

## RESEARCH ARTICLE



# Edit distance-based search approach for retrieving element-wise prosody/rhymes in Hindi-Urdu poetry

**OPEN ACCESS**

Received: 25.08.2020

Accepted: 24.09.2020

Published: 09.11.2020

Editor: Dr. Natarajan Gajendran

**Citation:** Khan MY, Rao MA, Wasi S, Minai TA, Raazi SMKR (2020) Edit distance-based search approach for retrieving element-wise prosody/rhymes in Hindi-Urdu poetry. Indian Journal of Science and Technology 13(39): 4189-4202. <https://doi.org/10.17485/IJST/v13i39.1489>

\* Corresponding author.

[yaseen.khan@jinnah.edu](mailto:yaseen.khan@jinnah.edu)

Funding: None

Competing Interests: None

**Copyright:** © 2020 Khan et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Published By Indian Society for Education and Environment ([iSee](https://www.isee.org/))

**ISSN**

Print: 0974-6846

Electronic: 0974-5645

**Muhammad Yaseen Khan<sup>1\*</sup>, Muhammad Adil Rao<sup>2</sup>, Shaukat Wasi<sup>1</sup>, Twaha Ahmed Minai<sup>2</sup>, Syed Muhammad Khaliq-ur-Rahman Raazi<sup>1</sup>**<sup>1</sup> Center for Language Computing, FoC, Mohammad Ali Jinnah University, Karachi, Pakistan<sup>2</sup> Department of Computer Science, DHA Suffa University, Karachi, Pakistan

## Abstract

**Background:** Prosody (rhyming words) is a connatural element of poetry, throughout its reach, across thousands of languages in the world. Since medieval era, the Indic poetry (principally the Hindi/Urdu poetry) has created an impactful flamboyance w.r.t the subjects, styles, and other creative aspects in poetry. Besides the message of heartfelt poetry, we see the Qafiya (i.e., rhyming words) is the core element, without which we may not consider anything Hindi/Urdu poetry but merely a piece of writing; alongside it, Radif (i.e., a phrasal suffix to qafiya) is also considered next to the intrinsic part in Ghazals. In this regard, the contributions of this paper are one–the development of an optimal technique for the prosodic (qafiya) suggestions/retrieval in Hindi/Urdu poetry; and two–the qafiya suggestions based on the attached subsequent radif. **Methods:** The work in this paper involves usage of a 13.46 M tokens tri-script corpus of poetry. Instead of phonetic value matching, the proposed methodology employs four different Edit Distances (i.e., Levenshtein, Damerau-Levenshtein, Jaro-Winkler, and Hamming distance) as the comparison measures for prosodic suggestions. **Findings:** The proposed work shows better results in comparison to ‘Qaafiya Dictionary’ powered by rekhta.org. Moreover, w.r.t the inter-metric similarity and running time Jaro-Winkler appears to be the most optimal algorithm for the rhyme suggestion, whereas the Levenshtein distance is the laziest technique. **Novelty/Applications:** This work benefits researchers of Indic natural language processing for lexical look-ups and analysis of creative literature, especially poetry.

**Keywords:** Natural language processing; information retrieval; poetry; prosody; Hindi; Urdu

## 1 Introduction

Throughout the history of the world and literature, the impression of poetry is beholding and profound. We may have lengthy texts and thick volumes, to ponder, on comparison of poetry and other sub-fields in literature, but Maugham<sup>(1)</sup> closes the debate by rightly

eulogizing poetry as:

*“The crown of literature is poetry. It is the end and aim. It is the sublimest activity of the human mind. It is the achievement of beauty”*

On comparison of poetry and other fields in arts and humanities, we see humans can learn to impart sermons; they can perfect their dancing and singing skills with practice, they can become excellent painters, but the art of saying poetry is divine. What is poetry? Rafi<sup>(2)</sup> cites Ibn Rachik’s definition as “بالقصد الشعر و هو الكلام موزون مقفى” i.e., poetry is a narration which is balanced, rhyming, and said unintentionally. Though in the aforementioned aspects, intention is arguable, but the other two i.e., balance (w.r.t poetic meters) and prosody/rhyme are quite indispensable. Crowe et al.<sup>(3)</sup> discussed the importance of prosody as:

*“...metered language is a double medium, with two systems of effect—a steady music... [and] a sense of a ritualistic occasion.”*

Al-Beruni<sup>(4)</sup> maintained that aside from the fact that it (prosody) makes memorizing the poem easier,

*“the soul yearns for anything that has symmetry and regularity and feels disgusted for that which has not regularity”.*

On rhyming, Rampuri<sup>(5)</sup> translates a famous critique made by Avicenna “عز متقى نہیں وہ ہمارے نزدیک شعر نہیں” (it is not poetry to us which is not rhyming). Verily, the musicality in poetry increases if it is balanced on the metrical grounds, accompanying the rhythm with properly addressed rhymes. Hence, the prosody is collectively considered as the core which defines the whole of poetry, irrespective of languages, in all genres of poems.

The fields of Computational Linguistics (CL) and Natural Language Processing (NLP) focus on the analysis and development of tools and techniques for the human languages, i.e., often called Natural Languages (NL)<sup>(6)</sup>. The research in these fields has contributed much phenomenal work in terms of information retrieval<sup>(7,8)</sup>, language translation/ transliteration<sup>(9,10)</sup>, text generation<sup>(11,12)</sup>, and text/document classification<sup>(13,14)</sup>. However, the contributions, as mentioned earlier, are to name a few examples from the interminable list of NLP research and applications. These researches generally revolve around the data, which broadly engages prose in terms of articles, blogs, stories, customer reviews, and feeds from social-media streams. However, all of these, irrespective of types, lack poeticism, or show least of it. Hence, we maintained that substantial work has now been done in prose, in comparison to a quite small work for poetry.

For the novice and aspiring creative writers, it is very difficult to remember the whole bunch of rhyming words, nor the conventional dictionary lookup helps them manually. This paper brings an interdisciplinary study that embraces the literary studies and computational aspects of dealing with NLPs and tries to contribute to the analysis of poetry for building a system that helps to retrieve and suggest appropriate rhymes/prosodies. The work is done for the Indic languages, such as Hindi and/or Urdu, which are morphologically rich<sup>(15)</sup>, having astounding and vivid contributions in the literature<sup>(16–18)</sup>, and being dominant in the list of world languages with the number of native speakers<sup>(19)</sup>; however, have shown a very little work in the computational treatment of the text. Hence, for this reason, these languages are, nowadays, termed as resource-poor and scared-source<sup>(20,21)</sup>. So, to do a little, this paper contributes to the following main points:

- Presents an edit-distance based technique for prosody (rhyming words, i.e., qafiya) searching and suggestion. In this regard, four well-known distance metrics have been used, i.e., Levenshtein, Damerau–Levenshtein, Jaro–Winkler, and Hamming distance.
- The resulting list of prosodies is compared in a pair-wise manner to evaluate the inter-similarity of the edit-distances.
- Further, for a thematic prosody suggestion—i.e., in the case of ghazals—radif-based qafiya suggestion, a probabilistic approach based on language modeling technique is also presented.
- Lastly, a running time analysis is made for both of these techniques.

The rest of the paper is organized as the §2 describes the historical background of Hindi/Urdu as language, the influence of other languages on it, mutual connection, and bonding therein, a brief survey related to the poetry system, the importance of prosody in Indic poetry. §3 shares the details of data and proposed methodology, followed by a discussion on results in §4, and conclusions in §5 in the end.

## 2 Background and Literature Review

### 2.1 Urdu and Hindi as a language

Hindi and Urdu are the most prominent languages in the Indic branch of the language family. On colloquial grounds<sup>(22)</sup>, both languages are mutually intelligible<sup>(23)</sup>; hence can be considered as the same language under the name of ‘Hindustani’ or ‘Hindi-Urdu’ (henceforth HU)<sup>(24)</sup>. Historically, Urdu is evolved in medieval India as a creole language<sup>(20)</sup>, with the influence of many languages: preliminary Persian, Arabic, and Turkic languages<sup>(15)</sup>; and lately, under the colonial rule, English

and Portuguese<sup>(25)</sup>. Today, the Urdu and Hindi languages, respectively, are the Persianised and Sanskritised registers of the Hindustani language<sup>(24,26)</sup>. The quantified similarity between the two languages is found at a greater extent; both on parts-of-speech wise and phonetic/articulatory features-based lexical similarity<sup>(27,28)</sup>. The Hindi-Urdu is a victim of digraphia, such as Urdu is supposed to be written with the modified Perso-Arabic, whereas Hindi is written in Devanagari scripts. Together, both languages appear at the 3<sup>rd</sup> place, as most spoken languages of the world, with 329.1M native speakers and 697.4M total speakers<sup>(19)</sup>.

## 2.2 Poetry and prosody in Hindi-Urdu: A brief survey

Likewise the tradition in various languages, Hindi-Urdu has the compendium of the meter<sup>(5,29)</sup>. The development of the Hindi-Urdu poetry system can be taken into account under the influence of ancient Indian languages, i.e., Vedic and Panini-Sanskrit, and Prakrits. These languages spanned over 2500 years, mainly divided into two ages, i.e., Old Indo-Aryan ca. 1500–300 BCE; and Middle Indo-Aryan (MIA) ca. 300 BCE–1500 CE<sup>(30)</sup>. Further, all varieties of MIA languages that evolved from Sanskrit are subsumed under the term Prakrit<sup>(31)</sup>. Alongside the Sanskrit, Persian and Arabic languages, have played the most instrumental role in the development of the poetry system in the 18–20<sup>th</sup> century<sup>(32,33)</sup>. We suspect that it would become an extensive debate on the comparison of Indic and non-Indic poetry symmetries and characteristics. Thus, we just summarized the points that are essential w.r.t the rhythmic structure and meter, and broadly brought into the practice of Hindi-Urdu poetry since medieval era (the relevant details of ancient Indic prosody—aligned with the Vedic and Paninian Sanskrit—are given in footnotes<sup>1</sup>). We maintain a conclusion by accounting the rule of Ghazal<sup>2</sup> ([γəzəl], غزل / गज़ल) and relevant Perso-Arabic genres of poetry in the medieval and post-medieval era in India. Also, under the influence of Perso-Arabic culture, classical Hindi-Urdu poetry principally followed Arabic poetry meters<sup>(2,5)</sup>.

Along with the meter, prosody in Hindi-Urdu poetry includes Qafiya ([q a:fi:ɑ:], قافیہ / قافیہ; pl. Qawafee [qəvɑ:fi:] قوافی / قوافی) that are the rhyming words; and Radif ([rəð i:f], رادیف / رادیف) which is the phrasal suffix to qafiya are intrinsic parts of the ghazal. However, the qafiya equally qualifies the concept of rhyming words, *tukant* तुकांत [t̪ u.kā:nt̪], in Sanskrit or Hindi poetry, which heavily draws on Sanskrit vocabulary. Similarly, radif equivalently corresponds to the *charnant* चरणांत [t̪ə.rə ŋ̃ ã:nt̪] in the same context. As per rule, the radif (preceded by the qafiya) has to appear at the end of every second hemstitch of ghazal<sup>(5,37)</sup>. The radif in any ghazal brings focus to the thematic mood of poetry and sets the philosophical disposition. Furthermore, the radif not only entices the alluring poetic characteristic but also brings challenging criteria (for poets) to end every couplet in the same contextual ambience. On an additional note, the usage of radif is found very common in Hindi-Urdu poetry and often observed in Persian and Turkic poetry. However, the usage of radif is absent in Arabic poetry.

It should be kept in mind that the Urdu language has the quality of syntactical construction of two words through the *Izāfats*; which are used exclusively in Hindi-Urdu poetry. This Perso-Arabic stylistic construction of words replaces the post-position का/का [ka:] and its different derived morphologies (کے/के [kæ:]; and کی/कि [ki:]) through reordering the surrounding words to bring rhythmic beauty in text/poetry. While the qafiya/rhyme will relate to the final word of the (syntactically constructed) compound word. The reordering is performed on two (or more) words in the following three manners (though the following text appears complex but in reality, it is per se sublime characteristics of Hindi-Urdu poetry):

1. Zer-e-Izafat (ZI): Appends a diacritic symbol ' (zer). For example., محری امید / سहर کی उम्मीद (/sahar ki umīd/; [literal] 'hope of dawn') will be rewritten as امید سحر / उम्मीद-ए सहर /umīd-e-sahar/ by appending zer at the end of first word امید. The transliteration systems for romanizing Urdu-Hindi use ZI in two different manners, as we see '-i' and '-e'<sup>(38)</sup>, we count both of them correct and valid for interchangeable use. For Devanagari script, we can use the vowel matra ए or its modified inherent form े. ZI also relates adjectives to the surrounding nouns; however, adjectives (with ZI) come after nouns. For e.g.; زلف سیاہ / जुल्फ-ए सयाह/zulf-e-siyah/ (black tresses); چشم نم / चश्म-ए नम/chashm-e-num/ (damp eye).
2. Hamza-e-Izafat (HI): Transforms the final alphabets if they are either 'ے' (bari ye), 'ی' (choti ye) or 'ہ' (choti he) with hamzah 'ء' to render as 'ئے' and 'ئی' and 'ہے' respectively. For example, the text وفا کا وعدہ / वफा का वादा /vafā ka vadā/

<sup>1</sup>Acharya Pingala (Indian mathematician and grammarian, c. 3<sup>rd</sup>/2<sup>nd</sup> century BCE) contributed the earliest work, *Chandas*, a disquisition for classical Sanskrit prosody. This work is named after him as Pingala shastra (पिङ्गल शास्त्र) and also known as Chand shastra (चन्द शास्त्र)<sup>(34)</sup>. Similarly, Kedara Bhatta (Indian prosodist, c. 950-1050 CE) made the most popular contribution to Sanskrit prosody<sup>(35)</sup>, i.e., Vrittaraṅgana [vṛttaraṅgana], which is comparatively largest repertoire to any other prosodic metrical tradition.

<sup>2</sup>Ghazal originated in Arabia, led to maturity in Persia, and found in the most hospitable state in the Indian subcontinent<sup>(36)</sup>. It is considered as a metered genre of poetry. These well-defined meters, known as بحر [bæhr], are used as a standard to judge the balance of couplets in ghazal<sup>(5,29)</sup>. Technical definition of its composition can be realized as the set of couplets اشعار / अशआर, [ə:ʔɑ:r ] (sing. شعر / शेर, [ʃəʔr ] with two hemstitches (مصرع / मिस्रे, [ misrəʔe ]; sing. مصرعہ / मिस्रा, [ misrəʔ ]) which are paired in each.

([literal] ‘promise of love/loyalty’) can be rewritten as **وفا** / **واदा-ए** वफ़ा /vadā-e-vafā/; and **कालाश की वादी** / **कलश** कि वादी /kalāsh ki wādi/ ([literal] valley of Kalash) will become **वादी कालाश** / **वादा-ए** कलश /wādi-e-kalāsh/. Another transformation is made explicitly with the alphabet ‘**ز**’ if the final alphabets are either ‘**ا**’ (alif) or ‘**و**’ (wao)<sup>(39)</sup>. For example, **دل کی صدا** / **दिल** कि **सदा** /dil ki sadā/ ([literal] voice of heart) will become **دل** / **صدائے** / **सदा-ए** **दिल** /sadā-e-dil/; and similarly, **قاتل کا بازو** / **कतील** का **बाजू** /qātil ka bāzu/ ([literal] arm of the killer) will be **बाजू** / **बाजू-ए** **कतील** /bāzu-e-qātil/. Likewise ZI, for Devanagari script HI employs vowel matra ए.

3. Wao-e-Izafat (WI): Inserts wao ‘**و**’ (wao) as an instead-shortened form of **اور** / **और** /aur/ (means and). For example, instead of writing **خواب اور خیال** / **खाब और खयाल** /khaab aur khayal/, ([literal] dreams and thought) we can rewrite it as **خواب و خیال** / **खा-ओ-खयाल** /khaab-o-khayal/. For Devanagari script, the vowel matra औ is used.

The example of qawafee and radif is elaborated in table 1, which renders the popular ghazal of famous Urdu poet Muhammad ‘Iqbal’. These are four couplets, where every couplet contains a qafiya (marked with red text). The detailed review of the rhyming words are, for example, in first and third couplets: **بے گانہ وار** / **बेगाना-वार** /begana-var/ (adjective; ‘apathetic’), **बार बार** / **baar baar** /; (adverb, ‘repeatedly’), and **انتظار** / **तिज़ार** [intizar]; (verb, ‘wait’). While the other examples of qawafee as compound words (with ZI) are given in second and fourth couplets. Although these are compound words, as mentioned earlier we have to look into the final word for rhyming. For example, in the **میشال شرار** / **मिसाल-ए-शरार** [misal-e-sharar] (adjective, ‘resembling sparks’); **ہستی ناپائیدار** / **हस्ती-ना-पाएदार** [hasti-e-na-pa.edar] (adjective, ‘transitory life’); and **نقش پایے یار نقش** / **नक्श-ए-कफ-ए-पा-ए-यार** [naqsh-e-kaf-e-pa-e-yar] (adjective, ‘decoration of beloved’s feet’) the words subjected to rhyme are **شرار** / **शरार** [sharar] (noun, ‘spark’); **نا-पाएदार** / **na-pā-e-dar** (adjective, ‘transient’), and **یار** / **यार** [yar] (noun, ‘friend’). Similarly, in the case of the presented example, the radif is **دیکھ** / **देख** [dekh] (verb, ‘look’ or ‘see’); shown with the blue colour). Thus, we can observe the intent of the poet throughout the ghazal that he is saying to look or observe about different topics in each couplet.

**Table 1.** Prosody limned in a famous ghazal said by ‘Iqbal’<sup>(40)</sup>. Red and blue colors mark qafiya and radif respectively. The English translation is excerpted from ‘Khalil’<sup>(41)</sup> (40).

Hindi (Scripted in Devanagari)	Romanized Transliteration
गुलज़ार-ए-हस्त-ओ-बूद न बेगाना-वार देख है देखने की चीज़ इसे बार बार देख	gulzār-e-hast-o-būd na begāna-vār dekh hai dekhne kī chiiz ise baar baar dekh
आया है तू जहाँ में मिसाल-ए-शरार देख दम दे न जाए हस्ती-ना-पाएदार देख	aayā hai tū jahāñ meñ misāl-e-sharār dekh dam de na jaa.e hasti-e-nā-pā.edār dekh
माना कि तेरी दीद के क़ाबिल नहीं हूँ मैं तू मेरा शौक़ देख मिरा इंतज़ार देख	maanā ki terī diid ke qābil nahīñ huuñ maiñ tū merā shauq dekh mirā intizār dekh
खोली हैं ज़ौक़-ए-दीद ने आँखें तिरी अगर हर रहगुजर में नक्श-ए-कफ़-ए-पा-ए-यार देख	kholī haiñ zauq-e-did ne āñkheñ tirī agar har rahguzar meñ naqsh-e-kaf-e-pā-e-yār dekh
Urdu (Scripted in Nastaliq)	English Translation
گلزار ہست و بود نہ بے گانہ وار دیکھ سے دیکھنے کا جہاں سے بار بار دیکھ آئے تو جہاں سے، میسال، شرار دیکھ دم دے نہ جائے ہستی ناپائیدار دیکھ مانا کہ تیری دید کے قابل نہ ہوں، ہستی، تو مراد شوق، دیکھ کہ مانتظار دیکھ کھول، ہستی، ذوق، دید نے آنکھوں میں، اگر ہر رہ گزر میں نقش کف پائے یار دیکھ	“Do not look at the garden of existence like a stranger It is a thing worth looking at, look at it repeatedly  You have come into the world like a spark, beware Lest your ephemeral life may end suddenly, beware  Granted that I am not worthy of your sight You should look: at my zeal, at my perseverance  If your eyes’ve been opened by the longing for sight Look for the foot prints of the beloved in every lane”

The radif in ghazal is of a single word; however, radif with over single word do exist in Hindi-Urdu poetry. For better comprehension, consider a line from the famous ghazal of ‘Momin’<sup>(42)</sup>, where the text, likewise before, are respectively in red and blue colour for qafiya/rhyming word and radif.

وہ جو ہم میں تم قرار تھا تمہیں یاد ہو کہ نہ یاد ہو  
 वो जो ह्य मिण तुम मिण करार था तुम्हेण इआद ही के ना याद हो

/vo jo ham meñ tum meñ qarār thā tumheñ yaad ho ki na yaad ho/  
 Translation of radif: “[perhaps] you remember [it] or not”.

## 2.3 Related Work and Research Gap

The computational treatment of the poetic prosody for the Hindi-Urdu language showed an inconsequential contribution. In other ways, speech prosody has got comparatively more focus than the poetic prosody. We assume that the coverage of speech prosody is extraneous for this paper; however, the most recent work on the phonological and phonetic aspects in the Urdu are contributed by Hussain et al. (43,44) and Jabeen et al. (45–47) etc.

The work on the phonological aspects of Urdu poetry by comparing it to moraic weight was pursued by Hussain (48). It also evaluates the possibility of mapping Urdu poetry on the rules that mask Urdu prose. In the same regard, there exist many detailed texts, (e.g., Rampuri (5), Pritchett and Khaliq (49), Abidi (29), and Rafi (2) etc.) for the comprehending Urdu poetry, prosody, and metrical rules; however, they suffuse the linguistically-theoretical aspects of the analysis, yet the computational approaches are missing.

The computational treatment of Urdu poetry is mainly offered at two forums: Aruuz.com (50) and Rekhta.org (51). Both of these are web-based solutions. The following two subsequent paragraphs are discussing both of these forums respectively.

Aruuz (50) offers the metrical tagging of Urdu poems; which to mean technically is the splitting of a couplet for syntactic parsing as per the grammatical rules/meters. So, on inputting (at max.) a total of 4 hemstitches in Nastalique script Aruuz finds the closest meter of the couplet and tags words therein accordingly with the metrical units. Since version 2.0, Aruuz started mentioning the fluency score of the couplet. Figure 1 shows the output of Aruuz (upon inputting 2 opening couplets of a ghazal penned by veteran Indian lyricist Javed Akhter). Aruuz describes the ghazal is said to be written in ‘Behar-e-Hindi’ with a fluency score of 8. Other than the aforementioned task, Aruuz also suggests similar works that are composed in the same meter.

Rekhta (51) is making a tremendous effort by maintaining the repository of Hindi-Urdu literature, which covers poetry in almost all genres and preservation of classical Urdu text in the form of e-books. It also offers an online lookup tool, namely, ‘Qafiya Dictionary’ (QD) for qafiya searching. On giving a word, Rekhta shows the rhyming words in terms of Exact and Close categories. However, the technique applied for searching and retrieval of rhyming words is unknown.

Besides the two of the aforementioned web-forums, we find no work done particularly for retrieval of rhyming words. Thus, we assume the work presented in this paper is novel per se. Forbye it, authors humbly maintained that the proposed methodology is bit straightforward and produces a baseline result, which at this moment cannot be compared with any of the previous work. Though, we have tried to compare our results with QD and found that not only QD lacks the entry of many words but also it lags behinds to show appropriate rhyming words.

## 3 Material and Method

### 3.1 Data

In recent times, the use of the resources available on the Internet has become a common practice. Following the same fashion, the dataset for this research work is also prepared from the websites designed for Urdu poetry. We opt Rekhta as the primary source for data scraping and further the construction of parallel corpus. As mentioned in §2.3, the website is not only a leading repository of Urdu poetry; it also offers poetry in various scripts to help the readers of Urdu poetry in Nastalique/modified Perso-Arabic; Devanagari and Roman Urdu scripts. We would also like to mention that the scraper is built with BeautifulSoup-4 (52), which is a well-known Python package for this sort of task.

A consolidated overview of the scraped data used in this research work is listed in table 2. Including ghazals, it shows a variety of all available poetry genres at Rekhta, such as nazam, dohe, qita, marsiya, mukhammas, manqabat, masnavi, naat, qasida, sehra, salaam, and rubai. Since, not every genre of poetry has a set of paired hemstitch (i.e., a couple), therefore, the count of hemstitches is reported. The statistics for the compounds are calculated with the list of tri-grams; such that the whole corpus is processed to workout list of n-grams (for word and characters both  $n \in 1 \dots 4$ ; (for Hindi) and ‘e’ (for Roman-Urdu) to form a ZI with surrounding tokens. Similarly, through the list of bi-grams of Urdu sub-set, we count every item where the first entity ends with the diacritic ‘zer’. Mathematically, suppose  $nW^g$  is the list of word n-grams for language g; then,

$$\text{Hindi ZI} = \|\{x \mid x \in \mathbf{3W}^{\text{hindi}} \text{ if } x[1] = \text{‘ए’}\}\| \quad (1)$$

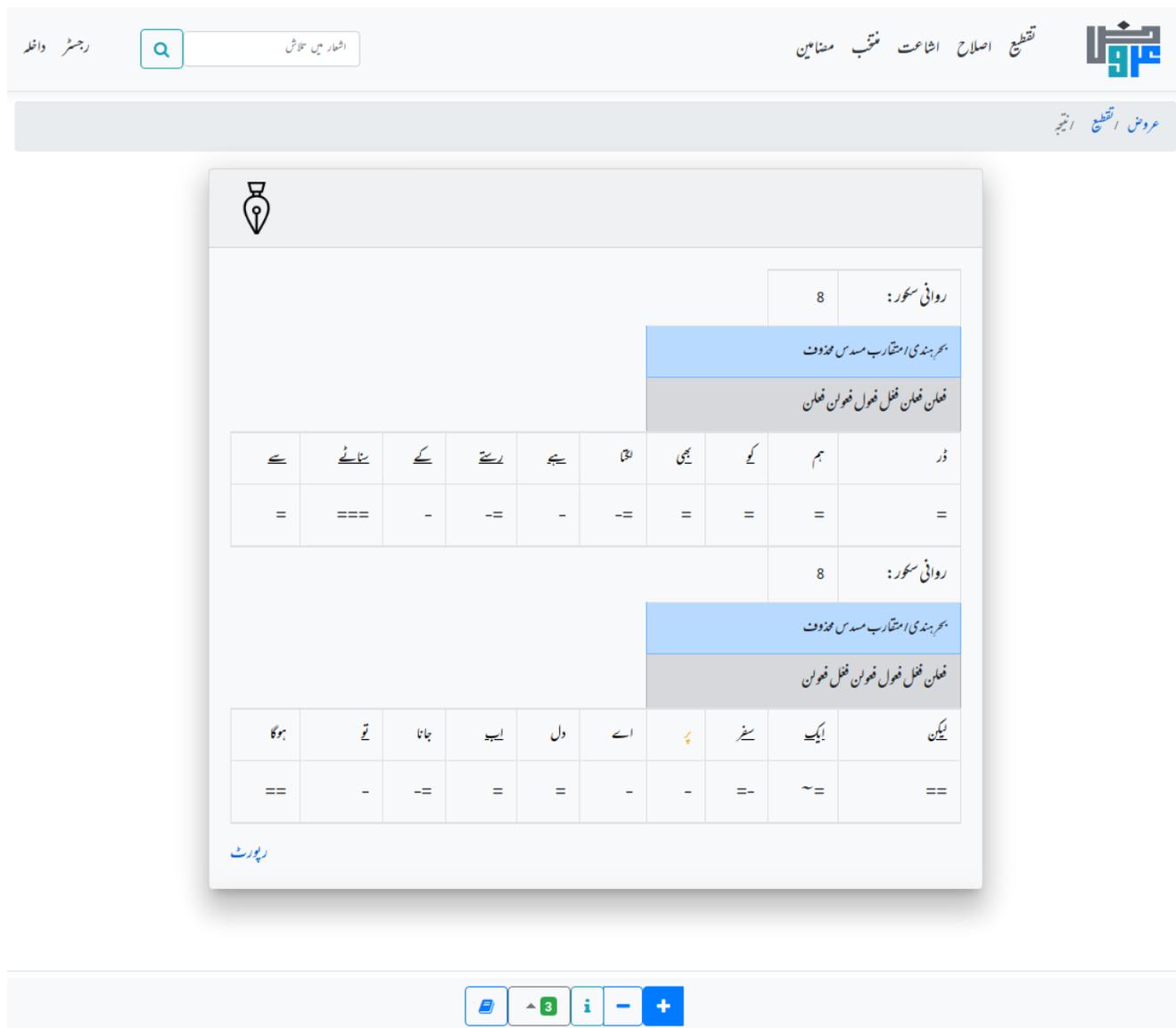


Fig 1. Screenshot of output rendered by Aruuz.com on the ghazal written by Javed Akhtar.

Table 2. Statistics of the data (tokens) w.r.t writing script.

Linguistic Features	Count		
	Urdu	Hindi	Roman Urdu
Total Hemistichs	619,646	583,703	518,573
Total Tokens	4.88 M	4.30 M	4.28 M
Distinct Tokens	50, 550	49, 469	43, 448
Distinct compounds with Zer-e-Izafat (ZI)	74, 985	54, 860	51, 889
Distinct compounds with Wao-e-Izafat (WI)	8,537	6,389	5,629
Distinct compounds with ZI & WI both	20,450	14,962	14,757
Distinct combinations of final character uni-gram	111	98	61
Distinct combinations of final character bi-gram	1,688	1,524	681
Distinct combinations of final character tri-gram	10,690	9,477	4,325

$$\text{Roman-Urdu ZI} = \|\{x \mid x \in 3\mathbf{W}^{\text{roman-urdu}} \text{ if } x[1] = 'e' \}\| \tag{2}$$

$$\text{Urdu ZI} = \|\{x \mid x \in 2\mathbf{W}^{\text{urdu}} \text{ if } x[0][l-1] = 'i' \}\|; \quad l = \|x[0]\| \tag{3}$$

The counting for WI undergoes the same process for Urdu, Hindi and Roman Urdu. The second entity in the tri-gram list is checked for ‘ओ’ (for Hindi), ‘o’ (for Roman-Urdu), and for Urdu, we use the list of tri-grams where the second entity is ‘;’ These alphabets can be substituted in the criteria mentioned in equation 1 with respective languages to get the count of WI. Moreover, for Hindi and Roman-Urdu, the number of compounds having both ZI and WI are counted where the entity at the odd indices of every tri-gram is the alphabet that delimits WI or ZI. Counting compounds having both ZI and WI for Urdu in Nastalique script is a complex task; authors find two ways of dealing with it: a) using tri-grams such that the last character of the first entity should be zer and the last entity to be ‘;’, and b) first entity should be ‘;’ followed by the last character to be zer in the second entity of tri-gram. Another insight into the dataset w.r.t combination of final character n-grams that the dataset seldom shows irregularities, such as after processing, we saw few non-Perso-Arabic alphabets, numbers, and symbols along with Perso-Arabic alphabets. However, n-grams with these irregularities are removed before employing the proposed methodology.

---

### Algorithm 1. Data Preprocessing

---

```

0: procedure PreProcess(C)           ◁ Let C be the corpus/set of poems.
1:   U ← D ← R ← { be an empty set }  ◁ Such that the symbols U, D and
   R respectively denote empty set for Nastalique, Devanagari, and Roman scripts.
2:   for all poem P ∈ C do
3:     for all  $i \in \{0 \dots \|\mathbf{P}[u] - 1\|\}$  do
4:       U ← U ∪ { set of tokenized items in  $\mathbf{P}[u][i]$  }
5:       D ← D ∪ { set of tokenized items in  $\mathbf{P}[d][i]$  }
6:       R ← R ∪ { set of tokenized items in  $\mathbf{P}[r][i]$  }
7:   G ← { final character  $n$ -gram of  $w \mid w \in \mathbf{U}, \forall n \in \{1, \dots, 3\}$  }
8:   I ← {  $g \{w \mid w \in \mathbf{U}; \text{ if final characater } \|g\| \text{-gram of } w = g\} \mid g \in \mathbf{G}$  }
9:   return I

```

---

While scraping poems from Rekhta.org, we saved every poem  $\mathbb{P}$ , with all three scripts (i.e., Nastalique, Devanagari, and Roman-Urdu) as a dictionary in the separate JSON files. Thus, let  $\mathbf{H}_U, \mathbf{H}_R,$  and  $\mathbf{H}_D$ , be the list of hemstitches in  $\mathbb{P}$  respectively corresponding to the Nastalique, Romanized, and Devanagari scripts; hence more formally  $\mathbf{P} \leftarrow \{u \rightarrow \mathbf{H}_U, r \rightarrow \mathbf{H}_R, d \rightarrow \mathbf{H}_D\}$ . The whole corpus  $\mathbb{C}$  is a collection of poems such that  $\mathbf{C} \leftarrow \{\mathbf{P}_1, \dots, \mathbf{P}_n\}$ , where  $n$  is the total number of poems. For  $\mathbb{C}$ , algorithm 1 shows the preprocessing steps in which we worked out a dictionary  $\mathbb{I}$  such as the character  $n$ -gram is the key against which a set of such words is retained that end on the respective character  $n$ -gram. Step 7 and 8 in algorithm 1 show the work with the Urdu vocabulary. However, in practice, these steps were also executed for the Devanagari and Roman-Urdu subsets as well.

### 3.2 Proposed methodology

This section is particularly divided into 2 subsections. Each one is dedicatedly describing the proposed methodology such that in §3.2.1 and §3.2.2 simple qafiya and radif-based qafiya searching is presented.

#### 3.2.1 Top-t Qafiya Search (QS)

After the preprocessing, retrieval of the top-t relevant words is made. In result, a list of qawafee, sorted in the ascending order of distances is brought. For calculating distance between two words, we used widely employed distance metrics such

as Hamming (Ham.)<sup>(53)</sup>, Levenshtein (Lev.)<sup>(54)</sup>, Damerau–Levenshtein (D–L)<sup>(55)</sup>, and Jaro–Winkler (J–W)<sup>(56)</sup> distance. Since these algorithms are quite well-known and well understood, therefore, we skip the technical details. Algorithm 2 shows the procedure for QS. In step 3 and 4, the symbol \ indicates set minus operation. In steps 5–7, one of the aforementioned metrics is shown with function  $\Delta(c, \mathbf{W})$  where  $\mathbb{C}$  and  $\mathbf{W}$  are the candidate and target words. In line 11, the set operator  $\widehat{\phantom{x}}$  denotes the sequential concatenating of two tuples, for example, suppose  $\lambda_1 = \langle x_1, x_2, x_3 \rangle$  and  $\lambda_2 = \langle y_1, y_3, y_2 \rangle$  then  $\lambda_1 \widehat{\lambda}_2 = \langle x_1, x_2, x_3, y_1, y_3, y_2 \rangle$ .

---

**Algorithm 2. Qafiya Searching**

---

0. **procedure**  $QS(\mathbf{I}, \mathbf{W}, t)$        $\triangleleft$   $\mathbf{I}$  be the preprocessed dictionary (see algorithm 1),  $\mathbf{W}$  be the target word; and  $t$  be the number of top suggestions.
  1.      $w_1, w_2, w_3 \leftarrow$  be the respective final character uni-gram, bi-gram and tri-gram of the word  $\mathbf{W}$ .
  2.      $\mathbf{L}_3 \leftarrow \mathbf{I}[w_3]$                        $\triangleleft$  be the set of words that end on  $w_3$ .
  3.      $\mathbf{L}_2 \leftarrow \mathbf{I}[w_2] \setminus \mathbf{L}_3$                $\triangleleft$  be the set of words that ends on  $w_2$  excluding words  $\in \mathbf{L}_3$ .
  4.      $\mathbf{L}_1 \leftarrow \mathbf{I}[w_1] \setminus (\mathbf{L}_2 \cup \mathbf{L}_3)$        $\triangleleft$  be the set of words that ends on  $w_1$  excluding words  $\in \mathbf{L}_3 \cup \mathbf{L}_2$ .
  5.      $\mathbf{S}_1 \leftarrow \{c \rightarrow \Delta(c, \mathbf{W}) \mid c \in \mathbf{L}_1\}$   $\triangleleft$  be the score dictionary w.r.t words in  $\mathbf{L}_1$ ; key is candidate word and value is  $\Delta(\cdot)$ .
  6.      $\mathbf{S}_2 \leftarrow \{c \rightarrow \Delta(c, \mathbf{W}) \mid c \in \mathbf{L}_2\}$   $\triangleleft$  be the score dictionary w.r.t words in  $\mathbf{L}_2$ ; key is candidate word and value is  $\Delta(\cdot)$ .
  7.      $\mathbf{S}_3 \leftarrow \{c \rightarrow \Delta(c, \mathbf{W}) \mid c \in \mathbf{L}_3\}$   $\triangleleft$  be the score dictionary w.r.t words in  $\mathbf{L}_3$ ; key is candidate word and value is  $\Delta(\cdot)$ .
  8.      $\mathbf{S}'_1 \leftarrow$   $\langle$  be the list of words, sorted w.r.t values in  $\mathbf{S}_1$   $\rangle$
  9.      $\mathbf{S}'_2 \leftarrow$   $\langle$  be the list of words, sorted w.r.t values in  $\mathbf{S}_2$   $\rangle$
  10.     $\mathbf{S}'_3 \leftarrow$   $\langle$  be the list of words, sorted w.r.t values in  $\mathbf{S}_3$   $\rangle$
  11.     $\mathbf{R} \leftarrow \langle \mathbf{S}'_3 \widehat{\mathbf{S}'_2} \widehat{\mathbf{S}'_1} \rangle$                $\triangleleft$  be the consolidated list of resultant qawafee/ rhyming words, merged sequentially.
  12.    **return**  $\mathbf{R}[0 : t]$  if  $t \neq -1$  else  $\mathbf{R}$
- 

**3.2.2 Radif-based Qafiya Search (RBQS)**

Authors exert a widely employed method for next word prediction through language modeling<sup>(57)</sup>. We calculate the chance for next word ( $w_i$ ) to appear via conditional probability( $\mathbb{P}$ ) with its previous words ( $\mathbb{P}$  is expandable through chain-rule, as shown in equation 4).

$$\begin{aligned}
 \mathbf{P}(w_1, w_2, \dots, w_{i-1}, w_i) &= \mathbf{P}(w_i \mid w_1, w_2, \dots, w_{i-1}) \\
 &= \prod_i \mathbf{P}(w_i \mid w_1 \dots w_{i-1}); \quad i > 1
 \end{aligned}
 \tag{4}$$

Since, for radif, there can be too many combinations of words (depending on its size, i.e., number of tokens therein), therefore, we remodel  $\mathbf{P}$  as  $\mathbf{P}'$  with Markov assumption, and limit the sequence of words to  $n$ -grams (see equation 5). Since qafiya appears before radif; and we are more interested into finding suitable qawafee w.r.t a target qafiya  $\mathbb{Q}$  on provided radif  $\mathbb{R}$ , thus, use of bi-grams in Markov Model (MM) would make sense<sup>(57)</sup>, such that the probability for candidate qafiya is calculated given the

first token in  $\mathbb{R}$ .

$$\begin{aligned} \mathbf{P}'(w_{i-n}, \dots, w_{i-1}, w_i) &\approx \mathbf{P}'(w_i | w_1, w_2, \dots, w_{i-1}) \\ &\approx \prod_i \mathbf{P}'(w_i | w_{i-n} \dots w_{i-1}); i > 1 \end{aligned} \tag{5}$$

Equation 6 renders MM for bi-gram with add-1 smoothed scoring function  $\Phi(\cdot)$  which takes a candidate qafiya  $c$ , first token  $r$  in radif  $\mathbb{R}$ , and dictionaries of uni- and bi-grams  $\mathbf{D1}, \mathbf{D2}$ ; while algorithm 3 shows the overall procedure for RBQS.

$$\Phi(c, r, \mathbf{D1}, \mathbf{D2}) = \mathbf{P}(c | r) = \frac{\|c, r\| + 1}{\|r\| + \text{vocab}} = \frac{\mathbf{D2}[\langle c, r \rangle] + 1}{\mathbf{D1}[r] + \|\mathbf{D1}\|} \tag{6}$$

---

### Algorithm 3. Radif-based Qafiya Searching

---

- 0: **procedure** *RBQS*( $\mathbf{I}, \mathbf{R}, \mathbf{D1}, \mathbf{D2}, \mathbf{Q}, t$ ) ◁  $\mathbf{I}$  be the preprocessed dictionary (see algorithm 1);  $\mathbf{R}$  be the tuple of tokenized radif;  $\mathbf{D1}$  and  $\mathbf{D2}$  respectively, be the dictionary of uni- and bi-grams, where the key is  $x$ -gram and value, is its frequency in the corpus;  $\mathbf{Q}$  be the target qafiya, and  $t$  be the number of top suggestions.
  - 1:  $\mathbf{C} \leftarrow QS(\mathbf{I}, \mathbf{Q}, -1)$  ◁ Using algorithm 2, we retrieve a list of all qawafee for the given  $\mathbf{Q}$ .
  - 2:  $\mathbf{S} \leftarrow \{c \rightarrow \Phi(c, \mathbf{R}[0], \mathbf{D1}, \mathbf{D2}) | c \in \mathbf{C}\}$  ◁ be a dictionary where key is candidate word and value is  $\Phi(\cdot)$ .
  - 3:  $\mathbf{S}' \leftarrow \langle \text{be the list of words, sorted w.r.t values in } \mathbf{S} \rangle$
  - 4: **return**  $\mathbf{S}'[0 : t]$  if  $t \neq -1$  else  $\mathbf{S}'$
- 

## 4 Results and Discussion

### 4.1 Proposed methodology vs Rekhta org

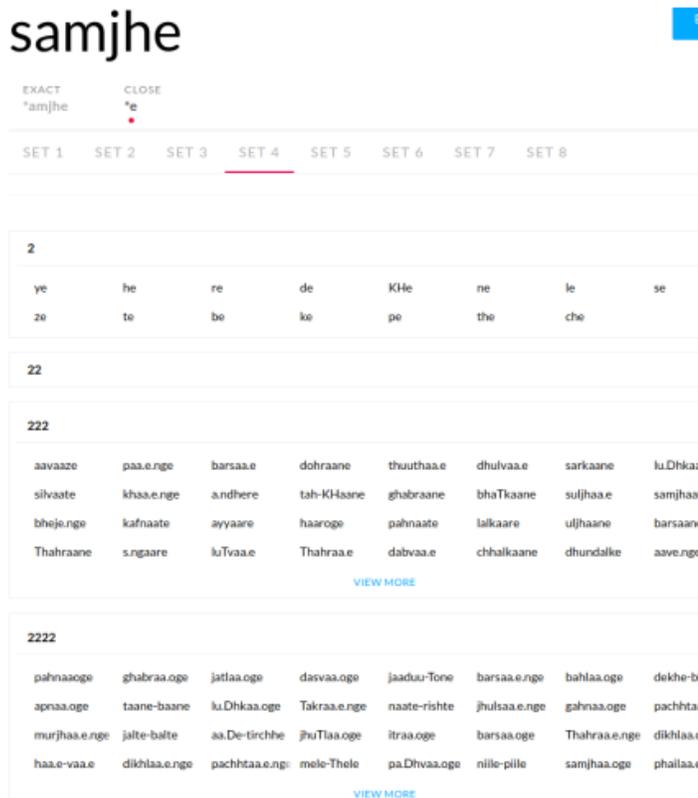
The presented work appeared to be unique for both methodology and results. The proposed method is thoroughly tested with various words and compared with the Qaafiya Dictionary (QD) powered by Rekhta. We found that QD often fails to produce good results when the target word is bigger in length. To a higher degree of surprise, a lot of common words are absent in the QD. Though the proposed method is thoroughly tested and the entire lists of retrieved rhyming words were manually examined but to keep the space top-3 rhyming words are reported for each distance metric.

To show the comparison between the QD and proposed work we have selected target word  /समझे [smd3<sup>h</sup>e:] (verb; ‘understand’). QD responded with no results in their Exact category, while the results in the category Close are not enough invigorating for prosody<sup>3</sup>. Figure 2a and 2b show the output of the QD for Exact and Close categories respectively for the given word. At times it appears that QD begins to produce derivational morphology of a word w.r.t future tense or to use the honorific style communication, for instance, pahnaaoge ([literal] ‘make dressed’) (see figure 2b, the first word under section ‘2222’), thus we cannot count it to be a better rhyming word. In comparison to QD, the method proposed in this work retrieves better words for rhyming as shown in table 3.

<sup>3</sup><https://www.rekhta.org/qaafiya/?qafiya=samjhe> (accessed May 30, 2020)

**Table 3.** Resulting rhymes (through the proposed methodology) on given the target word **سَمجھ** /समझ [səmdʒʰe].

<p><b>top-3 qafiya with Hamming Distance</b></p> <ol style="list-style-type: none"> <li>1. <b>سولجھ</b> [suldʒʰe] (untangle)</li> <li>2. <b>سُجھ</b> [su:dʒʰe] (brainstorm)</li> <li>3. <b>ساڙھ</b> [sa:dʒʰe] (partnership)</li> </ol> <p><b>top-3 qafiya with Levenshtein Distance</b></p> <ol style="list-style-type: none"> <li>1. <b>ساڙھ</b> [sa:dʒʰe]</li> <li>2. <b>رِجھ</b> [ri:dʒʰe] (fond of)</li> <li>3. <b>رِڙھ</b> [rɪ:dʒʰe] (plural of Punjabi-folk character)</li> </ol>	<p><b>top-3 qafiya with Damerau–Levenshtein Distance</b></p> <ol style="list-style-type: none"> <li>1. <b>سولجھ</b> [sʊldʒʰe]</li> <li>2. <b>سُجھ</b> [su:dʒʰe]</li> <li>3. <b>ساڙھ</b> [sa:dʒʰe]</li> </ol> <p><b>top-3 qafiya with Jaro–Winkler Distance</b></p> <ol style="list-style-type: none"> <li>1. <b>رِڙھ</b> [rɪ:dʒʰe]</li> <li>2. <b>بُڙھ</b> [bu:dʒʰe] (solved)</li> <li>3. <b>رِجھ</b> [ri:dʒʰe]</li> </ol>
---	--



(a) The image is set for the exact category.

(b) The image is set for the close category.

**Fig 2.** Screenshot of output rendered by Qaafiya Dictionary powered by Rekhta.org against the word **سَمجھ** /समझ [səmdʒʰe].

### 4.2 Agreement between distance metrics

We are also interested in the quantified answer to show the inter-agreement or inter-similarity between the word lists produced by different metrics. Hence, the two ordered sets of rhyming words (A and B) produced with different edit-distances are tested in a pair-wise fashion. Since the similarity between any two metrics is symmetrical, i.e., have shown only the lower triangle in table 4.

$$M(\mathbf{B}, \mathbf{A}) = M(\mathbf{A}, \mathbf{B}) = \frac{\sum_i 1 \text{ if } A_i = B_i \text{ else } 0}{\min(\|\mathbf{A}\|, \|\mathbf{B}\|)} \tag{7}$$

**Table 4.** Comparison of inter-distance metrics similarities. Abbreviations in columns (left to right) and rows (top to bottom) respectively point to Hamming, Levenshtein, Damerau–Levenshtein, and Jaro–Winkler distances. The rightmost column shows the average.

	Ham.	Lev.	D-L	J-W	$\bar{d}$
Ham.	1.				.0093
Lev.	.0124	1.			.1993
D-L	.0130	.5844	1.		.1995
J-W	.0027	.0011	.0011	1.	.0016

We can see (in table 4) the least similarity is found between J–W and D–L distances i.e., .1% and in contrast, D–L and Levenshtein distances perform roughly similar in comparison to the rest of distance metrics. However, the rest of all numeric figures (except diagonal) in table 4 are significantly low which to mean intuitively reflects word lists possessing a variety of different words.

Consider table 4 (the left portion, before the vertical bar) as a square matrix, namely  $D$ . Thus, from the matrix averaged similarity of a distance metric ( $D[\cdot]$ ) is calculated via equation 8. Forbye it, we have ensured to exclude the value of the metric itself (values at diagonal) in the computation.

$$\bar{d}_{[i]} = \frac{1}{n-1} \cdot \left( \sum_{c=0}^i D[i, c]; \quad \text{if } c \neq i \right) + \left( \sum_{r=i+1}^{n-1} D[r, i]; \quad \text{if } i+1 < n \right) \tag{8}$$

Where  $n$  is the number of metrics,  $i$  is the index of the corresponding metric,  $c$  and  $r$  represents the row and column incidences.

### 4.3 Radif-based Qafiya Search

The results for radif-based qafiya search are also positive. Results for the RBQS are shown in table 5, where (in row 1) we can see the suggested rhymes based on single word radif and similarly rhymes for the multi-word radif (in row 2). We notice that not only the rhymes are appropriate but also the intrinsic sense of formed phrases (radif+qafiya) are semantically correct. This is due to the employment of language modeling technique that (outside the supervised learning techniques) is very cursory in next word predictions; hence, the technique further elicits candidate rhymes according to the probability of its usage w.r.t the previous words available in the corpus.

**Table 5.** Top-3 rhymes based on subsequent radif.

Resulting rhymes with the radif consisting of a single word.	<p><b>Qafiya</b> دیکھیں / देखें [d̪e:kʰiː] (subjunctive form verb (SFV), ‘see’)</p> <p>1. رکھیں / रखें गे [rəkh̪iː] (SFV, ‘put’ or ‘keep’) → رکھیں گے / रखें ग</p> <p>2. بیٹھیں / बैठें गे [beːʃiː] (SFV, ‘sit’) → بیٹھیں گے / बैठें ग</p> <p>3. سمझें / समझें [səmdʒiː] (SFV, ‘understand’) → سمझیں گے / समझें गे</p>	<p><b>Radif</b> گے / गे [geː] (future modal verb, ‘will’)</p>
Resulting rhymes with the radif consisting of two words.	<p><b>Qafiya</b> ہوا / हुआ [hʊvâ] (perfective verb, ‘happen’)</p> <p>1. موآ / मुआ [muâ] (adjective, ‘poor’) → موآ کیا / मुआ क्या हे</p> <p>2. دعا / दआ [duʔaː] (verb, ‘pray’) → دعا کیا / दआ क्या हे</p> <p>3. چھوا / छुआ [tʃʰuɑː] (verb, ‘touch’) → چھوا کیا / छुआ क्या हे</p>	<p><b>Radif</b> کیا / क्या हे [kjaː fiɛ] (WH phrase, [literal] ‘what’)</p>

### 4.4 Running time analysis of distance metrics

The overall time taken in preprocessing datasets and producing dictionaries does not matter, but for the sake of reporting, the yielded figures are—on a machine with a configuration of core i7 and Ubuntu as the operating system—is  $\approx 36$  minutes. Thus, rather making discussion on the preprocessing time, we share insights into the running time elapsed by distance metrics we performed a full-throttle experiment; which involves retrieval of top 1500 rhyming words against 1000 most frequent and distinct words (excluding stopwords) in the poetry corpus. Keeping the symmetric behavior of distance metric in mind, we have implemented two specific checks to ensure inputting words are not the same; and the distance for the same pair of words is not re-calculated.

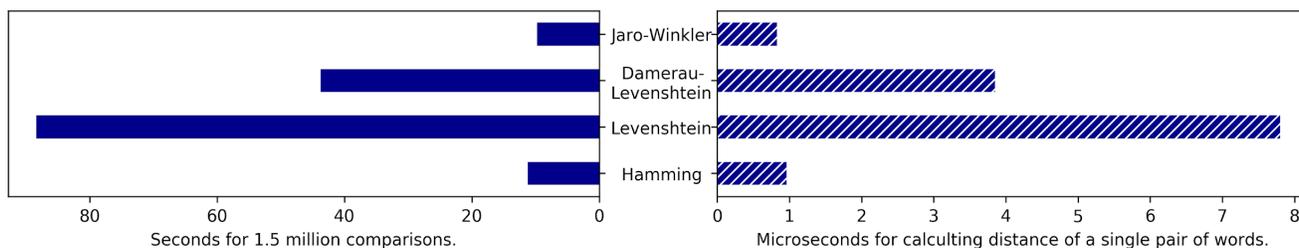


Fig 3. Total and average running time elapsed by distance metrics in retrieving top-1500 rhymes for 1000 distinct words.

Figure 3 shows the overall running time for all distance metrics as per the system limits, experiment settings, and criterion defined above. On the left subplot, the overall running time is shown whereas the right subplot shows the running time of a single word pair. Thus, for  $(1000 \times 1500 =) 1.5$  M comparison, we can see the most robust distance metric is Jaro–Winkler followed by Hamming distance. The laziest metric to calculate differences among 1.5 M distinct pairs of words is vanilla Levenshtein distance. The running time for the single word pair appears to be the same but in proportions of microseconds.

## 5 Conclusion and Future Work

A good assembly of rhyming words is not only core of the poetry but also adds poetic flavor to the prose. Regular expression-based methods can guarantee you the quick lookups but the retrieved results are not ranked. In contrast, this paper effectively shows the utility of the n-gram suffixes and edit-distance metrics for the retrieval of ranked-rhyme suggestions. Among all distance metrics, Jaro–Winkler distance is found to be the most favorable metric for rhyme suggestions in terms of the running time and the variety of rhymes. The work is done for Hindi/Urdu poetry by exploiting the tri-script poetry corpus. One natural question arises that instead of the poetry corpus, why did not a monolingual prose corpus is inducted? Hence, to answer the query, authors maintain that though the monolingual prose corpus will share more variety of words but on the same note poetic quality in n-grams diminish. Thus, we recommend enriching the poetry corpus instead of exploiting prose corpus.

We have noticed that for Hindi, character n-grams should be used where

In the future, the existing work can be appraised by considering phonetic value comparison; specifically for standard Urdu, where under human cognition it is very easy to pronounce words with or without diacritics (harakats); however, in practice words may sound differently even they end on the same final character n-gram. Another potential future work can be the retrieval of suggestions w.r.t the metrical weights of the Hindi/Urdu poetry system.

### Acknowledgment

Authors would like to thank Professor Syed Shahid Hasan Kamal of Government Dehli College, Karachi for his guidance, discussions, and useful insights.

### References

- 1) Maugham WS. Cakes and ale. Random House. 2010.
- 2) Hashmi R. اصناف ادب / Asnaaf-e-Adab. Lahore. Sang-e-Meel Publishers. 1991.
- 3) Crowe RJ, Delmore S, Hall WJ. American poetry at mid-century. 1958.
- 4) Abu Raihan Muhammad Ibn Ahmed Al-Biruni. تحقيق ما للهند من مقولة مقبولة في العقل او مردولة / Tahqiq Mā Li-l-Hind Min Maqūla Maqbūla Fī'l-fiaql Au Marḍūla. Dairat Al-Māarif Al-Umaniya, Hyderabad, reprint Beirut. Hyderabad, reprint Beirut. 1030.

- 5) Rampuri NUG. *بحر الفصاحت*/Beher-ul-Fasahat. Munshi Nawal Kishore. 1885.
- 6) Grishman R. Computational linguistics: An introduction. and others, editor;Cambridge University Press. 1986.
- 7) Bittcher S, Clarke LAC, Cormack GV. Information retrieval: Implementing and evaluating search engines. and others, editor;MIT Press. 2016.
- 8) Zhai C, Massung S. Text data management and analysis: a practical introduction to information retrieval and text mining. Association for Computing Machinery and Morgan & Claypool. 2016.
- 9) Luong MT, Pham H, Manning CD. Effective approaches to attention-based neural machine translation. arXiv arXiv:1508.04025. 2015.
- 10) Khan MY, Ahmed T. Pseudo transfer learning by exploiting monolingual corpus: An experiment on roman urdu transliteration. In: International Conference on Intelligent Technologies and Applications. Springer. 2019.
- 11) Wen TH, Gasic M, Mrksic N, Su PH, Vandyke D, Young S. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. 2015.
- 12) Radford A, Narasimhan K, Salimans T, Sutskever I. Improving language understanding by generative pre-training. 2018.
- 13) Allahyari M, Pouriyeh S, Assefi M, Safaei S, Trippe ED, Gutierrez JB, et al. A brief survey of text mining: Classification, clustering and extraction techniques. arXiv. arXiv:1707.02919. 2017.
- 14) Kowsari K, Meimandi KJ, Heidarysafa M, Mendu S, Brown D, Brown. Text Classification Algorithms: A Survey. *Information*. 2019;10(4). Available from: <https://dx.doi.org/10.3390/info10040150>.
- 15) Kachru Y. Hindi-urdu. *The Major Languages of South Asia, The Middle East and Africa*. 2003;p. 52–68.
- 16) Sahitya Akademi. A History of Urdu literature. and others, editor;SouthAsiaBooks. 1993.
- 17) Prakash B. Writing partition: Aesthetics and ideology in Hindi and Urdu literature. and others, editor;Pearson Education India. 2009.
- 18) Busch A. Poetry of kings: The classical Hindi literature of Mughal India. and others, editor;Oxford University Press. 2011.
- 19) Eberhard DM, Simons GF, Fennig CD. Ethnologue: Languages of Asia. and others, editor;SIL International. 2019.
- 20) Khan MY, Nizami MS. Urdu Sentiment Corpus (v1. 0): Linguistic Exploration and Visualization of Labeled Dataset for Urdu Sentiment Analysis. In: International Conference on Information Science and Communication Technology (ICISCT). IEEE. 2020.
- 21) Daud A, Khan W, Che D. Urdu language processing: a survey. *Artificial Intelligence Review*. 2017;47(3):279–311. Available from: <https://dx.doi.org/10.1007/s10462-016-9482-x>.
- 22) Everaert C. Tracing the boundaries between Hindi and Urdu: Lost and added in translation between 20th century short stories. .
- 23) Kuiper K. The Culture of India. and others, editor;Britannica Educational Publishing. 2010.
- 24) Basu M. The Rhetoric of Hindutva. and others, editor;Cambridge University Press. 2017.
- 25) García AIM. El gramático Pompeyo y el legado sintáctico de Servio. *L'antiquité classique*. 2013;82:69–90. Available from: <https://dx.doi.org/10.3406/antiq.2013.3827>.
- 26) Rao C, Vaid J, Srinivasan N, Chen HC. Orthographic characteristics speed Hindi word naming but slow Urdu naming: evidence from Hindi/Urdu biliterates. *Reading and Writing*. 2011;24(6):679–695. Available from: <https://dx.doi.org/10.1007/s11145-010-9256-9>.
- 27) Nizami MS, Khan MY, Ahmed T. Towards a generic approach for pos-tagwise lexical similarity of languages. In: International Conference on Intelligent Technologies and Applications. Springer. 2019.
- 28) Ahmed T, Nizami MS, Khan MY. Discovering lexical similarity through articulatory feature-based phonetic edit distance. 2020.
- 29) Taqi AS. *رموز شاعری*/Rumooz-e-Shairi. and others, editor;Alqamar Enterprises. 2003.
- 30) Reinöhl U. Grammaticalization and the rise of configurationality in Indo-Aryan;vol. 20. and others, editor;Oxford University Press. 2016.
- 31) Cardona G. Indo-aryan languages. In: and others, editor. The Major Languages of South Asia, the Middle East and Africa. Routledge. 2003;p. 29–34.
- 32) Khansir AA, Mozafari N. The impact of persian language on indian languages. *Theory & Practice in Language Studies*. 2014;4(11).
- 33) Deo A, Kiparsky P. Poetries in contact: Arabic, persian, and urdu. *Frontiers of comparative metrics*. 2011;p. 147–173.
- 34) Lochtefeld GJ. The Illustrated Encyclopedia of Hinduism;vol. 1. and others, editor;The Rosen Publishing Group, Inc. 2001.
- 35) Deo AH. The metrical organization of Classical Sanskrit verse. *Journal of Linguistics*. 2007;43(1):63–114. Available from: <https://dx.doi.org/10.1017/s0022226706004452>.
- 36) Rahman A. Hazaron Khawaishen Aisi: The Wonderful World of Urdu Ghazals. 2018.
- 37) Chand KK. Urdu Ghazals: An Anthology from 16th to 20th Century. and others, editor;Sterling Publishers Pvt. Ltd. 1995.
- 38) Khan M. 'Pasha'. Urdu readings section: Izāfat. 2008.
- 39) Delacy R. Teach Yourself Beginner's Urdu Script. 2003.
- 40) Muhammad A. 'Iqbal'. *بائاب*/Baang-e-Dara;vol. 1. and others, editor. 1905.
- 41) Khalil MAK. Bang-i-dara (the call of the marching bell) translation. and others, editor;Tayyib Printers. 1991.
- 42) Khan M. 'Momin'. *کلیات مومن*/Kulyat-e-Momin. Majlis-e-Trakkiye Adab, Lahore. 1964.
- 43) Mumtaz B, Urooj S, Hussain S, Haq EU. Break index (bi) annotated speech corpus for urdu tts. In: and others, editor. Conference of The Oriental Chapter of International Committee for Coordination and Standardization of Speech Databases and Assessment Techniques (O-COCOSDA). IEEE. 2016;p. 22–27.
- 44) Urooj S, Mumtaz B, Hussain S. Urdu intonation. *Journal of South Asian Linguistics*. 2019;10.
- 45) Butt M, Jabeen F, Bögel T. Verb cluster internal wh-phrases in urdu: Prosody, syntax and semantics/pragmatics. *Linguistic analysis*. 2016;40:445–487.
- 46) Butt M, Jabeen F, Bögel T. Ambiguity resolution via the syntax-prosody interface: the case of kya 'what' in urdu/hindi. Prosody in syntactic encoding. Berlin: de Gruyter. 2018.
- 47) Jabeen F, Braun B. Production and perception of prosodic cues in narrow & corrective focus in urdu/hindi. *9th International Conference on Speech Prosody*. 2018;p. 30–34.
- 48) Hussain SS, National University of Computing and Emerging Sciences –FAST. Prosody in urdu poetry - a phonological approach. 2005.
- 49) Pritchett FW, Khaliq AK. Urdu Meter: A Practical Handbook. Pritchett F, Khaliq KA, et al., editors. 1987.
- 50) Aruuz. Available from: <https://aruuz.com/taqi/.com>.
- 51) Rekhta. Available from: <https://rekhta.org>.
- 52) Richardson L. Beautiful soup documentation. 2007.
- 53) Bookstein A, Kulyukin AV, Raita T. Generalized hamming distance. *Information Retrieval*. 2002;5(4):353–375.
- 54) Levenshtein VI. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet physics doklady*. 1966;10(8).
- 55) Damerau FJ. A technique for computer detection and correction of spelling errors. *Communications of the ACM*. 1964;7(3):171–176. Available from: <https://dx.doi.org/10.1145/363958.363994>.
- 56) Winkler EW. The state of record linkage and current research problems. In: and others, editor. Statistical Research Division, US Census Bureau. 1999.

57) Manning CD, Raghavan P, Schütze H. Introduction to information retrieval. and others, editor;Cambridge university press. 2008.