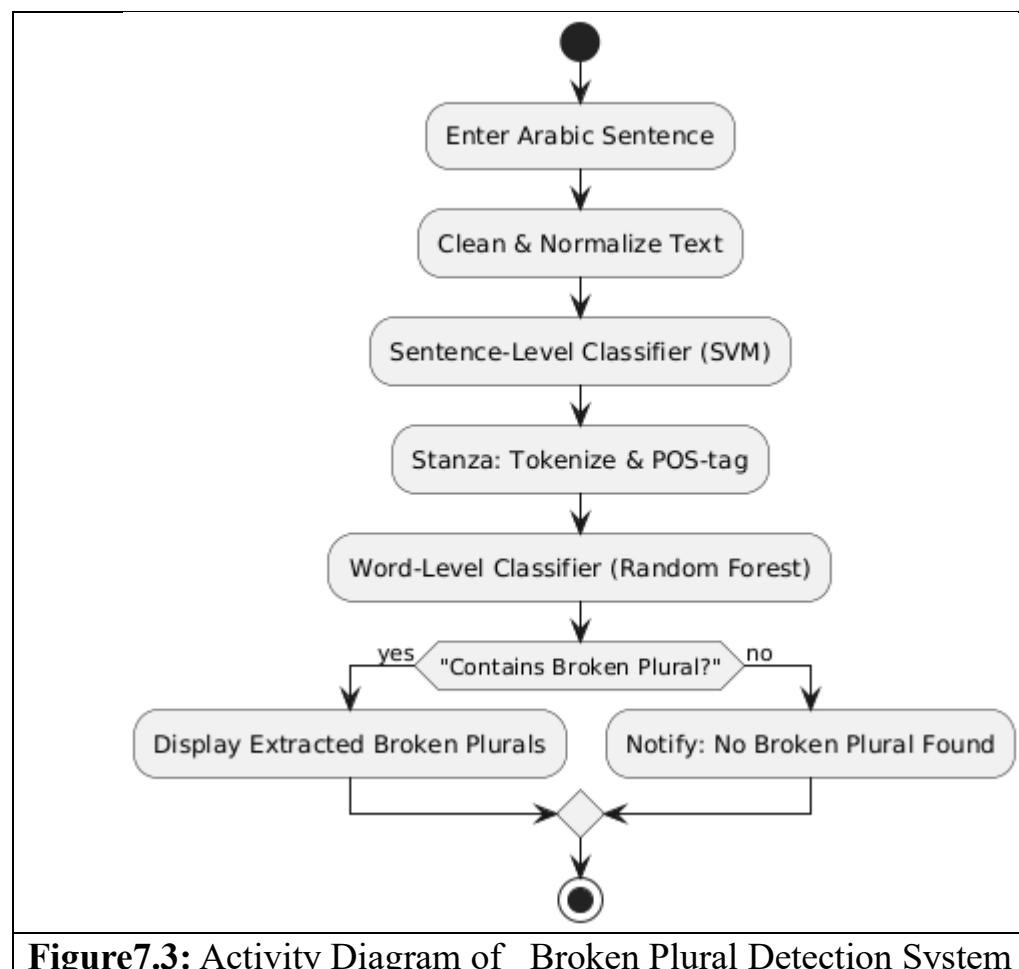


Figure7.3: Activity Diagram of Broken Plural Detection System

GitHub Repository

```
/ (root)
|-- haneen.py
|-- finalmodel.pkl
|-- finalmodelvectorizer.pkl
|-- rf_final_model.pkl
|-- rf_final_vectorizer.pkl
|-- requirements.txt
|-- logo.png
```

7.7 Summary

This chapter introduced a second interface and classification layer for Arabic broken plural detection. Unlike word-level classification, the sentence-based model leverages contextual cues and combines SVM + Random Forest in a two-stage design. It enhances the practicality of the system and supports real-world linguistic use cases through a modern, user-friendly Streamlit interface.

CHAPTER 8: CONCLUSION AND FUTURE WORK

8.1 Conclusion

Arabic broken plurals present a distinct challenge in computational linguistics due to their irregular and non-concatenative morphological structures [26]. Unlike sound plurals, which follow consistent suffix-based transformations, broken plurals require deeper linguistic analysis, making them more difficult to identify using standard machine learning approaches.

To address this challenge, this project developed a comprehensive system for the classification of Arabic plural forms, with a focus on broken plurals. A manually curated dataset of 1,701 annotated instances was used, consisting of 1,046 broken plurals, 335 sound feminine plurals, and 320 sound masculine plurals. Each instance was enriched with morphological patterns, lemma forms, diacritization, and sometimes translations.

The first part of the system implemented a word-level classifier using TF-IDF character-level n-grams and trained multiple models. The Random Forest model achieved the highest performance with an F1-score of 0.9976, and was integrated into a React + Flask web application. The second part extended the solution to sentence-level classification using an SVM model to identify sentences likely containing broken plurals. It then passed candidate plural words through the word-level classifier after part-of-speech tagging with Stanza. This sentence-level system was deployed using a Streamlit interface, providing real-time interaction.

Together, these components form a robust pipeline that bridges rule-based Arabic morphology and modern machine learning, enhancing accessibility for educators, researchers, and learners.

8.2 Future Work

While the system demonstrates high performance, several directions can be pursued to expand its capabilities:

1. **Deep Learning Integration**
Incorporating models like LSTM, GRU, or transformer-based architectures (e.g., AraBERT) could improve performance on complex or low-frequency patterns by capturing long-range dependencies [40][43].
2. **Context-Aware Classification**
Enhancing semantic understanding by using contextual embeddings such as BERT or FastText would improve disambiguation of plurals in context-rich sentences.
3. **Morphological Pattern Prediction**
Adding a module to predict morphological weights (e.g., fa‘āl, fu‘ūl) would be valuable for educational purposes and deeper linguistic analysis [27][29].
4. **Corpus Expansion**
Gathering texts from specialized domains such as medicine or law would enhance the model's generalization and domain adaptation.
5. **Educational Features**
Creating interactive tools such as grammar quizzes, plural-suggestion hints, or singular-to-plural converters could support Arabic language instruction.
6. **Cross-Language Research**
Comparative morphological studies with other Semitic languages (e.g., Hebrew, Amharic) and morphologically rich languages (e.g., Turkish, Persian) could enable cross-lingual applications and transfer learning [26].

7. Mobile and API Deployment

Packaging the system as a mobile app or open RESTful API would enable easy integration with other platforms, such as e-learning tools or Arabic grammar correction services.

8.3 Final Remarks

This project lays a strong foundation for advancing Arabic NLP with a focus on broken plural detection. By integrating machine learning, morphological preprocessing, and user-friendly interfaces, it provides a complete and scalable solution. The dual-level system enriches the possibilities for educational, linguistic, and technological applications, and paves the way for further innovation in Arabic language understanding.

CHAPTER 8: REFERENCES

- 1) [الراجحي.pdf](#)
- 2) معجم لسان العرب ابن منظور
<https://play.google.com/store/apps/details?id=com.smartbee991.app516805653354&hl=ar>
- 3) تطبيق الميزان الصرفي
<https://play.google.com/store/apps/details?hl=ar&id=com.halsys.searchconjugation>
- 4) جمع الكثرة في شعر المتنبي من خلال شرح الفسر
<http://mohamedrabeea.net/library/pdf/60da3f6e-bc48-4224-95a2-0eff5b39dff4.pdf>
- 5) <file:///C:/Users/PC%20LAND/Downloads/jma-altksir-jma-alqla-ojma-alkthra.pdf>
- 6) كتاب جمع التكسير وعلاقته بأسماء الجموع الأخرى
https://ia801307.us.archive.org/27/items/lis_m114/lisanarb.com_m1406.pdf
- 7) file:///C:/Users/PC%20LAND/Downloads/QARTS_Volume%2030_Issue%2052_Pages%20125-146.pdf
- 8) <file:///C:/Users/PC%20LAND/Downloads/%D8%A8%D9%88%D8%AF%D9%8A%D8%A9+%D8%AD%D8%AC%D8%A7%D8%AC.pdf>
- 9) <https://archive.org/details/4231pdf>
- 10) <https://www.twinkl.com/eg/blog/jm-altksyr>
- 11) https://www.almaany.com/#google_vignette
- 12) https://www.alukah.net/literature_language/0/7728/%D9%85%D8%AD%D8%B1%D9%83%D8%A7%D8%AA-%D8%A7%D9%84%D8%A8%D8%AD%D8%AB-%D9%81%D9%8A-%D8%A7%D9%84%D9%86%D8%B5%D9%88%D8%B5-%D8%A7%D9%84%D8%B9%D8%B1%D8%A8%D9%8A%D8%A9-%D9%88%D8%B5%D9%81%D8%AD%D8%A7%D8%AA-
- 13) https://mawdoo3.com/%D8%AE%D8%A7%D8%B5:%D8%A8%D8%AD%D8%AB_%D9%85%D9%88%D8%B6%D9%88%D8%B9?q=%D8%AC%D9%85%D8%B9+%D8%A7%D9%84%D8%AA%D9%83%D8%B3%D9%8A%D8%B1
- 14) <https://www.twinkl.com/teaching-wiki/jm-altksyr>

- 15) <https://mnahel.com/ar/home/article/%D9%85%D8%A7-%D9%87%D9%88-%D8%AC%D9%85%D8%B9-%D8%A7%D9%84%D8%AA%D9%83%D8%B3%D9%8A%D8%B1>
- 16) <https://www.arabici.com/2024/09/Crushing-plural.html>
- 17) <https://maqall.net/education/researches-scientific/collect-cracker/>
- 18) <https://3shal3arabia.com/%D9%85%D8%A7-%D9%87%D9%88-%D8%AC%D9%85%D8%B9-%D8%A7%D9%84%D8%AA%D9%83%D8%B3%D9%8A%D8%B1/>
- 19) <https://m3rfah.com/%D8%A8%D8%AD%D8%AB-%D8%B9%D9%86-%D8%AC%D9%85%D8%B9-%D8%A7%D9%84%D8%AA%D9%83%D8%B3%D9%8A%D8%B1-%D9%81%D9%8A-%D8%A7%D9%84%D9%84%D8%BA%D8%A9-%D8%A7%D9%84%D8%B9%D8%B1%D8%A8%D9%8A%D8%A9/>
- 20) <https://loghate.com/s/%D8%AC%D9%85%D8%B9-%D8%A7%D9%84%D8%AA%D9%83%D8%B3%D9%8A%D8%B1-%D8%AA%D8%B9%D8%B1%D9%8A%D9%81%D9%87-%D9%88%D8%A3%D9%82%D8%B3%D8%A7%D9%85%D9%87-%D9%88%D8%A5%D8%B9%D8%B1%D8%A7%D8%A8%D9%87>
- 21) https://mawdoo3.com/%D8%A3%D9%85%D8%AB%D9%84%D8%A9_%D9%88%D8%AA%D8%AF%D8%B1%D9%8A%D8%A8%D8%A7%D8%AA_%D8%B9%D9%84%D9%89_%D8%AC%D9%85%D8%B9_%D8%A7%D9%84%D8%AA%D9%83%D8%B3%D9%8A%D8%B1
- 22) <https://www.edarabia.com/ar/%D8%AC%D9%85%D8%B9-%D8%A7%D9%84%D8%AA%D9%83%D8%B3%D9%8A%D8%B1-%D9%88%D9%83%D9%8A%D9%81%D9%8A%D8%A9-%D8%B5%D9%8A%D8%A7%D8%BA%D8%AA%D9%87-%D9%88%D8%A3%D9%88%D8%B2%D8%A7%D9%86%D9%87-%D9%88%D8%A7%D8%B9/>
- 23) <https://dorar.net/arabia/1163/%D8%A7%D9%84%D8%A8%D8%A7%D8%A8-%D8%A7%D9%84%D8%AB%D8%A7%D9%85%D9%86-%D8%AC%D9%85%D8%B9-%D8%A7%D9%84%D8%AA%D9%83%D8%B3%D9%8A%D8%B1>
- 24) <https://www.mosoah.com/books-and-literature/guides-and-reviews/%D8%AC%D9%85%D8%B9-%D8%AA%D9%83%D8%B3%D9%8A%D8%B1/>
- 25) <https://www.modrsbook.com/2017/03/nahoo2.html>
- 26) Habash, Nizar. *Introduction to Arabic Natural Language Processing*. Morgan & Claypool Publishers, 2010.

- 27) Darwish, Kareem. "Building a shallow morphological analyzer in one day." *Proceedings of the ACL-02 workshop on Computational approaches to Semitic languages*, 2002.
- 28) Al-Taani, Ahmad T., and Mohammed M. Wahsheh. "Automatic text classification for Arabic language." *European Journal of Scientific Research* 40.4 (2010): 538-546.
- 29) Al-Hashimi, A. *Shadhā al- 'Urf fī Fan al-Ṣarf*.
- 30) Arabic Language Academy. *Al-Mu 'jam al-Wasīṭ*.
- 31) Haywood, J. A., & Nahmad, H. M. (1965). *A New Arabic Grammar of the Written Language*. London: Lund Humphries.
- 32) Farasa Arabic NLP Toolkit. <https://farasa.qcri.org/>
- 33) Haykin, Simon. *Neural Networks and Learning Machines*, 3rd Edition. Pearson, 2009.
- 34) Pedregosa, F., et al. "Scikit-learn: Machine learning in Python." *Journal of Machine Learning Research* 12 (2011): 2825-2830.
- 35) Simeone, Osvaldo. *A Brief Introduction to Machine Learning for Engineers*. Cambridge University Press, 2018.
- 36) Palani, S. *Signals and Systems*. Springer, 2021.
- 37) Scikit-learn documentation: <https://scikit-learn.org/stable/>
- 38) React Documentation: <https://reactjs.org/>
- 39) Flask Documentation: <https://flask.palletsprojects.com/>
- 40) Antoun, W., Baly, F., & Hajj, H. (2020). AraBERT: Transformer-based model for Arabic language understanding. *Proceedings of the LREC 2020 Workshop on Language Resources and Evaluation Conference*.
- 41) Buckwalter, T. (2004). Buckwalter Arabic Morphological Analyzer Version 2.0. *Linguistic Data Consortium*.
- 42) Darwish, K., Mubarak, H., & Abdelali, A. (2014). Using named entities to improve diacritization in Arabic. *Proceedings of the EMNLP*.
- 43) Nagoudi, E. M. B., Abdul-Mageed, M., & Bouamor, D. (2022). AraT5: Text-to-text transformer for Arabic language understanding and generation. *arXiv preprint arXiv:2202.01250*.
- 44) Obeid, O., et al. (2020). CAMEL Tools: An open-source Python toolkit for Arabic natural language processing. *Proceedings of LREC*.

- 45) Pasha, A., et al. (2014). MADAMIRA: A Fast, Comprehensive Tool for Morphological Analysis and Disambiguation of Arabic. *LREC 2014*.
- 46) المعجم الوسيط. مجمع اللغة العربية.
- 47) Haywood, J. A., & Nahmad, H. M. (1965). *A New Arabic Grammar of the Written Language*. London: Lund Humphries.
- 48) تحليل أخطاء الجمع في العربية في ضوء اللسانيات (2017). درقاوي، مختار. مجلة العلوم الإنسانية والاجتماعية، جامعة الجلفة /التطبيقية.
- 49) Al-Mawrid Arabic-English Dictionary. Dar El-Ilm Lilmalayin.
- 50) Almaany Multilingual Dictionary. <https://www.almaany.com/>
- 51) PyArabic Library: <https://github.com/linuxscout/pyarabic>
- 52) Ramos, J. (2003). *Using TF-IDF to determine word relevance in document queries*.
- 53) Flask Documentation – <https://flask.palletsprojects.com>
- 54) React Documentation – <https://reactjs.org>
- 55) Tailwind CSS Documentation – <https://tailwindcss.com>
- 56) Heroku Developer Docs – <https://devcenter.heroku.com>
- 57) Render Documentation – <https://render.com/docs>
- 58) Netlify Documentation – <https://docs.netlify.com>
- 59) كتاب "تسعة عشر" <https://www.rwaiaty.com/%D8%B1%D9%88%D8%A7%D9%8A%D8%A9-%D8%AA%D8%B3%D8%B9%D8%A9-%D8%B9%D8%B4%D8%B1/>
- 60) Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.
- 61) Qi, P., Zhang, Y., Zhang, Y., Bolton, J., & Manning, C. D. (2020). Stanza: A Python natural language processing toolkit for many human languages. arXiv preprint arXiv:2003.07082.
- 62) Streamlit Inc. (2020). Streamlit: The fastest way to build and share data apps. Retrieved from <https://streamlit.io>

CHAPTER 9: APPENDIX

Appendix A: Code Implementation Details

This appendix provides code-level documentation for building the Arabic plural classification system. The implementation includes preprocessing, feature engineering, model training, evaluation, and deployment using scikit-learn and Python

A.1 Data Preprocessing and Loading

```
import pandas as pd
from sklearn.model_selection import train_test_split
# Load CSV
df = pd.read_csv('data for training.csv')
df.columns = df.columns.str.strip()
# Basic filtering
stopwords = ["إلى", "من", "عن", "على", "في", "ب", "ك", "ل", "حتى", "ثم"]
df = df[df['word'].str.len() > 2]
df = df[~df['word'].isin(stopwords)]
# Normalize labels
df['label'] = df['label'].str.strip().str.lower()
df['label'] = df['label'].replace({
    'sound masculine': 'sound_masculine',
    'sound feminine': 'sound_feminine'
})
```

This step prepares the dataset by removing short/irrelevant words and normalizing class labels.

A.2 TF-IDF Feature Extraction

```
from sklearn.feature_extraction.text import TfidfVectorizer

X = df['word']
y = df['label']

vectorizer = TfidfVectorizer()
```

```
        analyzer='char_wb',
        ngram_range=(2, 6),
        min_df=2
    )
X_tfidf = vectorizer.fit_transform(X)
```

The TF-IDF vectorizer extracts character-level n-grams to effectively capture Arabic morphological patterns.

A.3 Model Training and Cross-Validation

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import StratifiedKFold, cross_val_score

model = RandomForestClassifier(n_estimators=100,
                              class_weight='balanced', random_state=42)
skf = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)
scores = cross_val_score(model, X_tfidf, y, cv=skf, scoring='f1_macro')

print("F1 Macro Scores:", scores)
print("Average F1:", scores.mean())
```

The model is trained and validated across 5 folds to ensure generalization and robustness.

A.4 Final Model Fit and Evaluation

```
from sklearn.metrics import classification_report, accuracy_score

X_train, X_test, y_train, y_test = train_test_split(X, y, stratify=y,
                                                    test_size=0.2, random_state=42)
X_train_tfidf = vectorizer.transform(X_train)
X_test_tfidf = vectorizer.transform(X_test)

model.fit(X_train_tfidf, y_train)
y_pred = model.predict(X_test_tfidf)

print("Accuracy:", accuracy_score(y_test, y_pred))
```

The model is evaluated on unseen test data using standard

classification metrics.

A.5 Model Saving and Inference

```
import joblib

# Save
joblib.dump(model, "rf_final_model.pkl")
joblib.dump(vectorizer, "rf_final_vectorizer.pkl")

# Load and Predict
model = joblib.load("rf_final_model.pkl")
vectorizer = joblib.load("rf_final_vectorizer.pkl")

def predict(word):
    X_vec = vectorizer.transform([word])
    pred = model.predict(X_vec)[0]
    prob = model.predict_proba(X_vec).max() * 100
    return pred, round(prob, 2)
```

This function allows the model to be deployed for real-time use in web or mobile applications

All code was tested using Python 3.10 and scikit-learn 1.2.2, ensuring full compatibility and reproducibility across environments. This implementation aligns with the architecture described in Chapter 4 and supports the real-time functionality outlined in Chapter 6.

Appendix B: Streamlit Sentence-Level Interface Details

This appendix describes the full implementation of the sentence-level classification interface developed using Streamlit.

B.1 Overview

The Streamlit app integrates a two-stage back-end model to analyze Arabic sentences:

1. Sentence-level SVM classifier detects whether the sentence contains any broken plural.
2. Word-level Random Forest classifier, assisted by Stanza POS tagging, identifies the specific broken plural words.

The interface is minimal and responsive, supporting both desktop and mobile devices.

B.2 File Structure

The following is the file layout for the sentence-level classification interface implemented in Streamlit:

```
/ (root)
├─ haneen.py                # Main Streamlit app file
├─ finalmodel.pkl           # Trained SVM model for sentence-level
prediction
├─ finalmodelvectorizer.pkl # Vectorizer for sentence TF-IDF
transformation
├─ rf_final_model.pkl       # Random Forest model for word-level
classification
├─ rf_final_vectorizer.pkl  # Vectorizer for word-level
classification
├─ requirements.txt         # Dependencies for deployment
├─ logo.png                 # Optional UI branding image
```

B.3 Running the App Locally

To run the application locally:

```
streamlit run haneen.py
```

This will start the interface in your browser at localhost:8501.

B.4 User Interface Design

- **Theme:** white background, orange buttons, dark sidebar
- **Branding:** centered circular logo with border
- **Mobile-friendly:** adjusts layout on screens $\leq 600\text{px}$
- **Features:**
 - Input box for Arabic sentence
 - Analyze, Reset, and Demo buttons
 - Real-time display of extracted broken plurals

B.5 Analysis Flow

```
import stanza
nlp = stanza.Pipeline(lang='ar', processors='tokenize,mwt,pos')

def clean_text(text):
    # Removes diacritics and punctuation
    return re.sub(r"[\u0617-\u061A\u064B-\u0652]", '',
text.translate(str.maketrans('', '', string.punctuation))).strip()


def analyze(text):
    cleaned = clean_text(text)
    prediction =
sentence_model.predict(sentence_vectorizer.transform([cleaned]))
    doc = nlp(cleaned)
    candidates = [w.text for s in doc.sentences for w in s.words if
w.upos in ["NOUN", "ADJ"] and "Number=Plur" in (w.feats or '')]
    result = word_model.predict(word_vectorizer.transform(candidates))
    return [w for w, p in zip(candidates, result) if p == 1 or p ==
'broken']
```

B.6 Deployment Option

For online access, deploy via [Streamlit Cloud](#) by linking the GitHub repository. No additional backend setup is require.

تعليمات:

1. أدخل جملة باللغة العربية 🇸🇦 في الصندوق الأبيض
2. اضغط على زر "تحليل" 🔍 ليتم تحليل الجملة واستخراج جمع التفسير إن وجد
3. اضغط "إعادة التعيين" 🔄 للبدء من جديد
4. اضغط "تجربة تلقائية" ⚡ لوضع جملة تجريبية



Broken Plural Extraction Model

نموذج ذكاء صناعي يحلل الجمل العربية لاستخراج جمع التفسير إن وجد

مثال: ذهب المعلمون إلى المدارس 🏫

أدخل جملة هنا

تحليل 🔍
تجربة تلقائية ⚡
إعادة التعيين 🔄

Share ★ 🔍 🔄 ⋮

Palestine Technical University - Kadoorie (PTUK) - Computer Systems Engineering Department

< Manage app

رابط المشروع

<https://github.com/raneen-jubahi/gradu-project>