Research Proposal

Discovering Hidden Tashbeeh in Arabic Poetry Using NLP and Deep Learning

Faisal Omari – 325616894

Assoc. Prof. Tsvi Kuflik, The Department of Information Systems

Dr. Ali Hussein, Department of Arabic Language and Literature

Department of Computer Science

May 2025

TABLE OF CONTENTS

1. Introduction

Arabic poetry is one of the richest literary traditions in the world, deeply intertwined with the culture, language, and identity of the Arab people. Within this poetic heritage, tashbīh (تشبيه), or simile, stands out as a central rhetorical device, used to create vivid imagery, evoke emotions, and clarify abstract concepts. Traditionally, tashbīh is expressed through explicit linguistic markers such as “كأن” (as if), “مثل” (like), or “كما” (as). For instance, the poet may write: "كأنّ سيوفهم نجومٌ في الليلة الظلماء" "As if their swords were stars in the dark night."

In this line, the simile is clearly marked by the word “كأنّ” and draws a comparison between swords and stars, emphasizing their brilliance and sharpness.

However, beyond these overt constructions lies a subtler and more complex form of rhetorical expression—hidden tashbīh (التشبيه الخفي). These similes lack explicit comparison words yet maintain the poetic function of analogy. For example, a poet may say:

"تلمع سيوفهم في الليل" "Their swords gleam in the night."

While the line does not include a formal simile word, the imagery implicitly evokes the same comparison to stars. These hidden similes are more difficult to detect, both for readers and for computational methods, as they require semantic and contextual understanding rather than keyword spotting.

Previous work on Arabic rhetoric, including research conducted on the Arabic Poetry Corpus developed at the University of Haifa, has focused on detecting explicit tashbīh using rule-based approaches—typically via handcrafted if-else conditions that look for tashbīh markers in poems. This method, while useful, is limited to only surface-level structures and fails to capture deeper, implicit rhetorical meanings.

This thesis aims to take a significant step forward by addressing the detection of hidden tashbīh using artificial intelligence. We propose to build machine learning models—specifically natural language processing (NLP) models fine-tuned on large pre-

trained Arabic transformers such as AraBERT—that are capable of learning what makes a poem "contain tashbīh," even when the tashbīh is not explicitly marked.

In the initial phase of this research, we have:

- Converted the legacy Java-based rule system to Python for scalability and integration with modern NLP pipelines;
- Used the existing rule-based output to generate a binary labeled dataset of poems with and without tashbīh;
- Preprocessed the poems by removing diacritics and cleaning the text;
- Fine-tuned the AraBERT model on this dataset, achieving a 96% accuracy in identifying tashbīh poems.

This strong result demonstrates the feasibility of teaching models to "understand" the rhetorical role of tashbīh in classical Arabic poetry.

The next challenge, and the core focus of this thesis, is to move from detecting obvious tashbīh to identifying hidden tashbīh: poetic lines where the comparison is embedded within context and meaning rather than structure. Our long-term goal is to build a model capable of identifying rhetorical richness in Arabic poetry even when surface cues are missing. To this end, we plan to:

- Manually transform a sample of explicit tashbīh poems into hidden ones;
- Investigate what features distinguish hidden tashbīh from non-tashbīh lines;
- Train and evaluate models that can generalize to these deeper forms of rhetorical expression.

Ultimately, this research contributes to the field of digital Arabic literary analysis by combining classical rhetorical theory with state-of-the-art AI techniques, opening new avenues for understanding and interpreting the depth of Arabic poetic language.

2. Background

Arabic rhetoric (balāgha) treats tashbīh (simile) as a foundational stylistic device. It is used to compare two entities by highlighting a shared attribute. The components of a classical simile are:

1) mushabbah (the object being compared)

2) mushabbah bihī (the object to which it is compared)

3) wajh al-shabah (the point of similarity)

4) the comparison particle (e.g., كأن، مثل، كما) [1].

• Explicit similes (tashbīh muṣarraḥ) state all four components clearly, e.g., 'Zayd is like a lion in bravery' [1].

• Implicit similes (tashbīh ḍimnī) omit the comparison particle and rely on context to imply similarity. For example, a poet might say: 'He is a lion in battle', implying courage without the word 'like' [2].

• Classical rhetoricians such as al-Jurjānī in 'Asrār al-Balāgha' and lexicographers such as al-Zubaidī in 'Tāj al-ʿArūs' elaborated on various types of simile and their rhetorical effects [3][4].

Related Work

• The Arabic Rhetoric Identifier (REI), developed at the University of Haifa, is a rule-based system for detecting rhetorical devices in Arabic poetry, including tashbīh. It uses explicit pattern-matching rules combined with NLP tools [5].

• The system operates on a large Arabic Poetry Corpus containing over 26,000 poems spanning 10 centuries. Only a subset of this corpus has been manually annotated due to the complexity of rhetorical analysis [6].

• Tools such as AlKhalil Morpho Sys 2 and MADAMIRA are integrated for morphological analysis and tokenization, which help detect comparative particles and relevant linguistic structures [7][8].

• Limitations: Rule-based approaches struggle with detecting implicit tashbīh due to their reliance on explicit markers. They require constant rule updates and are unable to generalize well to creative or indirect comparisons [5][9].

Advances in NLP for Arabic

• Recent Arabic NLP has benefited from large-scale pre-trained models such as AraBERT. These models can capture contextual nuances and semantic relationships, offering promise for detecting hidden similes [10].

• In English NLP, similar challenges have been addressed with transformer-based models for metaphor detection, such as DeepMet and MelBERT. These methods inspire the application of similar architectures to Arabic poetic text [11][12].

Detailed Overview of Arabic NLP Tools for Morphological Analysis:

• AlKhalil Morpho Sys 2: This is a robust Arabic morphological analyzer developed to handle both vocalized and unvocalized text. It uses a large internal lexicon and a set of rules to provide information such as the root, pattern, POS tags, and affixes of Arabic words. Pros: High accuracy on Classical Arabic; extensive lexicon coverage. Cons: Performance may degrade with dialectal or modern informal texts [7].

• MADAMIRA: An evolution of MADA and AMIRA, MADAMIRA is a Java-based tool offering tokenization, POS tagging, lemmatization, and diacritization. It is optimized for Modern Standard Arabic and provides a user-friendly interface with both batch and web modes. Pros: Fast, comprehensive, and widely used in academic NLP projects. Cons: Less effective for historical or poetic texts where language is highly figurative or archaic [8].

• BAMA (Buckwalter Arabic Morphological Analyzer): This was one of the earliest tools for analyzing Arabic morphology. It provides possible morphological analyses for Arabic words based on a large dictionary of stems and affixes. Pros: Fundamental and influential in early Arabic NLP research. Cons: Generates many ambiguous outputs and lacks disambiguation capabilities; limited support for contextual inference [13].

• SAMA (Standard Arabic Morphological Analyzer): An improved version of BAMA developed by the Linguistic Data Consortium. It refines and expands the analysis with updated lexicons and additional grammatical coverage. Pros: Better performance than BAMA and integrated in many academic projects. Cons: Still suffers from ambiguity and requires external tools for disambiguation [14].

• Farasa: A fast and efficient Arabic segmentation and POS tagging tool designed especially for processing large-scale data quickly. Pros: Speed and scalability; good for real-time applications. Cons: Less accurate in deeper morphological analysis compared to tools like MADAMIRA or AlKhalil [15].

• AMIRA: A tool for Arabic tokenization and base phrase chunking, offering rapid POS tagging with reasonably high accuracy. Pros: Lightweight and good for simple NLP pipelines. Cons: Outperformed by MADAMIRA and lacks support for complex morphological tasks [16].

3. Research Goals and Questions

The central goal of this research is to develop and evaluate automatic methods for identifying hidden tashbīh (implicit similes) in classical Arabic poetry, where the comparison is not signaled by explicit linguistic markers. This task lies at the intersection of classical Arabic rhetoric and modern natural language processing (NLP), aiming to bridge centuries-old poetic tradition with contemporary computational techniques.

Research Goals

To achieve this overarching aim, the research will pursue the following specific objectives:

- Construct a labeled dataset of Arabic poems containing both explicit and hidden tashbīh, based on an initial rule-based classifier and manual verification.
- Fine-tune transformer-based Arabic language models (e.g., AraBERT) to distinguish poems with tashbīh from those without, using both explicit and implicit instances.
- Design a transformation framework to manually or semi-automatically convert explicit tashbīh examples into implicit ones by removing or hiding the comparison marker while preserving semantic meaning and poetic structure.
- Develop a classification model capable of identifying tashbīh ḍimnī based on semantic features, syntactic context, and poetic discourse patterns.
- Evaluate the models' performance using standard metrics (accuracy, precision, recall, F1-score), alongside qualitative analysis of correctly and incorrectly classified examples.

Research Questions

The proposed research seeks to answer the following key questions:

- RQ1: Can machine learning models accurately identify tashbīh in Arabic poetry beyond rule-based lexical pattern matching?
- RQ2: What are the linguistic and semantic characteristics that distinguish hidden tashbīh from other poetic statements or metaphors?
- RQ3: How effective is a transformation-based method in creating high-quality examples of implicit tashbīh for training data augmentation?
- RQ4: To what extent can contextual language models like AraBERT generalize to detect implicit rhetorical devices in highly stylized, classical Arabic text?
- RQ5: What are the limitations and challenges in modeling deep figurative meaning computationally in the context of Arabic rhetoric?

4. Methodology and Tools

This section outlines the methodological approach and computational tools used to achieve the research goals of identifying hidden tashbīh (similes) in Arabic poetry. The methodology integrates rule-based extraction, data preprocessing, model fine-tuning, and evaluative analysis, all grounded in the principles of Arabic rhetoric and modern NLP.

Overview of the Research Pipeline

The research pipeline consists of the following stages:

1) Initial Rule-Based Detection of Tashbīh:
   - A rule-based system (converted from prior Java code to Python) was employed to detect explicit tashbīh in classical Arabic poems.
   - These rules searched for explicit simile particles (e.g., كأن، مثل، كما، كـ) and comparison-related verbs (e.g., شبه، ظن، خال) to identify overt similes.
2) Dataset Construction and Labeling:
   - Based on the output of the rule-based detection, a binary-labeled dataset was built: poems with tashbīh vs. poems without tashbīh.
   - Manual verification was conducted to ensure quality and correctness, especially for edge cases.
3) Text Cleaning and Preprocessing:
   - Diacritics (harakāt) were removed for consistency.
   - Poems were normalized (e.g., converting "ى" to "ي", removing punctuation).
   - Tokenization was applied using tools such as Farasa and MADAMIRA for further segmentation and linguistic analysis.
4) Model Training Using AraBERT:
   - The cleaned and labeled dataset was used to fine-tune AraBERT, a transformer-based model pre-trained on large Arabic corpora.

- A subset of poems was used for training, while the rest (over 28,000 poems) served as a test set.
- The fine-tuned classifier achieved around 96% accuracy, suggesting strong discriminative power in identifying tashbīh-laden verses.

5) Implicit Tashbīh Generation and Classification:
   - A manual process was used to convert explicit tashbīh into implicit forms, removing the comparison word while preserving meaning and rhetorical impact.
   - These transformed examples were added to the dataset to train a second model focused on detecting hidden tashbīh.
   - Feature extraction involved semantic embeddings, syntactic patterns, and context windows.
6) Evaluation and Error Analysis:
   - The model was evaluated using metrics such as accuracy, precision, recall, F1-score, and confusion matrix.
   - Special attention was given to false positives and false negatives to analyze whether the model captured the poetic simile beyond surface-level cues.

Tools and Frameworks

Several key tools and libraries are employed in this research:

- AlKhalil Morpho Sys 2: For root extraction, morphological analysis, and POS tagging on classical Arabic poetry.
  Strength: Excellent performance on vocalized/unvocalized classical texts.
  Weakness: Limited support for dialectal or informal Arabic.
- MADAMIRA: A comprehensive tool for morphological disambiguation and tokenization of Modern Standard Arabic.
  Strength: Integrates multiple preprocessing steps into a unified tool.

Weakness: Less robust on non-standard poetic constructions.

- Farasa: Lightweight Arabic segmenter and POS tagger, useful for fast preprocessing.
  Strength: High speed, scalable.
  Weakness: Lower precision in classical or rhetorical contexts.
- BAMA & SAMA: Classical tools from the Linguistic Data Consortium for Arabic morphological analysis.
  Strength: Foundational systems with wide lexical coverage.
  Weakness: Lack disambiguation and contextual awareness.
- AraBERT: A deep transformer-based model pre-trained on Arabic corpora.
  Strength: Strong contextual understanding, suitable for nuanced NLP tasks.
  Weakness: Requires careful fine-tuning and large GPU resources.
- scikit-learn & PyTorch: Used for training, evaluating, and visualizing model outputs and confusion matrices.

5. Initial Experiments and Results

The initial experiments focused on evaluating the performance of a fine-tuned AraBERT model in classifying Arabic poems into two categories: poems that contain tashbīh (similes) and those that do not.

Dataset and Setup

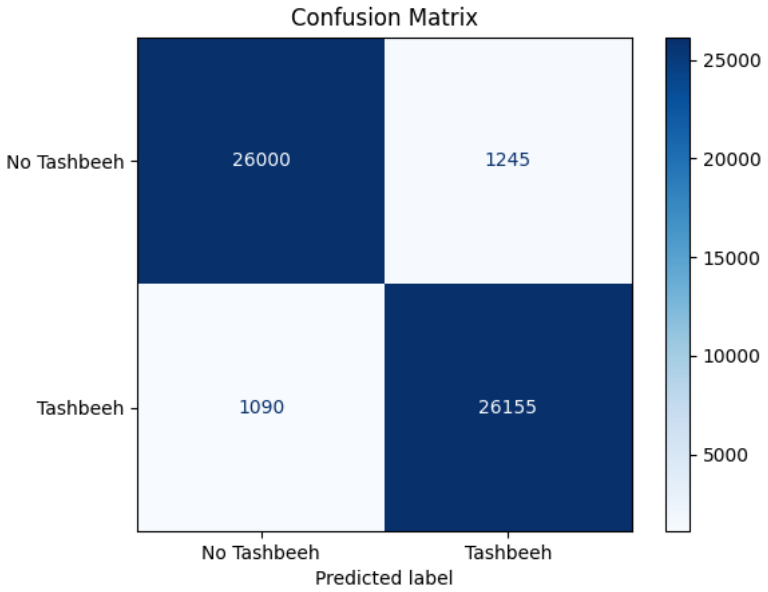
- The dataset was constructed by applying a rule-based detection system to label Arabic poems as containing tashbīh or not.
- From the full dataset, only 5% of the data was used for training, while the remaining 95% was split for validation and testing. Specifically:
  train_df, temp_df = train_test_split(df, test_size=0.95, stratify=df['label'], random_state=42)
  val_df, test_df = train_test_split(temp_df, test_size=0.98, stratify=temp_df['label'], random_state=42)
- This small training set was chosen due to the large size of the dataset and the high computational cost of training transformer-based models like AraBERT.
- Despite using only a small fraction of the data for training, the model achieved high performance, indicating the strength of contextual embeddings in Arabic NLP.

Results

The model's performance on the test set was evaluated using standard classification metrics:

- Accuracy: 95.71%
- Precision: 95.46%
- Recall: 95.99%

These scores show that the model is highly reliable in distinguishing between simile and non-simile poetry lines, even with limited training data.



The confusion matrix confirms the model's balanced ability to capture both classes with relatively low error rates.

Interpretation and Potential

The results confirm that AraBERT and similar transformer models are capable of learning complex semantic patterns in Arabic poetry. Even though the current dataset consists mostly of explicit tashbīh (with visible markers such as كأن, مثل), the model's success in capturing these patterns provides strong evidence that it understands rhetorical context rather than relying solely on keyword spotting.

This outcome suggests that such models can potentially be extended and fine-tuned further to identify implicit tashbīh (tashbīh ḍimnī) — where no clear comparative word exists — by learning more abstract patterns of analogy. These results form a solid

baseline and motivate the next stage of the thesis: focusing on generating, labeling, and detecting hidden similes.

6. Planned Research and Future Work

While the initial results demonstrate strong performance in detecting explicit tashbīh, the ultimate objective of this research is to extend this capability to implicit or hidden tashbīh (تشبيه ضمني)—those similes that lack overt comparative markers and rely on deeper semantic or contextual inference.

- Manual Transformation:

  A curated set of poems with explicit tashbīh will be manually rewritten to remove the simile marker (e.g., "مثل", "كأن") while preserving the core meaning and rhetorical structure. This will simulate how a poet might express a simile implicitly.

- Rule-Based and Semi-Automatic Transformation:

  Exploratory methods will be developed to automatically remove simile markers from known tashbīh lines, creating a larger set of implicit-style examples for training.

- Building a New Dataset:

  The transformed implicit tashbīh lines will be labeled and verified by Arabic language experts. A balanced dataset will be constructed containing explicit tashbīh, implicit tashbīh, and non-tashbīh lines.

- Developing Enhanced Detection Models:

  AraBERT or similar language models will be fine-tuned on the newly constructed dataset. Feature engineering will also be applied to enhance detection, such as sentence structure, syntactic trees, or rhetorical embeddings.

- Evaluation and Benchmarking:

  The model will be evaluated using accuracy, precision, recall, and confusion matrices. Qualitative analysis will also be conducted to interpret model decisions, particularly in ambiguous or misclassified cases

7. Contribution

This research aims to make significant contributions to the intersection of Arabic linguistics, digital humanities, and natural language processing. By focusing on the detection of both explicit and implicit tashbīh (similes) in classical Arabic poetry, this work bridges centuries-old rhetorical theory with modern AI tools.

Linguistic and Theoretical Contributions

- Modeling Hidden Rhetoric: The thesis explores and formalizes the concept of tashbīh ḍimnī (implicit simile) in computational terms—an area that has received very limited attention in Arabic NLP or literary analysis.
- Dataset Curation: It introduces a novel dataset of Arabic poetry annotated for both explicit and implicit similes, offering a resource that can support future research in figurative language processing for Arabic.
- Augmentation of Classical Theory: Through the transformation of explicit into implicit tashbīh, this work contributes to the study of Arabic balāghah by demonstrating new rhetorical equivalencies grounded in semantic preservation.

Technical and Computational Contributions

- Pipeline for Tashbīh Detection:

  The research presents a hybrid methodology that combines rule-based techniques with deep learning models (e.g., AraBERT) to achieve high-accuracy classification of poetic verses.
- Benchmarking NLP Models on Arabic Poetry:

  It evaluates the performance of transformer-based models on a uniquely complex domain—classical Arabic poetry—which is morphologically rich and syntactically free-form.
- Demonstrating Semantic Learning:

By achieving high accuracy even on a small training subset, the model shows that transformer models can meaningfully capture rhetorical intent in Arabic, setting the foundation for detecting deeper forms of metaphor and simile.

- Open-Source Codebase and Tools:

  The work includes clean, reusable Python code (available on GitHub) for processing, training, and evaluating Arabic poetry classification tasks—serving the community and future researchers.

8. References

[1] al-Hāshimī, A. (1998). Jawāhir al-Balāghah. Dar al-Kutub al-ʿIlmiyya.

[2] al-Jurjānī, ʿA. (1954). Asrār al-Balāgha. Istanbul.

[3] al-Zubaidī, M. M. (1975). Tāj al-ʿArūs. Kuwait.

[4] Ali, A. & Ahmad, H. (2015). Classical Arabic rhetoric. Journal of Arabic Literature.

[5] Abd Alhadi, H., Hussein, A. A., & Kuflik, T. (2023). Automatic identification of rhetorical elements. Digital Humanities Quarterly.

[6] Hussein, A. A. (2015). The Rhetorical Fabric of the Traditional Arabic Qaṣīda. Wiesbaden: Harrassowitz Verlag.

[7] Boudchiche, M. et al. (2017). AlKhalil Morpho Sys 2. Journal of King Saud University.

[8] Pasha, A. et al. (2014). MADAMIRA. Proceedings of LREC.

[9] Hussein et al. (2023). Limitations of REI. Digital Humanities Quarterly.

[10] Antoun, W. et al. (2020). AraBERT. Workshop on Open-Source Arabic Tools.

[11] Su, Y. et al. (2020). DeepMet. ACL.

[12] Leong, C. et al. (2020). MelBERT. EMNLP.

[13] Buckwalter, T. (2002). Buckwalter Arabic Morphological Analyzer Version 1.0. Linguistic Data Consortium, University of Pennsylvania.

[14] Maamouri, M., Bies, A., Buckwalter, T., & Mekki, W. (2010). Standard Arabic Morphological Analyzer (SAMA) Version 3.1 (LDC2010L01). Linguistic Data Consortium.

[15] Abdelali, A., Darwish, K., Durrani, N., & Mubarak, H. (2016). Farasa: A fast and furious segmenter for Arabic. In Proceedings of the 2016 Conference of the North American

Chapter of the Association for Computational Linguistics: Demonstrations (pp. 11–16). Association for Computational Linguistics.

[16] Diab, M. (2009). Second generation AMIRA tools for Arabic processing: Fast and robust tokenization, POS tagging, and base phrase chunking. In Proceedings of the 2nd International Conference on Arabic Language Resources and Tools (MEDAR), Cairo, Egypt.