Recognition of Classical Arabic Poems

Abdulrahman Almuhareb **Ibrahim Alkharashi** **Lama AL Saud** **Haya Altuwaijri**
Computer Research Institute
KACST
Riyadh, Saudi Arabia
{muhareb, kharashi, lalsaud, htuwaijri}@kacst.edu.sa

Abstract

This work presents a novel method for recognizing and extracting classical Arabic poems found in textual sources. The method utilizes the basic classical Arabic poem features such as structure, rhyme, writing style, and word usage. The proposed method achieves a precision of 96.94% while keeping a high recall value at 92.24%. The method was also used to build a prototype search engine for classical Arabic poems.

1 Introduction

Searching for poetry instances on the web, as well as identifying and extracting them, is a challenging problem. Contributing to the difficulty are the following: creators of web content do not usually follow a fixed standard format when publishing poetry content; there is no special HTML tags that can be used to identify and format poetry content; and finally poetry content is usually intermixed with other content published on the web.

In this paper, a classical Arabic poetry recognition and extraction method has been proposed. The method utilizes poem features and writing styles to identify and isolate one or more poem text bodies in a given piece of text. As an implementation of the poetry recognition and extraction method, a prototype Arabic poetry search engine was developed.

The paper is organized as follows. In Section 2, the related works are briefly discussed. Section 3 gives a general overview of Arabic poems features. Section 4 discusses the methodology used to identify and extract poem content from a given text. It also presents the used evaluation method. In Section 5, we discuss the experimentation including the used dataset and results. A prototype implementation of the method is presented in Section 6 followed by conclusions and future work plans.

2 Related Work

To the best of our knowledge, this work[1]is the first attempt to explore the possibility for building an automated system for recognizing and extracting Arabic poems from a given piece of text. The most similar work related to this effort is the work that has been done independently by Tizhoosh and Dara (2006) and Tizhoosh et al. (2008). The objective of Tizhoosh and his colleagues was to define a method that can distinguish between poem and non-poem (prose) documents using text classification techniques such as naïve Bayes, decision trees, and neural networks. The classifiers were applied on poetic features such as rhyme, shape, rhythm, meter, and meaning.

Another related work is by Al-Zahrani and Elshafei (2010) who filed a patent application for inventing a system for Arabic poetry meter identification. Their invention is based on Al-khalil bin Ahmed theory on Arabic poetry meters from the 8th century. The invented system accepts spoken or written Arabic poems to identify and verify their poetic meters. The system also can be used to assist the user in interactively producing poems based on a chosen meter.

Work on poem processing has been also conducted on other topics such as poem style and meter classification, rhyme matching, poem generation and quality evaluation. For example, Yi

[1] Parts of this work are also presented in Patent Application No.: US 2012/0290602 A1.

Proceedings of the Second Workshop on Computational Linguistics for Literature, pages 9–16,
Atlanta, Georgia, June 14, 2013. ©2013 Association for Computational Linguistics

et al. (2004) used a technique based on term connection for poetry stylistics analysis. He et al. (2007) used Support Vector Machines to differentiate bold-and-unconstrained styles from graceful-and-restrained styles of poetry. Hamidi et al. (2009) proposed a meter classification system for Persian poems based on features that are extracted from uttered poems. Reddy and Knight (2011) proposed a language-independent method for rhyme scheme identification. Manurung (2004) and Netzer et al. (2009) proposed two poem generation methods using hill-climbing search and word associations norms, respectively. In a recent work, Kao and Jurafsky (2012) proposed a method to evaluate poem quality for contemporary English poetry. Their proposed method computes 16 features that describe poem style, imagery, and sentiment. Kao and Jurafsky's result showed that referencing concrete objects is the primary indicator for professional poetry.

3 Features of Classical Arabic Poems

Traditionally, Arabic poems have been used as a medium for recording historical events, transferring messages among tribes, glorifying tribe or oneself, or satirizing enemies. Classical Arabic poems are characterized by many features. Some of these features are common to poems written in other languages, and some are specific to Arabic poems. Features of classical Arabic poems have been established in the pre-Islamic era and remained almost unchanged until now. Variation for such features can be noticed in contemporary (Paoli, 2001) and Bedouin (Palva, 1993) poems. In this section, we describe the Arabic poetic features that have been utilized in this work.

3.1 Presence

Instances of classical Arabic poems, as well as other types of poems, can be found in all sorts of printed and electronic documents including books, newspapers, magazines, and websites. An instance of classical Arabic poems can represent a complete poem or a poem portion. A single document can contain several classical Arabic poem instances. Poems can occur in designated documents by themselves or intermixed with normal text. In addition, poems can be found in non-textual media including audios, videos and images.

In the web, Arabic poem instances can be found in designated websites[2]. Only-poem websites normally organize poems in categories and adapt a unified style format that is maintained for the entire website. Hence, poem instances found in such websites are almost carefully written and should contain fewer errors. However, instances found in other websites such as forums and blogs are written in all sorts of styles and may contain mistakes in the content, spelling, and formatting.

3.2 Structure

Classical Arabic poems are written as a set of verses. There is no limit on the number of verses in a poem. However, a typical poem contains between twenty and a hundred verses (Maling, 1973). Arabic poem verses are short in length, compared to lines in normal text, and of equivalent length. Each verse is divided into two halves called hemistiches which also are equivalent in length.

3.3 Meter

The meters of classical Arabic poetry were modeled by Al-Khalil bin Ahmed in the 8th century. Al-Khalil's system consists of 15 meters (Al-Akhfash, a student of Al-Khalil, added the 16th meter later). Each meter is described by an ordered set of consonants and vowels. Most classical Arabic poems can be encoded using these identified meters and those that can't be encoded are considered unmetrical. Meters' patterns are applied on the hemistich level and each hemistich in the same poem must follow the same meter.

3.4 Rhyme

Classical Arabic poems follow a very strict but simple rhyme model. In this model, the last letter of each verse in a given poem must be the same. If the last letter in the verse is a vowel, then the second last letter of each verse must be the same as well. There are three basic vowel sounds in Arabic. Each vowel sound has two versions: a long and a short version. Short vowels are written as diacritical marks below or above the letter that precedes them while long vowels are written as whole letters. The two versions of each basic vowel are considered equivalent for rhyme purposes. Table 1

[2] adab.com is an example for a dedicated website for Arabic poetry.

shows these vowel sets and other equivalent letters. These simple matching rules make rhyme detection in Arabic a much simpler task compared to English where different sets of letter combinations can signal the same rhyme (Tizhoosh & Dara 2006). On the other hand, the fact that, in modern Arabic writing, short vowels are ignored adds more challenges for the rhyme identification process. However, in poetry typesetting, typists tend not to omit short vowels especially for poems written in standard Arabic.

Table 1: Equivalent vowels and letters

Equivalent Vowels		**Equivalent Letters**	
/a/, /a:/	ا، ى، َ، ً	*ta*, *ta marbutah*	ت، ة
/u/, /u:/	و، وا، ُ، ٌ	*ha*, *ta marbutah*	ه، ة
/i/, /i:/	ي، ِ، ٍ		

Verse 1	**H1**	يقضون بالأمر عنها وهي غافلة
	H2	ما دار في فلك منها وفي قُطُبِ
Verse 2	**H1**	لو بيَّنت قطّ أمراً قبل موقعه
	H2	لم تُخْفِ ماحلَّ بالأوثان والصلُبِ
Verse 3	**H1**	فَتْحُ الفُتوحِ تَعَالَى أَنْ يُحيطَ بِهِ
	H2	نَظْمٌ مِن الشعْرِ أَوْ نَثْرٌ مِنَ الخُطَبِ
Verse 4	**H1**	فتحٌ تفتَّحُ أبوابُ السَّماءِ لهُ
	H2	وتبرزُ الأرضُ في أثوابها القُشُبِ

Figure 1: An example of classical Arabic poems with four verses written in Style 1. H1 and H2 are the first and second hemistich.

3.5 Writing Styles

There are three predominant writing styles of classical Arabic poems: (1) the poem is written in a single column with each verse in two rows; (2) the poem is written in a single column with each verse in two rows where the first half of each verse is written aligned to the right and the second half of each verse is aligned to the left; and (3) the poem is written such that each verse is written as two halves on the same row and separated by one or more punctuation marks or spaces. In some cases, this style can also be written without any separators and the end of the first half and the start of the second half have to be guessed by the reader. Figures 1 to 3 show examples of the three writing styles of classical Arabic poems.

Verse 1	**H1**	يقضون بالأمر عنها وهي غافلة
	H2	ما دار في فلك منها وفي قُطُبِ
Verse 2	**H1**	لو بيَّنت قطّ أمراً قبل موقعه
	H2	لم تُخْفِ ماحلَّ بالأوثان والصلُبِ
Verse 3	**H1**	فَتْحُ الفُتوحِ تَعَالَى أَنْ يُحيطَ بِهِ
	H2	نَظْمٌ مِن الشعْرِ أَوْ نَثْرٌ مِنَ الخُطَبِ
Verse 4	**H1**	فتحٌ تفتَّحُ أبوابُ السَّماءِ لهُ
	H2	وتبرزُ الأرضُ في أثوابها القُشُبِ

Figure 2: An example of classical Arabic poems with four verses written in Style 2.

	Hemistich 2		**Hemistich 1**
Verse 1	ما دار في فلك منـــها وفي قُطُـــبِ	**	يقضون بالأمر عنها وهي غافلة
Verse 2	لم تُخْفِ ماحلَّ بالأوثان والصلُـــبِ	**	لو بيَّنت قطّ أمراً قـــبل موقعـــه
Verse 3	نَظْمٌ مِن الشعْرِ أَوْ نَثْرٌ مِنَ الخُطَبِ	**	فَتْحُ الفُتوحِ تَعَالَى أَنْ يُحيـــطَ بِهِ
Verse 4	وتبرزُ الأرضُ في أثوابها القُشُـــبِ	**	فتحٌ تفتَّحُ أبـــوابُ السَّمـــاءِ لـــهُ

Figure 3: An example of classical Arabic poems with four verses written in Style 3.

3.6 Word Usage

It is very noticeable that classical Arabic poets tend not to use words repetitively in a given poem. To evaluate this observation, we analyzed a random set of 134 poem instances. We found duplicate start words (excluding common stop words) in 22% of the poems. Duplicate end words were found in 31% of the poems. However, the probability of encountering a verse with a duplicate start in the same poem is only 3% and 4% for a duplicate end word.

4 Method

The proposed method for standard Arabic poem recognition utilizes the poetic features described previously including structure, rhyme, writing style, and word usage. The meter feature was not literally used in the proposed method and may be used in a future work. The system operation is summarized by the flowchart shown in Figure 5 and described by the following steps:

1. Read input text line by line accepting only lines with reasonable size (e.g., lines of size between 2 and 20 words).
2. Collect consecutive lines that have equivalent length: compute the length of the line by counting the characters in the line. Lines are considered equivalent in length if the length difference is below a certain threshold (e.g., 40%, as has been used in the experiment discussed below).
3. Identify lines with separators to process Style 3 candidate verses. Separators are identified by searching for a set of white spaces or punctuations in the middle of the line between two halves. If identified, transform Style 3 to Style 1 shape for normalization.
4. Identify candidate rhymes at the end of each line.
5. Identify poems: searching for candidate poems in a consecutive list of candidate half-verses can produce several solutions based on rhyme. Select solution that produces poems with the maximum possible lengths. Figure 4 shows an example for a multiple solution case.
6. Repeat steps 1 to 5 until the end of the text body is reached.

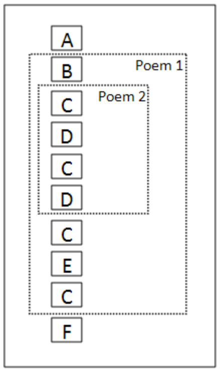


Figure 4: An example for multiple solutions based on rhyme. A list of 10 candidate half-verses indicated by their rhymes from A to F. Poem 1 starts at line 2 and ends at line 9 with 4 verses and rhyme C. Poem 2 starts at line 3 and ends at line 6 with 2 verses and rhyme D. The proposed method will select Poem 1 instead of Poem 2 since it has more verses.

Following these steps, the proposed method can recognize instances of classical Arabic poems of size at least two verses in any plain text. Detecting instances of a single verse is not covered in this work because the recognition process is only triggered by repetitive patterns that can't occur within single verse instances.

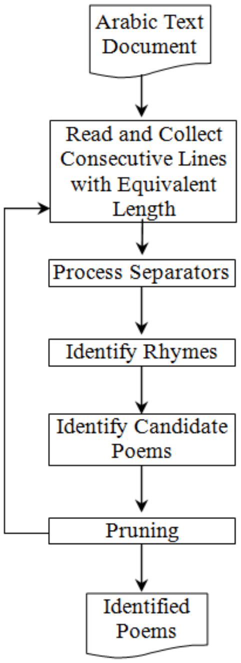


Figure 5: A flowchart of the proposed system for Arabic poems recognition.

4.1 Handling ill-formed cases

The proposed method can be applied on plain text from any source regardless of formatting and typing quality. Common formatting and typing mistakes and ambiguity are resolved as follows:

1. Mismatched and false separators: Mismatched separators occur when a set of candidate verses share the same rhyme but with different verse separators. Here, we treat the separators as if they were similar assuming that the separators were incorrectly typed. False separators, on the other hand, is identified when a set of candidate verses share the same rhyme and one or more verses were identified as having separators and the remaining verses have not. In this case, we ignore the identified separators assuming that these misidentified separators are just normal punctuation

marks. Figure 6 and 7 show real examples from the web for mismatched and false separators, respectively.

هذا الذي تعرف البطحاء وطاته **** والبيت يعرفه والحل والحرم

هذا ابن خير عباد الله كلهم **** هذا التقي النقي الطاهر العلم

اذا راته قريش قال قائلها *** الي مكارم هذا ينتهي الكرم

ينمي الي ذروة العز التي قصرت ** عن نيلها عرب الاسلام والعجم

Figure 6: An example of mismatched separators for a poem instance with four verses that share the same rhyme. The first two verses share the same separator while the third and the forth verses have similar but not exact separators.

مايقول الشعر .. من باله خلي

ومن نساه الهم لايطري القصيد

ماكتبت الشعر قصدي تزعلي

وان فرحتي فيه ما اقصد اكيد

حس يشعرني .. وضيقة تتجلي

ودار تسكني وانا عنها بعيد

Figure 7: An example of false separators for a poem instance with three verses that share the same rhyme. The first half of the first and third verses contain dots (..) at the middle of the line which can mistakenly be identified as separators.

2. Absence of short vowels: To treat missing short vowels in rhyme, we, recursively, assume the existence of the vowel if missing in a given verse and exists in a neighboring verse. Here, the last character in the former verse must match the second last character in the neighboring verse. Figure 8 shows an example of this case.

كالزهر في ترفٍ والبدر في شرفٍ *** و البحر في كرمٍ و الدهر في همم

كأنه و هو فردٌ من جلالته *** في عسكر حين تلقاه و في حشم

كأنما اللؤلؤ المكنون فى صدفٍ *** من معدني منطق منه و مبتسم

لا طيب يعدل تُرباً ضم أعظمهُ *** طوبى لمنتشقٍ منه و ملتثمِ

Figure 8: An example of short vowels absence for a poem instance with four verses. The first three verses neglect the short vowel *Kasrah* that exists at the end of the fourth verse.

3. Absence of separators: This case is triggered when encountering a set of consecutive lines sharing the same rhyme, and having line length in words that exceed half of the threshold for valid lines, and of course have no identifiable separators. The proposed remedy is to locate the closest whitespace to the center of each line and split the lines at those points and generate a verse of two hemistiches from each line. Figure 9 shows an example of this case.

مولاي صلي وسلم دائماً أبدا على حبيبك خير الخلق كلهم

أبان مولده عن طيب عنصره يا طيب مبتدأ منه ومختتم

يومٌ تفرَّس فيه الفرس أنهم قد أنذروا بحلول البؤْس والنقم

وبات إيوان كسرى وهو منصدعٌ كشمل أصحاب كسرى غير ملتئم

والنار خامدة الأنفاس من أسفٍ عليه والنهر ساهي العين من سدم

Figure 9: An example of absence of separators for a poem instance with five verses.

4.2 Pruning

Based on our observations during the development phase of the proposed method, it was noticeable that the robustness of the method correlates positively with the number of verses in the candidate poem. This is because with each additional verse the accumulated evidences are reconfirmed repetitively. This is not the case with few verses candidates. The probability of encountering a false matching rhyme for example with two or three verses is much higher. To resolve these cases and improve the precision of the proposed method, we introduce the following pruning tests to be applied only to short candidate poems:

1. Reject short candidate instances with low average number of words per half-verses. For example, using a threshold of 3 words.
2. Accept only short candidate instances that have at least two letters rhymes.
3. Reject short candidate instances when number of words per half-verse is not equivalent.
4. Reject short candidate instances with duplicate starting or ending words that exceed a threshold of 20%, for example.

4.3 Evaluation Measure

To evaluate the proposed method, we applied the F-measure (Swets, 1969) based on the precision and recall measures. Precision, as shown in Equation 1, is calculated by dividing the total number of the correct lines produced by the method over the total number of lines in the output. Given that our method processes the input data and generates output as half-verse per line. Recall, as shown in Equation 2, is computed similarly except that we divide over the model total number of correct lines. The model resembles the prefect solution for such input data.

$$Precision = \frac{System \quad Total \quad Number \quad of \quad Correct \quad Lines}{System \quad Total \quad Number \quad of \quad Lines} \quad (1)$$

$$Recall = \frac{System \quad Total \quad Number \quad of \quad Correct \quad Lines}{Model \quad Total \quad Number \quad of \quad Correct \quad Lines} \quad (2)$$

5 Experiment

5.1 Dataset

During the development phase of the method, we used several development datasets utilizing data drawn from the web. For evaluation purposes, we assembled a dataset using text from hundred randomly selected HTML web-pages. The set contains 50 HTML pages with classical Arabic poem instances (positive set) and 50 pages without poem instances (negative set). To select the positive set, we randomly chose 5 poets and searched Google and selected the first 10 pages that contain poem instances for each poet. The negative set was similarly chosen by selecting the first 50 pages that contain no poem instances for an arbitrary query. Text from the selected web-pages was converted to plain text using the Apache Tika toolkit[3] and saved in a single large text file. This resulted in a text file that contains about 23K non-empty lines including 161 classical Arabic poem instances having 4,740 half-verses.

5.2 Result

The poem dataset was used to evaluate the proposed poem recognition method. Figure 10 shows the results using five different pruning levels. The levels indicate the minimum number of verses for the pruning tests to be applied. Level 0 shows the performance without applying any of the pruning tests. The remaining levels show the results when the pruning is applied on candidates with at most two, three, and four verses, respectively. Level 4* is similar to Level 4 but here the fourth pruning test (duplicate words test) is applied on every candidate instance instead of only candidates with at most four verses.

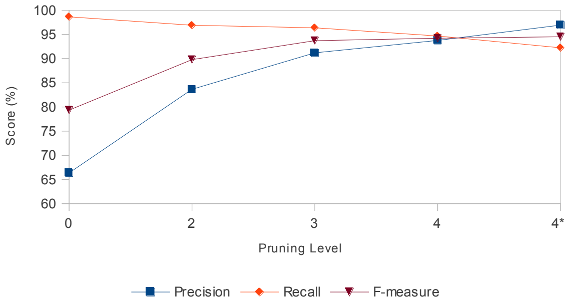


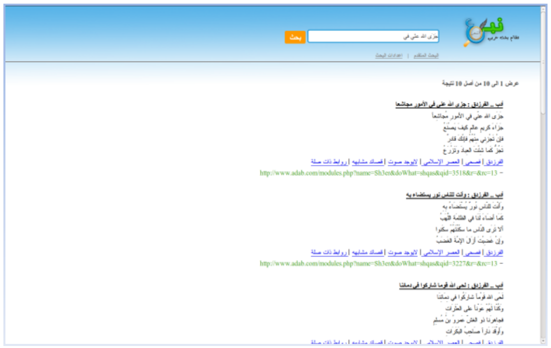
Figure 10: Evaluation results using five different pruning levels.

6 A Prototype Poem Search Engine

In order to assess the performance of the proposed poem recognition method in a real-life application, a prototype search engine for Arabic poems was implemented[4]. The search engine was built using the Apache Nutch web crawler and the Solr search engine to provide regular search engine services including crawling, parsing, and indexing. The HTML parsing plug-in in Nutch was extended using the proposed method to be able to recognize Arabic poems. Using this scenario, the search engine was successfully used to crawl a set of websites, identify all poem and non-poem instances, and index poem instances only. Figure 11 shows a snapshot of the search engine website.

[3] The Apache Tika toolkit can be downloaded from http://tika.apache.org/

[4] The Arabic poem prototype search engine can be accessed at http://naba.kacst.edu.sa



جزى الله عني في

عرض 1 الى 10 من أصل 10 نتيجة

أدب .. الفرزدق : جزى الله عني في الأمور مجاشعا

جَزى اللهُ عَني في الأُمورِ مُجاشِعاً
جَزاءَ كَريمٍ عالِمٍ كَيفَ يَصنَعُ
فَإِن تَجزِني مِنهُم فَإِنَّكَ قادِرٌ
تَجُزُّ كَما شِئتَ العِبادَ وَتَزرَعُ

الفرزدق | فصحى | العصر الإسلامي | لايوجد صوت | قصائد مشابهه | روابط ذات صلة
http://www.adab.com/modules.php?name=Sh3er&doWhat=shqas&qid=3518&r=&rc=13 -

أدب .. الفرزدق : وأنت للناس نور يستضاء به

وَأَنتَ لِلناسِ نورٌ يُستَضاءُ بِهِ
كَما أَضاءَ لَنا في الظُلمَةِ اللَهَبُ
أَلا تَرى الناسَ ما سَكَّنتَهُم سَكَنوا
وَإِن غَضِبتَ أَزالَ الإِمَّةَ الغَضَبُ

الفرزدق | فصحى | العصر الإسلامي | لايوجد صوت | قصائد مشابهه | روابط ذات صلة
http://www.adab.com/modules.php?name=Sh3er&doWhat=shqas&qid=3227&r=&rc=13 -

أدب .. الفرزدق : لحى الله قوما شاركوا في دمائنا

لَحى اللهُ قَوماً شارَكوا في دِمائِنا
وَكُنّا لَهُم عَوناً عَلى العَثَراتِ
فَجاهَرَنا ذو الغِشِّ عَمرُو بنُ مُسلِمٍ
وَأَوقَدَ ناراً صاحِبُ البَكَراتِ

الفرزدق | فصحى | العصر الإسلامي | لايوجد صوت | قصائد مشابهه | روابط ذات صلة

Figure 11: A snapshot of the prototype poem search engine

7 Conclusions and Future Work

In this paper, we proposed a method for classical Arabic poem recognition. The proposed method was able to identify Arabic poems in any unstructured text with a very high accuracy. The method utilizes the common features of classical Arabic poems such as structure, writing style, and rhyme; and employs them in the recognition process. A specialized search engine for classical Arabic poems was implemented as a prototype using the proposed method with promising results. For the future, we plan to enhance the method by introducing the well known meter model for classical Arabic poems. We would also like to extend the coverage of the method to include other types of Arabic poetry, namely contemporary Arabic. For the specialized search engine, we plan to add more features such as providing different search boundaries, for example, within a poem, a verse, or a hemistich. Moreover, we would like to find automatic ways to relate a poem to its poet.

Acknowledgments

The authors would like to thank Waleed Almutairi and Abdulelah Almubarak from KACST for their assistance in implementing the prototype poem search engine.

References

Al-Zahrani, A.K., Elshafei, M., 2010. Arabic poetry meter identification system and method. Patent Application US 2010/0185436.

Hamidi, S., Razzazi, F., Ghaemmaghami, M.P., 2009. Automatic Meter Classification in Persian Poetries Using Support Vector Machines. Presented at the IEEE International Symposium on Signal Processing and Information Technology (ISSPIT).

He, Z.-S., Liang, W.-T., Li, L.-Y., Tian, Y.-F., 2007. SVM-Based Classification Method for Poetry Style. Presented at the Sixth International Conference on Machine Learning and Cybernetics, Hong Kong.

Kao, J., Jurafsky, D., 2012. A Computational Analysis of Style, Affect, and Imagery in Contemporary Poetry, in: In Proceedings of the NAACL-HLT 2012 Workshop on Computational Linguistics for Literature. Montreal, Canada, pp. 8–17.

Maling, J., 1973. The theory of classical Arabic metrics (dissertation).

Manurung, H.M., 2004. An Evolutionary Algorithm Approach to Poetry Generation (PhD thesis).

Netzer, Y., Gabay, D., Goldberg, Y., Elhadad, M., 2009. Gaiku: Generating Haiku with Word Associations Norms. Presented at the Workshop on Computational Approaches to Linguistic Creativity (CALC ’09).

Palva, H., 1993. Metrical problems of the contemporary Bedouin Qasida: A linguistic approach. Asian Folklore Studies 52, 75–92.

Paoli, B., 2001. Meters and Formulas: The Case of Ancient Arabic Poetry. Belgian Journal of Linguistics 15, 113–136.

Reddy, S., Knight, K., 2011. Unsupervised Discovery of Rhyme Schemes. Presented at the 49th Annual Meeting of the Association for Computational Linguistics, Portland, Oregon, pp. 77–82.

Swets, J.A., 1969. Effectiveness of information retrieval methods. American Documentation 20, 72–89.

Tizhoosh, H.R., Dara, R.A., 2006. On Poem Recognition. Pattern Analysis and Applications, Springer 9, 325–338.

Tizhoosh, H.R., Sahba, F., Dara, R., 2008. Poetic Features for Poem Recognition: A Comparative Study. Journal of Pattern Recognition Research 3.

Yi, Y., He, Z.-S., Li, L.-Y., Yu, T., 2004. Studies on Traditional Chinese Poetry Style Identification. Presented at the Third International Conference on Machine Learning and Cybernetics, Shanghai.