

Pattern matching in meter detection of Arabic classical poetry

Abdelmalek Berkani
Information Management Institute
University of Neuchâtel
Neuchâtel, Switzerland
abdelmalek.berkani@unine.ch

Adrian Holzer
Information Management Institute
University of Neuchâtel
Neuchâtel, Switzerland
adrian.holzer@unine.ch

Kilian Stoffel
Information Management Institute
University of Neuchâtel
Neuchâtel, Switzerland
kilian.stoffel@unine.ch

Abstract—Arabic classical poetry meter is a sequence of patterns. A poetry verse is characterized by a meter and consists of two parts. Detecting classical poetry meter is important for teaching purposes, for poetry and prose categorization, for authorship recognition and for computational aesthetics. Automatically detecting the meter of any single verse written as a normal sentence is challenging. We need a global approach that processes phonological verse preparation, deals with distinguishing the first part of the verse from the second, handles meters disambiguation and covers the verse parts concordance. To tackle this challenge, we introduce a novel solution called the Arabic Meters Identification System (AMIS) that combines an exhaustive pattern data set, pattern matching and similarity. We evaluate our system on a vocalized poetry corpus and reach a precision of 99.3%.

Keywords—Arabic classical poetry, Arabic Arud, Arabic prosody, Arabic poetry patterns, Arabic poetry meters

I. INTRODUCTION

Meter is an essential feature in classical poetry. It characterizes verse arrangement and harmony. Meter is used in teaching [1], in poem classification [2] and authorship recognition. Beside other poetic features, meter is also used in computational aesthetics [3]. Thus, extracting meter from the verse is important both for education and research purposes.

Automatic meter detection is a challenging task that needs a complete approach in terms of data collection and processing. In this article, we introduce a new global approach AMIS for Arabic Meters Detection System based on an exhaustive data set of pattern combinations, on pattern matching and similarity. Since each meter may have many pattern alterations that lead to sequence redundancies and conflicts between meters [4], we need to disambiguate meter variants while keeping concordance between the two parts of the verse.

In classical poetry, even if the end of a verse part is written as a short vowel, it has to be pronounced as long vowel. The last phoneme of each verse part has to be adapted consequently [5]. In the context of a single poetry verse written as a normal sentence, the main issue is making distinction between verse parts. We have to adapt the data set, to manage both cases when the first part ends with short or long vowel.

To the best of our knowledge, the best accuracy of meter detection of a single poetry verse is 75% [6]. AMIS improves existing work by focusing on both data and processing aspects.

The data aspect is for building an exhaustive pattern combinations data set for all meter variants and pruning the data set by removing meter conflicts and non-concordant patterns between verse parts. The processing side is for verse phonological preparation, syllables segmentation, exact pattern matching and similarity to mitigate text preparation imperfections.

The next sections present background on Arabic poetry, review previous work on meter detection and motivate the approach we used. We then introduce our results and evaluation.

II. BACKGROUND

Classical poetry is the earliest form of Arabic literature [7]. The poem is a set of verses. Every verse “bayt” is composed of two parts. The first part “Sadr” and the second part “Ajoz”. In some rare exceptions, the verse does not have a second part. The verse is characterized by a meter. Meters are identified by unique names and aim to define the rhythm of the verse.

In Arabic, meters field of study is called “Arud” founded by Al-Khalil ibn Ahmad Al Farahidi (718 - 786). Arud is the prosody used for classical poetry in Arabic, Ottoman, Persian, Urdu and other eastern languages [1]. Table I shows the original sixteen classical meters with their respective patterns sequence.

A meter is a set of ordered patterns. A pattern is a named group of syllables (long: L, short: S, absence: A). The number, the type and the order of patterns make the difference between meters. The original sixteen meters of Table I are built on ten distinct patterns [5] shown in Table II.

The syllables sequence determines patterns names and meter. The following example shows an original occurrence of meter “Kamil” with its 6 patterns.

```
{Six groups of syllables: three in each part}
[(SSLSL)(SSLSL)(SSLSL)] [(SSLSL)(SSLSL)(SSLSL)]
{Six corresponding patterns: three in each part}
[(motafaa'ilon)(motafaa'ilon)(motafaa'ilon)]
[(motafaa'ilon)(motafaa'ilon)(motafaa'ilon)]
```

In terms of original patterns usage, [4] notices that most of poetry verses do not conform to the original form of meters. Poets use variants instead. A pattern variant occurs in case of change in its syllables sequence. A meter variant is an alteration of its original form due to change in patterns or their number. A meter variant that keeps the same number of

TABLE I
ORIGINAL POETRY METERS

Meter	Pattern sequence First Part Second Part	Patterns
Tawil	fa'oolon mafaa'ilon fa'oolon mafaa'ilon fa'oolon mafaa'ilon fa'oolon mafaa'ilon	8
Madid	faa'ilaaton faa'ilon faa'ilaaton faa'ilaaton faa'ilon faa'ilaaton	6
Bassit	mostaf'ilon faa'ilon mostaf'ilon faa'ilon mostaf'ilon faa'ilon mostaf'ilon faa'ilon	8
Wafir	mofaa'alaton mofaa'alaton fa'oolon mofaa'alaton mofaa'alaton fa'oolon	6
Kamil	motafaa'ilon motafaa'ilon motafaa'ilon motafaa'ilon motafaa'ilon motafaa'ilon	6
Hazaj	mafaa'ilon mafaa'ilon mafaa'ilon mafaa'ilon	4
Rajaz	mostaf'ilon mostaf'ilon mostaf'ilon mostaf'ilon mostaf'ilon mostaf'ilon	6
Ramal	faa'ilaaton faa'ilaaton faa'ilaaton faa'ilaaton faa'ilaaton faa'ilaaton	6
Sarii	mostaf'ilon mostaf'ilon faa'ilon mostaf'ilon mostaf'ilon faa'ilon	6
Monsarih	mostaf'ilon maf'oolaato mostaf'ilon mostaf'ilon maf'oolaato mostaf'ilon	6
Khafif	faa'ilaaton mostaf'i lon faa'ilaaton faa'ilaaton mostaf'i lon faa'ilaaton	6
Modarii	mafaa'ilon faa'i laaton mafaa'ilon faa'i laaton	4
Moqtadab	maf'oolaato mostaf'ilon maf'oolaato mostaf'ilon	4
Mojtath	mostaf'i lon faa'ilaaton mostaf'i lon faa'ilaaton	4
Motaqaarib	fa'oolon fa'oolon fa'oolon fa'oolon fa'oolon fa'oolon fa'oolon fa'oolon fa'oolon	8
Motadaarak	faa'ilon faa'ilon faa'ilon faa'ilon faa'ilon faa'ilon faa'ilon faa'ilon	8

TABLE II
ORIGINAL POETRY PATTERNS

Pattern	Syllables
faa'ilon	(LSL)
fa'oolon	(SLL)
mafaa'ilon	(SLLL)
mostaf'ilon	(LLSL)
mostaf'i lon	(LLS L)
mofaa'alaton	(SLSSL)
motafaa'ilon	(SSLSL)
maf'oolaato	(LLLS)
faa'ilaaton	(LSLL)
faa'i laaton	(LS LL)

patterns as the original form is called “Complete”. Any variant that uses less than the number of the original form is called “Partial”. e.g. “Kamil” Complete is a variant with six patterns as the original one. “Kamil” Partial is a version that has four patterns.

Meter detection relies on the vowels in the verse. Short vowels are diacritics above consonants known as “Fatha”, “Damma”, and “Kasra”. The absence of vowel is called “Sokoun”. Long vowels elongate the sound of short vowels. Each short vowel has a correspondent long vowel called respectively “Alif”, “Waw” and “Yae”.

III. RELATED WORK

Researchers have proposed meter detection methods based on the sixteen theoretical meters and patterns in three steps [6]. The first step is text conversion in order to keep only spoken letters. The second step is the segmentation phase where text is converted to syllables. Meter is detected in the last step by comparing the syllables sequence with the grammar stored previously. The data set used to evaluate meter detection consists of 128 verses from different Arabic poems with a success rate of 75%. Others have used recurrent neural network (RNN) to detect sixteen poetry Arabic meters and four English meters with an overall accuracy of 96.38% and 82.31%, respectively [8].

Further research goes beyond the theoretical meters and considers variants of original meters [9]–[12]. For instance [9] proposes a detection based on editing, consultation and knowledge bases modules. They evaluate the system on 20 poems and report good results without giving figures on detection accuracy. Further, in [10], researchers encode the prosody of each input text using Khashan’s method called “numerical prosody”. Authors report an overall accuracy of 98.6% based on the whole poem evaluation.

The rule based approach presented in [11] describes an algorithm that detects the correct meter in five steps. The algorithm is based on predefined rules for text conversion in prosody form. It uses only the first part of the verse. The algorithm was evaluated on a sample of classical Arabic poems and achieves an accuracy of 82%.

In terms of meter usage and value, the authors of [12] show that linguistic features based on the Arabic poetry meters are good attributes for authorship attribution. Authors argue that meter-based features outperform the usual linguistic features commonly used in authorship studies like word frequencies. They have also shown that features of Arabic classical poetry meters are suitable to distinguish authors in English as well as Arabic.

Some other researchers also used the Arabic poetry meter in authorship attribution [13]–[15]. They used meter as a feature to distinguish authors.

The main difference of the approaches is the detection phase. Some of them use only the first part of the verse [11] with the risk to detect a prose sentence or free poetry as a classical poetry verse.

Some methods use the whole poem or more than one verse [1] or rely on the writing style [16], special characters or spaces as criteria to separate the first part and the second part. This approach is challenging because of the possible different styles in the same poem.

Other methods use only theoretical set of patterns that compose each meter. These theoretical sets are rarely used. Most of them are altered and many variants exist for each meter [4].

Researches in Arabic meter identification expand more upon the detection process than the building of patterns data set they have used as reference. Making this data set exhaustive and available, is a step forward in research on Arabic text analysis.

IV. ARABIC METERS IDENTIFICATION SYSTEM

The main contribution of this paper is a novel system called AMIS that aims at identifying any original or variant meter of any Arabic classical poetry verse. AMIS is made up with two main modules. The first one is for data collection and the second one is for meter detection processing. Both modules are divided in sub-modules.

A. AMIS data collection

The data collection module shown in Figure 1 aims to build, in an iterative way, a data set of the essential pattern combinations that allow accurate meter detection of any poetry verse. The data set has to contain only the necessary rows, i.e. neither too much nor not enough in order to avoid redundancies, to prevent meter conflicts and to ensure uniqueness of pattern sequences. We start by collecting combinations from literature, adding combinations to handle the single verse case, pruning the data set to keep concordance verse parts and removing ambiguities between meters.

Each row in the pattern combinations data set represents an occurrence of one possible arrangement of patterns for a given meter. The data set contains meter name, patterns and their corresponding syllables. The data set accepts up to eight patterns per row. That is the maximum number of patterns in classical Arabic poetry. At the end of building the data set, the concatenation of syllables is unique.

```

meter: meter name
{First part}
fp1p: pattern 1, fp1s: syllables of pattern 1
fp2p: pattern 2, fp2s: syllables of pattern 2
fp3p: pattern 3, fp3s: syllables of pattern 3
fp4p: pattern 4, fp4s: syllables of pattern 4
{Second part}
sp1p: pattern 1, sp1s: syllables of pattern 1
sp2p: pattern 2, sp2s: syllables of pattern 2
sp3p: pattern 3, sp3s: syllables of pattern 3
sp4p: pattern 4, sp4s: syllables of pattern 4

```

1) *Pattern combinations collection from literature*: We start building the data set by adding pattern combinations specified in the original forms of the Arabic meters as shown in Table I. Then we added successively combinations documented in literature [4], [5], [17], [18]. After each adding, we tested meter detection against poetry corpus to check redundancies and to measure accuracy.

2) *Single Poetry Verse*: An Arabic classical poem is written in a way that the first part of the verse is separated from the second part either by many space characters or by writing the second part on a second line preceded by a clear indentation. However, when a standalone verse is written as a sentence or surrounded by prose text, the first part is separated from the second part by one space character only. Even if the number of patterns is the same in the first and second parts, the number of syllables may vary between the two parts making parts distinction difficult. Due to the saturation rule in Arabic poetry, the end of a verse part has to be pronounced as a long vowel even if it is written as a short one. Thus, the data set has to handle the case of first part ending with short vowel. e.g. The

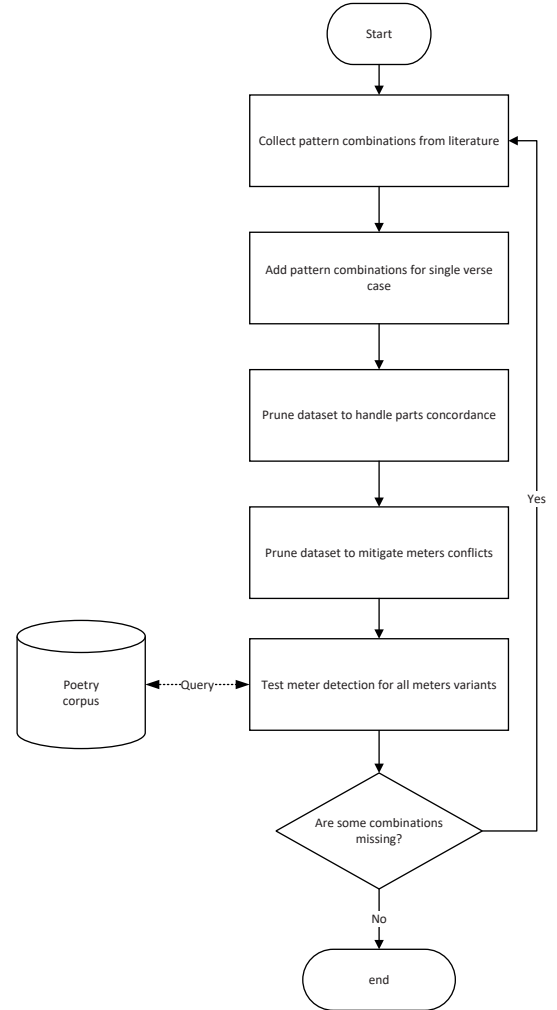


Fig. 1. AMIS Data Collection

meter “Bassit” complete, in addition to its other combinations, has to expect these both short and long vowel at the end of the first part:

```

[LLSLSLLLSLL(L)] [SLSLLSLLLSLL]
[LLSLSLLLSLL(S)] [SLSLLSLLLSLL]

```

3) *Verse parts concordance*: In order to preserve verse harmony, there is a concordance, for some meters, between the last pattern of the first part and the last one of the second part [17]. Non-concordant combinations are not relevant for meter detection. Here are the concordant patterns we kept at the end position of each verse part.

```

{"Tawil" Complete}
["mafaa' iilon", "mafaa' iilo", "mafaa' ilon",
"mafaa' ilo"] and ["mafaa' iilon", "mafaa' ilon"]
["fa' oolon", "fa' oolo"] and ["fa' oolon"]

```

```

{"Madid" Complete}
["faa' ilaaton", "faa' ilaato", "fa' ilaaton",

```

"fa'ilaato"] and ["faa'ilaaton","fa'ilaaton"]
 ["faa'ilon","faa'ilo"] and
 ["faa'ilon","faa'ilaan","fa'ilon","fi'lon"]
 ["fa'ilon","fa'ilo"] and
 ["faa'ilon","fa'ilon","fi'lon"]

```
{ "Madid" Partial }
["faa'ilon","faa'ilo"] and
["faa'ilon","faa'ilaan","fa'ilon","fi'lon"]
["fa'ilon","fa'ilo"] and
["faa'ilon","fa'ilon","fi'lon"]
```

```
{ "Bassit" Complete }
["fa'ilon","fa'ilo"] and ["fa'ilon","fi'lon"]
["fi'lon","fi'lo"] and ["fi'lon"]
```

```
{ "Bassit" Partial 2 }
["maf'oolon","maf'oolo"] and ["maf'oolon"]
["fa'oolon","fa'oolo"] and ["fa'oolon"]
```

```
{ "Kamil" Complete }
["fa'ilon","fa'ilo"] and ["fa'ilon","fi'lon"]
```

```
{ "Rajaz" complete and "Rajaz" partial 1 }
["maf'oolon","maf'oolo","fa'oolon","fa'oolo"]
and ["maf'oolon","fa'oolon"]
["maf'oolaan","fa'oolaan"] and
["maf'oolaan","fa'oolaan"]
```

```
{ "Sarii" complete }
["faa'ilon","faa'ilo"] and
["faa'ilon","faa'ilaan","fi'lon"]
["fa'ilon","fa'ilo"] and ["fa'ilon","fi'lon"]
["fi'lon","fi'lo"] and ["fi'lon"]
```

```
{ "Khafif" complete }
["faa'ilon","faa'ilo"] and ["faa'ilon"]
```

```
{ "Motadaarak" all variants }
Avoid "faa'ilon" or "faa'ilo"
with "fi'lon" or "fi'lo" in the same verse
```

4) *Meter ambiguities*: Pattern alterations lead to sequence redundancies in different meters. e.g. a sequence of six times the pattern "mostaf'ilon" is present in both complete versions of meters "Kamil" and "Rajaz". We can decide to display both meters as a result or to remove one of both. To provide the most relevant meter as result, we used pattern dominance in the theoretical original meter composition shown in Table I. The dominant pattern is the most present one in the meter's patterns sequence.

```
{ "Kamil" complete and "Rajaz" complete }
{ Pattern "mostaf'ilon" is dominant in "Rajaz" }
delete data
where meter = 'kamilC' and fp1p = 'mostaf'ilon'
and fp2p = 'mostaf'ilon'
and fp3p in ('mostaf'ilon','mostaf'ilo')
and sp1p = 'mostaf'ilon' and sp2p = 'mostaf'ilon'
and sp3p = 'mostaf'ilon';
```

```
delete data
where meter = 'rajazC'
and fp1p = 'mostaf'ilon' and fp2p = 'mostaf'ilon'
and fp3p in ('mostaf'ilon','mostaf'ilo')
and sp1p = 'mostaf'ilon' and sp2p = 'mostaf'ilon'
and sp3p = 'maf'oolon';
```

```
{ "Kamil" complete and "Sarii" complete }
{ Pattern "mostaf'ilon" is dominant in "Sarii" }
delete data
where meter = 'kamilC' and fp1p = 'mostaf'ilon'
```

TABLE III
DATA SET COMBINATION EVOLUTION

Phase	Added	Removed	Final
Collection from literature	31402	0	31402
Single verse case	25502	0	56904
Verse parts concordance	0	33732	23172
Meter ambiguities	0	54	23118

```
and fp2p = 'mostaf'ilon'
and fp3p in ('fa3ilon','fa3ilo')
and sp1p = 'mostaf'ilon' and sp2p = 'mostaf'ilon'
and sp3p in ('fa3ilon','fi3lon');
```

```
{ "Kamil" partial and "Rajaz" partial }
{ Pattern "mostaf'ilon" is dominant in "Rajaz" }
delete data
where meter = 'kamilP' and fp1p = 'mostaf'ilon'
and fp2p in ('mostaf'ilon','mostaf'ilo')
and sp1p = 'mostaf'ilon' and sp2p = 'mostaf'ilon';
```

```
{ "Wafir" partial and "Hazaj" }
{ Pattern "mafaa'iilon" is dominant in "Hazaj" }
delete data
where meter = 'wafirP' and fp1p = 'mafaa'iilon'
and fp2p in ('mafaa'iilon','mafaa'ilo')
and sp1p = 'mafaa'iilon'
and sp2p = 'mafaa'iilon';
```

```
{ "Rajaz" partial and "Hazaj" }
{ Pattern "mafaa'iilon" is dominant in "Hazaj" }
delete data
where meter = 'rajazP' and fp1p = 'mafaa'ilon'
and fp2p in ('mafaa'ilon','mafaa'ilo',
'fa'oolon','fa'oolo')
and sp1p = 'mafaa3ilon'
and sp2p in ('mafaa'ilon','fa'oolon');
```

```
{ "Ramal" partial and "Madid" partial }
{ Pattern "faa'ilaaton" is dominant in "Ramal" }
delete data
where meter = 'madidP'
and fp1p in ('faa'ilaaton','fa'ilaaton')
and fp2p in ('faa'ilon','faa'ilo',
'fa'ilon','fa'ilo')
and sp1p in ('faa'ilaaton','fa'ilaaton')
and sp2p in ('faa'ilon','fa'ilon','faa'ilaan');
```

The Table III shows the evolution of data collection.

B. AMIS processing

Beside the patterns data set, AMIS handles all the steps of meter detection as shown in Figure 2.

1) *Text preparation*: The aim of this sub-module is to keep only what is pronounced. The final result is text written in prosody form to highlight phonemes in order to make possible syllables segmentation later.

```
{ Text cleaning: keep only Arabic characters, space
and diacritics }
text = removeNonArabicChars(text);
text = removeSpecialChars(text);
text = removePunctuation(text);
text = removeDigits(text);
```

```
{ Vowel elongation for some known words written with
short vowels and pronounced with long vowels }
text = elongateVowels(text);
```

```
{Definite article: depends on the type of the letter
that follows the article ("lunar" or "solar")
and the position of the word in the text}
text = rewriteDefiniteArticle(text);
```

```
{Emphasis character "Shadda": double the consonant
that has the emphasis character}
text = rewriteEmphasis(text);
```

```
{Nunation: replace double diacritics at the end of
word by letter "N", simple diacritic and an
absence of vowel}
text = rewriteNunation(text);
```

```
{Sandhi: The pronunciation of a text does not stop
at the end of the word. It includes the next one
}
text = handleSandhi(text);
```

```
{Saturation: vowel elongation at the end of verse
parts and also in case of letter "H" for
masculine possessive pronoun or "M" for plural}
text = handleSaturation(text);
```

2) *Syllables segmentation*: Syllables segmentation sub-module includes two steps. The first one takes as input the text in prosody form and encodes vowels to 0 and 1. The second one converts the resulting code to syllables.

```
{Vowels: short vowel is converted to 1, Long or
absence of vowel is converted to 0}
code = encodeVowels(text);
```

```
{Syllables: L (Long), S (Short), A (Absence)
1 followed by 0 is converted to 'L',
Individual 1 is converted to 'S'.
Remaining 0 is converted to 'A'.}
syllables = getSyllables(code);
```

```
{Example}
Code: 101011010101101010110111011010
Syllables: LLSLLLLLSLSSLSLL
```

3) *Exact pattern matching*: This sub-module searches, in the pattern data set, the row that satisfies two conditions:

- The length of the input syllables equals the number of syllables stored in the data set.
- the syllables of the verse match regular expressions, based on the stored pattern sequences in the data set, from the beginning and the end of the sequence.

```
{Example}
Verse sequence: 28 syllables
[LLSLLLLSLSLS][SLSLSSLSLSSLSL]
```

```
Found sequence in the data set: 28 syllables
First part: [LLSLLLLSLSLS]
Second part: [SLSLSSLSLSSLSL]
```

```
From the beginning:
matching('LLSLLLLSLSLSLSSLSLSSLSL',
'^LLSLLLLSLSLS') = true
```

```
From the end:
matching('LLSLLLLSLSLSLSSLSLSSLSL',
'SLSLSSLSLSSLS') = true
```

```
{Result: displays the meter name, [0] for exact
matching and the patterns sequence found for the
verse}
```

```
[0] "Kamil" complete:
["mostaf'ilon" "mostaf'ilon" "motafaa'ilon"]
["motafaa'ilon" "motafaa'ilon" "motafaa'ilon"]
```

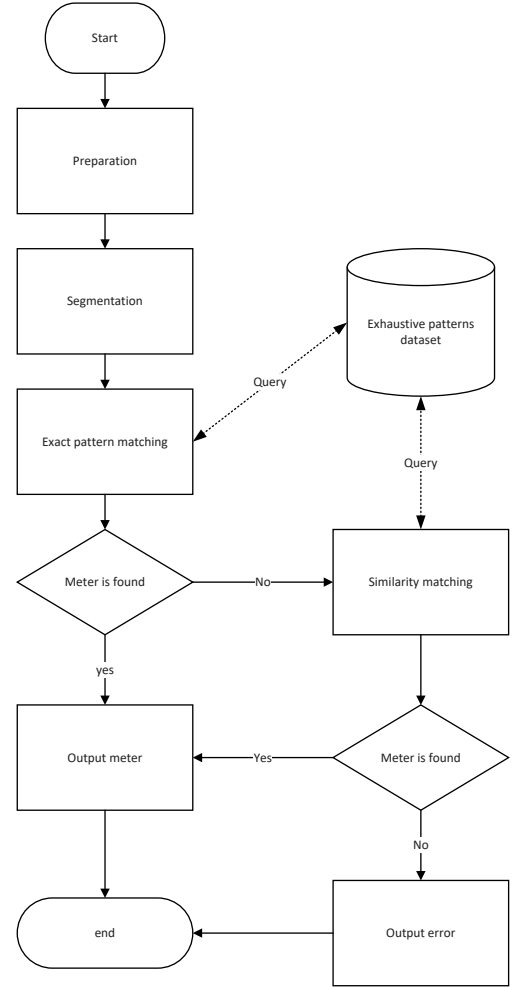


Fig. 2. AMIS Processing

4) *Similarity*: In most cases where the exact matching sub-module fails to detect the meter, it is because of the quality of verse preparation. The saturation rule implementation has some known issues. It uses only morphological and syntactic layers while detecting possessive pronouns or plurals needs a complete part of speech analysis. Not all words ending with "H" are possessive pronoun and not all those ending with "M" are plural [5]. We introduce an additional step to make an approximate matching with a tolerance threshold of maximum two errors. We look, in the pattern combinations data set, for the closer sequence that has the same syllables number and positions with at most two differences.

```
{Example}
Verse sequence:
[SLSLSSLSLSSLS][LLL(S)L)LSLLLL]
Closer sequence in the data set:
[SLSLSSLSLSSLS][LLL(S)LSLLLL]
```

TABLE IV
METERS PATTERN COMBINATIONS

Meter variant	Patterns	Combinations	Syl. Range
Tawil complete	8	640	0
Madid complete	6	352	1
Madid partial	4	16	0
Bassit complete	8	1944	1
Bassit partial 1	6	864	1
Bassit partial 2	6	324	0
Wafir complete	6	32	2
Wafir partial	4	30	2
Kamil complete	6	442	4
Kamil partial	4	128	3
Hazaj complete	4	288	1
Rajaz complete	6	7680	1
Rajaz partial 1	4	858	1
Rajaz partial 2	3	112	n/a
Rajaz partial 3	2	20	n/a
Ramal complete	6	1536	2
Ramal partial	4	384	2
Sarii complete	6	3072	1
Monsarih complete	6	648	1
Khafif complete	6	416	1
Khafif partial	4	48	1
Modarii complete	4	8	0
Moqtadab complete	4	18	0
Mojtath complete	4	48	1
Motaqaarib complete	8	1280	2
Motaqaarib partial	6	320	2
Motadaarak complete	8	1022	4
Motadaarak partial 1	6	474	4
Motadaarak partial 2	4	114	3
	Total	23118	

There is one difference. The meter is:

[1] "Kamil" complete:
["motafaa'ilon" "motafaa'ilon" "motafaa'ilon"]
["mostaf'ilon" "motafaa'ilon" "mostaf'ilon"]

V. RESULTS

In this research, we used the sixteen meters known in literature and all their variants. We found 23118 pattern combinations as shown in Table IV. This data set contains essential possibilities to detect accurately the meter of any single verse in Arabic classical poetry.

A. Meters

The following lines show, the exhaustive final pattern positions in the data set. e.g. in meter "Tawil", the indication "fa'oolo"(1,3,4,5,7) means that the pattern "fa'oolo" is present in positions 1,3,4,5 and 7 but not in 2, 6 and 8. Meter "Tawil" has 8 patterns, 4 in each part.

{Meter "Tawil"}
"fa'oolo"(1,3,4,5,7), "fa'oolon"(1,3,4,5,7,8),
"mafaa'iilo"(4), "mafaa'iilon"(2,4,6,8),
"mafaa'ilo"(4), "mafaa'ilon"(2,4,6,8)

{Meter "Madid" Complete}
"faa'ilaan"(6), "faa'ilaato"(3),
"faa'ilaaton"(1,3,4,6), "faa'ilo"(3),
"faa'ilon"(2,3,5,6), "fa'ilaato"(3),
"fa'ilaaton"(1,3,4,6), "fa'ilo"(3),
"fa'ilon"(2,3,5,6), "fi'lon"(6)

{Meter "Madid" Partial}
"faa'ilaan"(4), "faa'ilaaton"(1,3), "faa'ilo"(2),
"faa'ilon"(2,4), "fa'ilaaton"(1,3), "fa'ilo"(2),
"fa'ilon"(2,4), "fi'lon"(4)

{Meter "Bassit" Complete}
"faa'ilon"(2,6), "fa'ilo"(4), "fa'ilon"(2,4,6,8),
"fi'lo"(4), "fi'lon"(4,8), "mafaa'ilon"(1,3,5,7),
"mofta'ilon"(1,3,5,7), "mostaf'ilon"(1,3,5,7)

{Meter "Bassit" Partial 1}
"faa'ilon"(2,5), "fa'ilon"(2,5), "fa'oolon"(6),
"mafaa'ilo"(3), "mafaa'ilon"(1,3,4),
"maf'oolon"(6), "mofta'ilo"(3),
"mofta'ilon"(1,3,4), "mostaf'ilaan"(6),
"mostaf'ilo"(3), "mostaf'ilon"(1,3,4,6)

{Meter "Bassit" Partial 2}
"faa'ilon"(2,5), "fa'ilon"(2,5), "fa'oolo"(3),
"fa'oolon"(3,6), "mafaa'ilon"(1,4),
"maf'oolo"(3), "maf'oolon"(2,3,5,6),
"mofta'ilon"(1,4), "mostaf'ilon"(1,4)

{Meter "Wafir" Complete}
"fa'oolo"(3), "fa'oolon"(3,6),
"mafaa'iilon"(1,2,4,5), "mofaa'alaton"(1,2,4,5)

{Meter "Wafir" Partial}
"mafaa'iilo"(2), "mafaa'iilon"(1,2,3,4),
"mofaa'alato"(2), "mofaa'alaton"(1,2,3,4)

{Meter "Kamil" Complete}
"fa'ilaato"(3), "fa'ilaaton"(3,6), "fa'ilo"(3),
"fa'ilon"(3,6), "fi'lon"(6), "maf'oolon"(6),
"mostaf'ilo"(3), "mostaf'ilon"(1,2,3,4,5,6),
"motafaa'ilo"(3), "motafaa'ilon"(1,2,3,4,5,6)

{Meter "Kamil" Partial}
"mostaf'ilaato"(2), "mostaf'ilaaton"(2,4),
"mostaf'ilo"(2), "mostaf'ilon"(1,2,3),
"motafaa'ilaan"(4), "motafaa'ilaato"(2),
"motafaa'ilaaton"(2,4), "motafaa'ilo"(2),
"motafaa'ilon"(1,2,3,4),

{Meter "Hazaj"}
"fa'oolo"(2), "fa'oolon"(2,4),
"mafaa'iilo"(1,2,3), "mafaa'iilon"(1,2,3,4),
"mafaa'ilo"(1,2,3), "mafaa'ilon"(1,2,3,4)

{Meter "Rajaz" Complete}
"fa'ilaton"(1,2,4,5), "fa'oolaan"(3,6),
"fa'oolo"(3), "fa'oolon"(3,6), "mafaa'ilo"(3),
"mafaa'ilon"(1,2,3,4,5,6), "maf'oolaan"(3,6),
"maf'oolo"(3), "maf'oolon"(3,6), "mofta'ilo"(3),
"mofta'ilon"(1,2,3,4,5,6), "mostaf'ilo"(3),
"mostaf'ilon"(1,2,3,4,5,6)

{Meter "Rajaz" Partial 1}
"fa'ilaton"(1,3), "fa'oolaan"(2,4), "fa'oolo"(2),
"fa'oolon"(2,4), "mafaa'ilo"(2),
"mafaa'ilon"(1,2,3,4), "maf'oolaan"(2,4),
"maf'oolo"(2), "maf'oolon"(2,4), "mofta'ilo"(2),
"mofta'ilon"(1,2,3,4), "mostaf'ilo"(2),
"mostaf'ilon"(1,2,3,4)

{Meter "Rajaz" Partial 2}
{"Rajaz" Partial 2 has no second part.
Its 3 patterns are in the first part only}
"fa'ilaton"(1,2), "fa'oolaan"(3), "fa'oolon"(3),
"mafaa'ilon"(1,2,3), "maf'oolaan"(3),
"maf'oolon"(3), "mofta'ilon"(1,2,3),
"mostaf'ilon"(1,2,3)

{Meter "Rajaz" Partial 3}

{ "Rajaz" Partial 3 has no second part.
Its 2 patterns are in the first part only }
"fa' ilaton"(1), "fa' oolon"(2), "mafaa' ilon"(1,2),
"maf' oolon"(2), "mofta' ilon"(1,2),
"mostaf' ilon"(1,2)

{ Meter "Ramal" Complete }
"faa' ilaan"(3,6), "faa' ilaataan"(3,6),
"faa' ilaato"(3), "faa' ilaaton"(1,2,3,4,5,6),
"faa' ilo"(3), "faa' ilon"(3,6), "fa' ilaan"(3,6),
"fa' ilaataan"(3,6), "fa' ilaato"(3),
"fa' ilaaton"(1,2,3,4,5,6),
"fa' ilo"(3), "fa' ilon"(3,6)

{ Meter "Ramal" Partial }
"faa' ilaan"(2,4), "faa' ilaataan"(2,4),
"faa' ilaato"(2), "faa' ilaaton"(1,2,3,4),
"faa' ilo"(2), "faa' ilon"(2,4), "fa' ilaan"(2,4),
"fa' ilaataan"(2,4), "fa' ilaato"(2),
"fa' ilaaton"(1,2,3,4), "fa' ilo"(2),
"fa' ilon"(2,4)

{ Meter "Sarii" Complete }
"faa' ilaan"(6), "faa' ilo"(3), "faa' ilon"(3,6),
"fa' ilaton"(1,2,4,5), "fa' ilo"(3),
"fa' ilon"(3,6), "fi' lo"(3), "fi' lon"(3,6),
"mafaa' ilon"(1,2,4,5), "mofta' ilon"(1,2,4,5),
"mostaf' ilon"(1,2,4,5)

{ Meter "Monsarih" Complete }
"faa' ilaato"(2,5), "mafaa' iilo"(2,5),
"mafaa' ilon"(1,4), "maf' oolaato"(2,5),
"maf' oolon"(6), "mofta' ilo"(3),
"mofta' ilon"(1,3,4,6), "mostaf' ilo"(3),
"mostaf' ilon"(1,3,4)

{ Meter "Khafif" Complete }
"faa' ilaato"(3), "faa' ilaaton"(1,3,4,6),
"faa' ilo"(3), "faa' ilon"(3,6),
"fa' ilaato"(3), "fa' ilaaton"(1,3,4,6),
"mafaa' ilon"(2,5), "maf' oolo"(3),
"maf' oolon"(3,6), "mostaf' ilon"(2,5)

{ Meter "Khafif" Partial }
"faa' ilaaton"(1,3), "fa' ilaaton"(1,3),
"fa' oolon"(4), "mafaa' ilo"(2),
"mafaa' ilon"(2,4), "mostaf' ilo"(2),
"mostaf' ilon"(2,4)

{ Meter "Modaarii" }
"faa' ilaato"(2), "faa' ilaaton"(2,4),
"mafaa' iilo"(1,3), "mafaa' iilon"(1,3)

{ Meter "Moqtadab" }
"faa' ilaato"(1,3), "mafaa' iilo"(1,3),
"maf' oolaato"(1,3), "mofta' ilo"(2),
"mofta' ilon"(2,4)

{ Meter "Mojtath" }
"faa' ilaato"(2), "faa' ilaaton"(2,4),
"fa' ilaato"(2), "fa' ilaaton"(2,4),
"mafaa' ilon"(1,3), "maf' oolon"(4),
"mostaf' ilon"(1,3)

{ Meter "Motaqaarib" Complete }
"fa' "(4,8), "fa' al"(4,8), "fa' ool"(4,8),
"fa' oolo"(1,2,3,4,5,6,7),
"fa' oolon"(1,2,3,4,5,6,7,8)

{ Meter "Motaqaarib" Partial }
"fa' "(3,6), "fa' al"(3,6), "fa' ool"(3,6),
"fa' oolo"(1,2,3,4,5), "fa' oolon"(1,2,3,4,5,6)

{ Meter "Motadaarak" Complete }

TABLE V
METER DETECTION PRECISION

Matching	Verses	%
Exact	2597	95.80
Similarity 1 error	78	2.90
Similarity 2 errors	17	0.60
Undetected	19	0.70
Total	2711	100

"faa' ilo"(4), "faa' ilon"(1,2,3,4,5,6,7,8),
"fa' ilo"(4), "fa' ilon"(1,2,3,4,5,6,7,8),
"fi' lo"(4), "fi' lon"(1,2,3,4,5,6,7,8),

{ Meter "Motadaarak" Partial 1 }
"faa' ilaan"(3,6), "faa' ilo"(3),
"faa' ilon"(1,2,3,4,5,6), "fa' ilo"(3),
"fa' ilon"(1,2,3,4,5,6), "fi' lo"(3),
"fi' lon"(1,2,3,4,5,6)

{ Meter "Motadaarak" Partial 2 }
"faa' ilaan"(2,4), "faa' ilo"(2),
"faa' ilon"(1,2,3,4), "fa' ilo"(2),
"fa' ilon"(1,2,3,4), "fi' lo"(2),
"fi' lon"(1,2,3,4)

B. Evaluation

To evaluate meter detection, we build a vocalized corpus of 2711 verses. We used available sources on the Internet where the poem meter is known [19] and complement information about verses and their vocalization using alternatives sources [20]. The corpus includes all meter variants used by 61 distinct poets in 107 poems of 9 poetry eras from pre-Islamic to modern period.

1) *Precision*: We calculate the detection precision by dividing the number of correct detection by the number of all verses in the corpus.

$$Precision = \frac{VersesDetected}{AllVerses}$$

Among 2711 verses, 2597 are correctly detected in the phase of the exact pattern matching. As shown in Table V, we reach a precision of 95.8%. We detect 95 verses in the similarity phase while 19 verses remain undetected. Thus, if we add the exact matching to the similarity detection, we achieve a global precision of 99.3%.

Undetected meters shown in Table VI are due, in most cases to text vocalization or verse preparation imperfections. Another reason is the fact that some poets vary the meters, intentionally, inside the same verse. Some others use broken verses where the meter is different from the known meter variants making the detection incomplete.

VI. DISCUSSION AND CONCLUSION

As shown in Table IV, the 23118 essential combinations are distributed over 29 meter variants. The number of combinations vary significantly between meters while the range of the number of syllables between first and second parts vary from 0 to 4. Two special meter variants "Rajaz" partial 2 and 3 have no second part.

TABLE VI
UNDETECTED VERSES

Meter	Total verses	Undetected verses	Percent
Bassit complete	178	2	1.12%
Bassit partial 1	21	1	4.76%
Bassit partial 2	159	3	1.89%
Kamil partial	79	1	1.27%
Rajaz complete	112	2	1.79%
Rajaz partial 1	176	3	1.70%
Rajaz partial 3	50	1	2.00%
Sarii complete	110	1	0.91%
Monsarih complete	101	2	1.98%
Moqtadab complete	102	2	1.96%
Mojtath complete	88	1	1.14%

In this research, we used 43 patterns: 8 originals and 35 altered variants. The two patterns “mostaf’ilaan” and “motafaa’ilaan” are never present in the first part of the verse. Eleven patterns are not present in the second part: “fa’ilaato”, “fa’ilo”, “faa’ilo”, “fi’lo”, “maf’oolo”, “mofaa’alato”, “mofta’ilo”, “mostaf’ilaato”, “mostaf’ilo”, “motafaa’ilaato”, and “motafaa’ilo”.

We found that some patterns are more used than others: 80% of all possibilities are hold by only ten patterns: “mostaf’ilon”, “mafaa’ilon”, “mofta’ilon”, “fa’ilon”, “fa’ilaton”, “fa’oolon”, “faa’ilon”, “fa’oolo”, “fi’lon” and “fa’ilaaton”.

Many Arabic meters are close to each other. Any difference in the verse can change the meter variant. In the context of a single verse, we can rely only on the detection result. In some cases, the meter detected is different from the announced meter for the whole poem because in literature, poem’s meter is the meter of the first verse.

This research has addressed accurate meter detection of any single verse of classical Arabic poetry. We proposed a global approach to build an exhaustive data set of the necessary pattern combinations, that keeps concordance between both parts of the verse and resolve meter ambiguities. Meter detection is a challenging task. Highlighting sounds from text requires phonological text preparation, vowels encoding and syllables segmentation. Beside exact pattern matching, we used similarity comparison to mitigate the impact of text preparation imperfections.

The future work is improving the AMIS system, testing it on larger corpora and making it publicly available with the patterns combinations data set and the poetry corpus we used.

REFERENCES

- [1] Atakan Kurt and Mehmet Kara. An algorithm for the detection and analysis of arud meter in diwan poetry. *Turkish Journal of Electrical Engineering and Computer Sciences*, 20:948–963, 2012.
- [2] Hamid R Tizhoosh, Farhang Sahba, and Rozita Dara. Poetic features for poem recognition: A comparative study. *Journal of Pattern Recognition Research*, 3(1):24–39, 2008.
- [3] Justine Kao and Dan Jurafsky. A computational analysis of style, affect, and imagery in contemporary poetry. In *Proceedings of the NAACL-HLT 2012 workshop on computational linguistics for literature*, pages 8–17, 2012.
- [4] Mustapha Harakat. *Poetry meters (Awzan Al Shear)*. Al Dar Al Thaqafiya LiNachr, 1998.

- [5] Abdul Aziz Atiq. *Science of prosody (Ilmu al-Arudh wa al-Qafiyah)*. Dar Al Nahda Al Arabia, 1987.
- [6] Mohammad Awwad AlNagdawi. Finding arabic poem meter using context free grammar. *Journal of Communications and Computer Engineering*, 3:52, 2016.
- [7] Alfalahi Ahmed, Ramdani Mohamed, and Bellafkih Mostafa. Authorship attribution in arabic poetry using nb, svm, smo. *SITA 2016 - 11th International Conference on Intelligent Systems: Theories and Applications*, pages 1–5, 2016.
- [8] Waleed A Yousef, Omar M Ibrahime, Taha M Madbouly, and Moustafa A Mahmoud. Learning meters of arabic and english poems with recurrent neural networks: a step forward for language understanding and synthesis. *arXiv preprint arXiv:1905.05700*, 2019.
- [9] M. A. Ismail, M. I. El Adawy, Hesham Keshk, and S. A. Saleh. Expert system for testing the harmony of arabic poetry. *Journal of Engineering Science*, 32:401–411, 2004.
- [10] Maytham Alabbas, Zainab A. Khalaf, and Khashan M. Khashan. Basrah: An automatic system to identify the meter of arabic poetry. *Natural Language Engineering*, 20:131–149, 2014.
- [11] Belal Abuata and Asma Al-Omari. A rule-based algorithm for the detection of arud meter in classical arabic poetry. *Researchgate.Net*, 2016.
- [12] Ahmed Ibrahim Ahmed Omer and Michael Philip Oakes. Arud, the metrical system of arabic poetry, as a feature set for authorship attribution. *Proceedings of IEEE/ACS International Conference on Computer Systems and Applications, AICCSA*, 2017-Octob:431–436, 2018.
- [13] Hazel Scott. *Pegs, cords, and ghuls: Meter of classical Arabic poetry*. PhD thesis, Swarthmore College. Department of Linguistics, 2010.
- [14] Al-Falahi Ahmed, Ramdani Mohamed, and Bellafkih Mostafa. Machine learning for authorship attribution in arabic poetry. *International Journal of Future Computer and Communication*, 6:42–46, 2017.
- [15] A Ahmed, Ramdani Mohamed, and Bellafkih Mostafa. Machine learning for authorship attribution in arabic poetry. *Int. J. Future Comput. Commun*, 6(2):42–46, 2017.
- [16] Abdulrahman Almuhareb, Ibrahim Alkharashi, Lama AL Saud, and Haya Altuwaijri. Recognition of classical arabic poems. In *Proceedings of the Workshop on Computational Linguistics for Literature*, pages 9–16, 2013.
- [17] Ahmad Al Hachimi. *Gold balance in building Arabic Poetry (Meezan Al Dahab Fi Sinaat Shear Al Arab)*. Maktabat Al Adab, 1997.
- [18] Ghazi Yammout. *Arabic poetry meters (Bohour Al Shaer Al Arabi)*. Dar Al Fikr Al Lobnani, 1992.
- [19] AlDiwan. *Arabic Poetry Encyclopedia*, 2019.
- [20] Adab. *International Encyclopedia of Arabic Poetry*, 2018.