BRILL

CAHIERS DE LINGUISTIQUE ASIE ORIENTALE
EAST ASIAN LANGUAGES AND LINGUISTICS
51 (2022) 46–80

CAHIERS DE LINGUISTIQUE ASIE ORIENTALE
brill.com/clao

Leveraging graph algorithms to speed up the annotation of large rhymed corpora

Julien BALEY | ORCID: 0000-0003-1056-6211
SOAS University of London, London, UK
julien.baley@gmail.com

Abstract

Rhyming patterns play a crucial role in the phonological reconstruction of earlier stages of Chinese. The past few years have seen the emergence of the use of graphs to model rhyming patterns, notably with List's (2016) proposal to use graph community detection as a way to go beyond the limits of the link-and-bind method and test new hypotheses regarding phonological reconstruction. List's approach requires the existence of a rhyme-annotated corpus; such corpora are rare and prohibitively expensive to produce. The present paper solves this problem by introducing several strategies to automate annotation. Among others, the main contribution is the use of graph community detection itself to build an automatic annotator. This annotator requires no previous annotation, no knowledge of phonology, and automatically adapts to corpora of different periods by learning their rhyme categories. Through a series of case studies, we demonstrate the viability of the approach in quickly annotating hundreds of thousands of poems with high accuracy.

Keywords

data annotation – Chinese rhymes – rhyme network – Middle Chinese phonology

Résumé

Les rimes jouent un rôle crucial dans la reconstruction phonologique des stades antérieurs du chinois. Ces dernières années ont vu l'émergence de l'utilisation des graphes pour modéliser les schémas de rimes, notamment avec la proposition de List (2016) d'utiliser la détection de communauté de graphes afin de dépasser les limites de la

© JULIEN BALEY, 2022 | DOI:10.1163/19606028-bja10019

méthode link-and-bind et de tester de nouvelles hypothèses concernant la reconstruction de la phonologie. L'approche de List requiert l'existence d'un corpus annoté de rimes; de tels corpus sont rares et d'un coût prohibitif à produire. Le présent article résout ce problème en introduisant plusieurs stratégies pour automatiser l'annotation. Entre autres, la principale contribution est l'utilisation de la détection de communauté de graphes elle-même pour construire un annotateur automatique. Cet annotateur ne nécessite aucune annotation préalable, aucune connaissance en phonologie, et s'adapte automatiquement aux corpus de différentes périodes en apprenant leurs catégories de rimes. À travers une série d'études de cas, nous démontrons la viabilité de l'approche en annotant rapidement des centaines de milliers de poèmes avec une grande précision.

Mots-clés

annotation de données – rimes du chinois – réseau de rimes – phonologie du chinois moyen

1 Introduction

Throughout the history of Chinese poetry, rhyme has occupied an important place: already present in inscriptions on bronze vessels,[1] its use is pervasive in the *Shījīng* 詩經 (*Classic of Poetry*),[2] the earliest compendium of Chinese poetry dating back to the 1st millennium BC.[3]

Beyond the study of poetry, rhyme has been of interest to historical linguistics, and in particular to the phonological reconstruction of earlier stages of the Chinese language: since the Chinese script does not indicate precise phonetic information, rhyming texts allow historical linguists to deduce that a set of characters, because they rhyme in a given text, must have some phonetic similarities.

[1] Behr, 'Inscriptional Evidence and the Origins of Poetic Form in Early China'.

[2] This article uses the following romanisation conventions: proper nouns, book and poem titles, as well as technical jargon are romanised using *Hànyǔ pīnyīn*. For single Chinese characters cited from poems and rhyme books, for which the goal is to give an idea of their pronunciation at the time of composition, the romanization follows the Middle Chinese (MC) reconstruction by Baxter and Sagart (2014). Where relevant, Early Middle Chinese (EMC), Late Middle Chinese (LMC) and Early Mandarin (EM) reconstructions by Pulleyblank (1991) are also cited; in those cases, the use of Pulleyblank reconstructions is always made explicit; otherwise, Baxter-Sagart's MC is to be assumed.

[3] Dobson, 'Linguistic Evidence and the Dating of the "Book of Songs"'.

In order to carry out the analysis of rhyming patterns, researchers need access to a corpus of poetry that has been annotated to highlight these patterns. List, Hill and Foster have proposed an annotation standard[4] that is intended to be simple, language-independent, human- and machine-readable and would allow researchers to contribute the result of their annotations towards a shared collection of annotated poetry; List further developed tools to help researchers speed up their annotation effort.[5]

Two issues have become apparent with this need for annotated poetry: first, rhyme annotation is a costly activity which requires expertise in the particular period and dialect of the text being annotated, and is a particularly slow task to carry out, making poor use of a researcher's time. Second, the analysis of these annotations becomes impractical as the size of the accumulated data increases. To address the latter issue, List proposed to use a graph theory technique from the family of community detection algorithms[6] and demonstrated that it can be used to accelerate the generation and testing of Old Chinese reconstruction hypotheses. In this paper, I address the remaining issue of annotation cost by proposing to develop automated annotators: such annotators are able to process thousands of poems in a matter of seconds, freeing up researchers to spend more time on the analysis of their data. In the rest of this article, I present the concepts behind the creation of automated annotators, how flexible the concept is, how it enables the development of various annotation strategies and how we can assess the quality of the those strategies. Finally, I propose to reuse List's proposed community detection algorithms to produce an effective annotator and demonstrate its power on a range of case studies of Tang and Song poetry.

1.1 *The need for rhyme annotation*

To study rhyming practices, researchers need to have access to corpora in which poems are annotated for rhyme. Typically, this means that for a given poem, the characters which rhyme should be marked as such so that further analysis can be carried out. Such corpora are rare and researchers publishing research on rhyme tend to only publish the results of their analysis, without publishing the data that afforded such analysis. List highlighted this lack of data publication[7]

[4] List, Hill, and Foster, 'Towards a Standardized Annotation of Rhyme Judgments in Chinese Historical Phonology'.

[5] List, 'PoePy. A Python Library for the Quantitative Handling of Poetry'.

[6] List, 'Using Network Models to Analyze Old Chinese Rhyme Data'.

[7] List, Hill, and Foster, 'Towards a Standardized Annotation of Rhyme Judgments in Chinese Historical Phonology'.

and further proposed a language-independent scheme for the annotation of poems and developed a set of tools that allow researchers to easily annotate poems using his proposal.[8]

Corollary to the question of publication of annotation is the question of annotation cost. To carry statistical weight, studies of rhyme need to rely on the annotation of relatively large amounts of poems; in the case of testing specific hypotheses like the existence of the **-r* coda in List's paper, one could envision annotating only poems that exhibit the **-j*, **-n* and hypothesized **-r*. The *Shījīng*, however, is a heavily studied corpus and its relatively small size allowed it to be annotated by hand. The extant Chinese rhymed corpus, on the other hand, is thousands of times larger and most of it has never been annotated. By aiming to drastically reduce the cost of annotation and instead focus effort on verification—which is much faster, especially if areas of annotation 'doubt' are highlighted—we think it is possible to quickly reach a stage where the vast majority of the extant Chinese rhymed corpus would be annotated for rhyme and publicly available. Automating the annotation process could allow us to reproduce this type of analysis at a lower cost, and on larger corpora. This would allow us to further detect patterns that do not fit pre-imagined scenarios, and perhaps discover language change phenomena that have hitherto escaped the attention of researchers.

2 Annotator formalisation

Before proceeding to creating automated annotators, we propose to formally define what constitutes an annotator, the shape of its inputs and outputs (and therefore an annotation scheme); we then present a trivial example of annotator that fits this formal definition.

2.1 *Annotation scheme*

For annotation purposes, we use the format proposed by List et al.,[9] which consists of adding marks like [a], [b], [c] etc. before the characters that rhyme in a poem: if two characters rhyme, then they are attributed the same mark; if they do not, they are attributed a different mark. Characters that are not involved in any rhyme are not marked. For instance, Table 1 presents an annotation of Wū

[8] List, 'PoePy. A Python Library for the Quantitative Handling of Poetry'.

[9] List, Hill, and Foster, 'Towards a Standardized Annotation of Rhyme Judgments in Chinese Historical Phonology'.

TABLE 1 Wū yè tí 烏夜啼 (Crow Calls at Night) by Lǐ Yù 李煜 (937–978) (QTWDC 4.450)

Original poem	Rhyme	MC	Rhyme group	Annotated poem
無言獨上西樓，	樓	*luw*	a	無言獨上西[a]樓，
月如鉤。	鉤	*kuw*	a	月如[a]鉤。
寂寞梧桐深院鎖清秋。	秋	*tshjuw*	a	寂寞梧桐深院鎖清[a]秋。
剪不斷，	斷	*twan*$_H$	b	剪不[b]斷，
理還亂，	亂	*lwan*$_H$	b	理還[b]亂，
是離愁。	愁	*dzrjuw*	a	是離[a]愁。
別是一般滋味在心頭。	頭	*duw*	a	別是一般滋味在心[a]頭。

yè tí 烏夜啼 (Crow Calls at Night) by Lǐ Yù 李煜 (937–978) (QTWDC 4.450). In this example, the rhyme is pretty clear: in the Baxter-Sagart MC reconstruction, all characters either rhyme in *-uw* ([a]) or *-an*$_H$ ([b]). In cases where the rhymes are less perfect, we estimate the poet's intention, and annotate accordingly. For instance, here, one could ask whether *-uw* and *-juw* rhyme, and decide to annotate them separately as [a] and [c], resulting in the pattern aacbbca, as opposed to the proposed aaabbaa.

2.2 *Formal definition of an automatic annotator*

For the purpose of devising various automatic annotators, we formally define what constitutes an annotator. Any system that possesses the following characteristics qualifies as an annotator:

- Input: an arbitrary string *input_text*.
- Output: a string *output_text* that is an augmentation of *input_text*, with zero or more annotations. Aside from the annotations, *outfloloput_text* is identical to *input_text* (i.e. the annotation process can be reversed by stripping annotations). There is no requirement for *output_text* to contain any annotations: if the annotator finds nothing to rhyme, it is valid for *output_text* to be identical to *input_text*.
- Annotations consist of a pair of square brackets containing one or more characters (e.g. [a], [20], [-ang] etc.) that are placed in front of the rhyming unit (a word or a character). The string enclosed between the square brackets is called a *label*. Although the labels *can* be chosen to carry a meaning (e.g. [-ang]), this is not a requirement: labels should be treated as purely symbolic (e.g. [a]). This means that there is no meaningful distinction between a poem being annotated as [a][a][b][b] or [d][d][a][a].

– Two units preceded by identical annotations (i.e. carrying identical labels) are considered to rhyme; in other words, the annotator expresses its judgement by the choice of the labels it applies to certain words or characters.[10]

2.3 *Naïve automated annotation*

As a trivial example, we propose to use the simplest annotation strategy possible: in a given poem, consider that all characters that are in rhyming position belong to a single rhyme.[11] In terms of the formalism defined above, this means that all characters in rhyming position will be annotated with the same label (e.g. [a]). Taking the example of a regulated quatrain, we usually have 2 characters in rhyme position[12] (the last characters of even-numbered lines[13]), both annotated [a], which we can summarise as exhibiting the pattern ‘aa’. With a 8-line poem, similarly, we obtain the pattern ‘aaaa’ and applying the principle to any poem, we would always obtain a pattern of the shape ‘aaa … aaa’.

It is worth noting that, while naïve, this strategy has some merit because many poems actually fit this pattern: regulated verse—which occupies a large part of the extant corpus—is supposed to exhibit a single rhyme throughout the poem; in the specific case of quatrains, it is practically always the case that the pattern is ‘aa’, as there would otherwise be no rhyme. This means that using this annotation strategy would yield a non-zero accuracy, and could serve as a statistical baseline for other annotators. We will also see later that such an annotator can serve as the first step of a more complex annotating strategy.

[10] Note that any label appearing exactly once in *output_text* could just as well not appear at all since the value of annotations reside in them coming in pairs or sets. It makes no difference whether such an annotation is removed or left as is.

[11] The opposite approach, namely considering that all characters belong to different rhymes, is of no practical value.

[12] Note that this supposes that our system is able to assess which characters are in rhyming position, i.e. to find the characters that could participate in a rhyme pair. This requires being able to segment the poem into lines and knowing which lines may or may not participate in a rhyme pattern. This is a hard problem in the general case (see List, ‘Using Network Models to Analyze Old Chinese Rhyme Data’, 222 n. 5), as different styles of poetry have different rules regarding versification, and we might also be interested in non-poetic rhyming texts. For these reasons, the question of segmentation and the highlighting of characters as being in rhyming position is out of the scope of the present study: in the rest of this article, we assume this information to be provided to us. In practice, this assumption is often met, as will be the case in our Tang and Song corpora.

[13] The first line of a quatrain can also participate in the rhyme; for simplicity and to keep this article short (see previous note), we ignore this case and restrict our focus to the even-numbered lines, leaving exploration of rhyming schemes for later treatment.

3 Annotators from explicit phonological data

Although the naïve annotator presented above has the potential to be correct in a large number of cases and therefore save annotation time by pre-filling an annotation, its behaviour is rather simplistic. Going a step up in complexity, we can rely on previous rhyme judgement to annotate new material: for instance, if in a certain poem we annotated *loj* 來 and *khoj* 開 as rhyming (i.e. gave them the same label), the knowledge of this rhyming pair can be used to save time and annotate all poems in which these two characters occur in rhyming position.[14]

Generalising this idea, any dataset that can be interpreted as a partition of characters into rhyming sets can be used to build an annotator, and we can call such annotators "set annotators". Datasets that can be interpreted in such a way already abound: rhyme books are probably the most obvious examples as they are literally structured around the idea of rhyming sets of characters, but one could also imagine using *xiéshēng* 諧聲 series or repurposing phonological reconstructions; in the first case, the series themselves are sets and an annotator built purely out of those sets would label as rhyming only characters that belong to the same series.[15] In the case of repurposing phonological reconstructions, using the rhyme part of the reconstruction of each character in a given system as the label (e.g. using Pulleyblank's reconstruction of Late Middle Chinese, [aj] for both *laj* 來 and *kʰaj* 開) effectively produces a set annotator that produces an output consistent with the rhyme judgements such a reconstruction system would produce. Such annotation strategies can provide a significant speed-up of the annotation work.

3.1 *Rhyme book-based annotation*

To illustrate the design decisions that the creation of a set annotator can entail, we create a *Guǎngyùn*-based 廣韻 annotator. The use the *Guǎngyùn* is motivated by its broad compatibility with the *Qièyùn* 切韻,[16] its size—over 26,000 characters, over twice as large as the *Qièyùn*—and, of paramount importance, its availability: we have access to a copy of the *Sòngběn Guǎngyùn* 宋本廣韻

[14] One could use this principle to build an annotator based on a published annotated corpus, which provides another reason to publish one's annotations in a standard format.

[15] As is, such an annotator is likely to perform poorly, so this example is provided mainly for illustration. We could however imagine annotators that rely on several bodies of knowledge to produce a rhyme judgement, and *xiéshēng* could be one of them.

[16] E.G. Pulleyblank, *Middle Chinese: A Study in Historical Phonology*, 135.

digitised into an XML[17] format that is easy to use. At the time of writing, we are not aware of a similarly convenient digitisation of other rhyme books.[18]

Relying on the digitised copy of the *Guǎngyùn*, we perform the preparatory steps once:

- For a rhyme category heading (*tuwng* 東, *towng* 冬, *tsyowng* 鐘, *kæwng* 江 etc.), we load all the characters in that category (e.g. under *tuwng* 東, we find *duwng* 同, *duwng* 童 etc.; under *towng* 冬 we find *nowng* 農 etc. and under *tsyowng* 鐘 we find *tsyhowng* 衝, *ljowng* 龍 etc.) and build a mapping going from the characters to the category: 同⇒東, 童⇒東, 農⇒冬, 衝⇒鐘, 龍⇒ 鐘 etc.
- In the table of contents of the rhyme book, we parse all the *dúyòng* 獨用 ("used singly") and *tōngyòng* 通用 ("used interchangeably") annotations, e.g. "二冬與鍾通" ("second category, *towng* 冬, is interchangeable with *tsyowng* 鍾"). With that information, we amend the mapping above to replace *tsyowng* 鍾 by *towng* 冬 etc., resulting in the final annotation mapping: 同⇒東, 童⇒東, 農⇒冬, 衝⇒冬, 龍⇒冬 etc. Note that such a step is not strictly necessary to build an annotator: we merge the *tōngyòng* rhymes because the "interchangeability" annotation essentially allows poets to treat characters from two distinct rhyme categories as rhyming; it is therefore justified to annotate them as such. In practice, we have found the annotator to perform poorly when this merging step was turned off.

For a given poem, the annotator then performs the following steps:

- We annotate each character in rhyming position with the rhyme category found in the map above. If the character appears in several categories (e.g. $twan_{X/H}$ 斷 can be $hwan_{X}$ 緩 or $hwan_{H}$ 换), we make a note of the ambiguity.[19] The annotation scheme proposed by List et al. does not cover ambiguity or indecision, so we propose here to use a forward slash between labels: [斷/緩/ 换]
- We go a second time through the poem to resolve the possible ambiguities: if, of the two or more options, one of them occur elsewhere in the poem, but not the others, we keep that option. In the above example, if we find

[17] '*Sòngběn Guǎngyùn* 宋本廣韻 XML', accessed 9 August 2020.

[18] We note here that such digitisation and compilation would be of great value for future computational projects. Websites exist that contain the *Qièyùn*, *Zhōngyuán yīnyùn* and later rhyme books, but to our knowledge, none that presents it in a conveniently downloadable format.

[19] Note that this means that set annotators do not necessarily need to strictly partition the character set into non-overlapping sets: a character can belong to several sets, e.g. when it has several pronunciations.

TABLE 2 Guǎngyùn-based annotation of Wū yè tí 烏夜啼 (Crow Calls at Night) by Lǐ Yù 李煜 (937–978) (QTWDC 4.450)

Original poem	Rhyme	Interchangeable category	Resolved ambiguity	Annotated poem
無言獨上西樓，	樓	尤	尤	無言獨上西[a]樓，
月如鉤。	鉤	尤	尤	月如[a]鉤。
寂寞梧桐深院鎖清秋。	秋	尤	尤	寂寞梧桐深院鎖清[a]秋。
剪不斷，	斷	換/緩	換	剪不[b]斷，
理還亂，	亂	換	換	理還[b]亂，
是離愁。	愁	尤	尤	是離[a]愁。
別是一般滋味在心頭。	頭	尤	尤	別是一般滋味在心[a]頭。

another *hwan*$_H$ 換 rhyme in the poem but no *hwan*$_X$ 緩, then we consider the ambiguity resolved and choose *hwan*$_H$ 換. (cf. worked example below). If ambiguity remains, we leave it unresolved.[20]

- Finally, based on those categories, we annotate the poem with symbolic labels [a], [b], [c] etc., starting from [a] and moving to the next letter when we encounter a rhyme category not previously encountered in the poem. In the case of ambiguities, we use a symbolic label for each possibility, e.g. [a/b/c].

We apply the process to the example poem of the Annotation scheme section in Table 2. Reading from left to right, first we identify the rhyme character, map it to its "interchangeable" *Guǎngyùn* category; we then resolve the *hwan*$_H$ 換/*hwan*$_X$ 緩 ambiguity on the grounds of the presence of an unambiguous *hwan*$_H$ 換 below in the poem (meaning that *hwan*$_H$ 換 / *hwan*$_X$ 緩 should be read as *hwan*$_H$ 換), and finally we replace the categories by letters, producing 'aaabbaa'. In this case, this happens to be the same result as was produced by the manual annotation.[21]

[20] This 2-pass approach can be borrowed by any set annotator in which rhyme sets overlap (i.e. one character can belong to multiple sets).

[21] Note that this also addresses the doubt formulated during the manual annotation explanation, with regards to the *-uw* and *-juw* categories: since the Guǎngyùn lists *huw* 侯 and *'jiw* 幽 as interchangeable, the two categories rhyme.

4 Annotators from implicit phonological data

Although rhyme books are useful to quickly build annotators, they are likely to be effective annotators only for the period in which they were written, and since the earliest extant one—the *Qièyùn* 切韻—dates from 601CE, we may want to develop methods that are more generally applicable across time.

Earlier, we mentioned reusing existing rhyme judgement to accelerate the process of annotation. This could lend itself to a semi-manual annotation process, whereby a manual annotation tool (such as the one developed by List et al. as an addition to their rhyme annotation proposal paper) could be extended to support an annotation memory: the system would memorise previously annotated poems and new poems would then be pre-annotated to the extent that the memory permits.

This process, which borrows its idea from the bind and link method (*xìlián fǎ* 系聯法), relies on the assumption that rhyme is a transitive property: if A rhymes with B and B rhymes with C, then A rhymes with C. Although this assumption seems reasonable in most cases, there can be situations where it does not hold, especially if the corpus contains multiple authors from different periods and places and therefore having different phonological systems. To take an extreme example, although Modern Standard Mandarin *rù* 入 (MC: nyip) rhymes with *lù* 路(MC: lu_H) and Middle Chinese *nyip* 入 (pīnyīn: rù) rhymes with with *kip* 急 (pīnyīn: jí), there is no transitivity as 路 rhymes with 急 neither in MSM (lù / jí) nor in MC (lu_H / kip). Aside from this trivial example, according to List's calculations,[22] assuming rhyme transitivity in the *Shījīng* alone would have us conclude that, of the 1,845 characters that appear in rhyme position, 1,539 (83 %) of them rhyme with each other; this can be explained by the presence of unusual rhymes which create bridges between otherwise non-rhyming sets of characters. List then argues that the problem with this method is the lack of weighting of the rhyming evidence, attributing as much weight to a rare rhymes as to common ones. This is a problem that any memory-based annotator is likely to suffer from; unless remedial measures are taken—e.g. annotating certain rhymes as dubious and not to be memorised—such an annotator is probably undesirable.

[22] Johann-Mattis List, 'Using Network Models to Analyze Old Chinese Rhyme Data', 230.

TABLE 3 Dēng gǔ Yè chéng 登古鄴城 (Climbing the old Ramparts of Ye) by Cén Shēn 岑參 (718?–769?) (QTS 199.2061)

Poem	Rhyming character	Middle Chinese
下馬登鄴城，		
城空復何見。	見	*hen*$_H$
東風吹野火，		
暮入飛雲殿。	殿	*den*$_H$
城隅南對望陵臺，	臺	*doj*
漳水東流不復回。	回	*hwoj*
武帝宮中人去盡，		
年年春色爲誰來。	來	*loj*

4.1 *Rhyme community detection*

To address the problem of weighting, List proposes to represent poem rhymes as graphs and apply an algorithm to remove the noise generated by rare rhymes.

4.1.1 Representing rhymes as graphs

First, we start with the graph representation: rhyming characters are represented by circles called “nodes” (or “vertices”) and the rhyming relationship between two characters is represented by a line called an “edge”. For illustration purposes, let us look at the following three short poems: The first poem is *Dēng gǔ Yè chéng* 登古鄴城 (Climbing the old Ramparts of Ye) by Cén Shēn 岑參 (718?–769?) (QTS 199.2061). It is presented in Table 3, along with an analysis of its rhymes and a Middle Chinese reconstruction of the rhyming characters, for illustration.

In this poem, we get two groups of rhymes: in the first quatrain, the -*en*$_H$ rhyme, and in the second quatrain the -*oj* rhyme. Using this poem, we can create two sets of characters that rhyme: {見, 殿} and {臺, 回, 來}. The process is repeated in Table 4 a poem that also contains *loj* 來 and *hwoj* 回, *Fā Liú láng pǔ* 發劉郎浦 (Departing from Lord Liu’s Shore) by Dù Fǔ 杜甫 (712–770) (QTS 223.2373).

Here again, we get two sets of rhymes: -*u*$_X$ {浦, 午, 虎} and -*oj* {回, 催, 來}. A third and final poem is presented in Table 5, *Chángmén yuàn* 長門怨 (Grief at the Long Gate) by Xú Huì 徐惠 (627–650) (QTS 5.59), which produces a new set of rhymes, {殿, 扇, 賤, 薦}. This gives us the five sets {見, 殿}, {臺, 回, 來}, {浦, 午, 虎}, {回, 催, 來} and {殿, 扇, 賤, 薦}. We used these sets to produce the graph in Figure 1.

TABLE 4 Fā Liú láng pǔ 發劉郎浦 (Departing from Lord Liu's Shore) by Dù Fǔ 杜甫 (712–770) (QTS 223.2373)

Poem	Rhyming character	Middle Chinese
挂帆早發劉郎浦，	浦	phu_X
疾風颯颯昏亭午。	午	ngu_X
舟中無日不沙塵，		
岸上空村盡豺虎。	虎	xu_X
十日北風風未回，	回	*hwoj*
客行歲晚晚相催。	催	*tshwoj*
白頭厭伴漁人宿，		
黃帽青鞋歸去來。	來	*loj*

TABLE 5 Chángmén yuàn 長門怨 (Grief at the Long Gate) by Xú Huì 徐惠 (627–650) (QTS 5.59)

Poem	Rhyming character	Middle Chinese
舊愛柏梁臺，		
新寵昭陽殿。	殿	den_H
守分辭芳輦，		
含情泣團扇。	扇	syen_H
一朝歌舞榮，		
夙昔詩書賤。	賤	dzjen_H
頹恩誠已矣，		
覆水難重薦。	薦	tsen_H

With this representation, the original information is better preserved and yet keeps the spirit of the linking method: each "component" (a group of nodes that are linked together, directly or indirectly) corresponds to a linking method's merged set, and the actual graph edges tell us whether the two characters actually rhymed in a poem or not; for instance, this graph tells us that *doj* 臺 and *tshwoj* 催 never rhymed together in a poem, although they belong to the same set because they were both used to rhyme with *loj* 來 and *hwoj* 回. Not represented in this graph is the idea of node and edge weights: as we build the graph, we keep track of how many times each character and each connection between characters has been seen. Although we do not display these weights

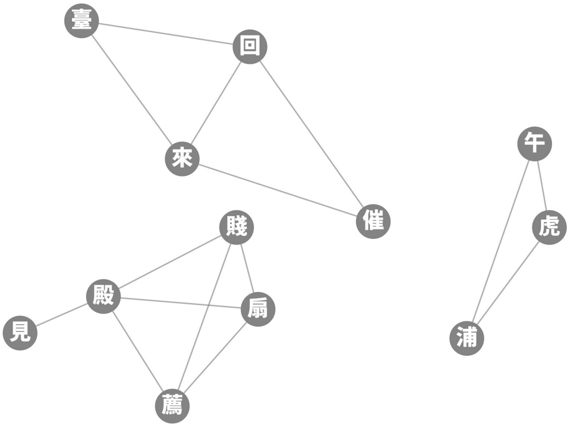

FIGURE 1
The rhymes of Cén Shēn, Dù Fǔ and Xú Huì's poems, displayed as a rhyme graph

in the images of this article in order to keep the graphs somewhat readable, weights will play a role in rhyme community detection.

4.1.2 Getting some clarity through rhyme communities

As mentioned earlier, this process—when applied to the *Shījīng*—produces a massive component containing 83 % of all the rhyming characters. To address this problem, List proposes to use a family of algorithms called "graph community detection".[23] The technique comes from the field of graph theory and was originally developed for the analysis of social and biological networks.[24] The idea behind community detection is precisely to decide the questions of transitivity and evidence weight across the graph, edge by edge, and stems from studies of social networks: given a graph of friendships between members of the network, "A is friends with B" and "B is friends with C", how likely is A to be friends with C? Such a situation is shown in Figure 2.

If we then find that D is friends with both B and C, we can perhaps think of the friendship between B and C as stronger, and this may increase the probability that C could be friends with A. Finally, if we find that D is also friends with A—as shown in Figure 3—then it means that C and A have two friends in common, and that these two friends are friends with each other: A and C are now more likely to be friends, and if they are not friends already, then at least their mutual friendships would make it likely for them to become friends.

[23] The specific algorithm he uses (and which we use too) is called 'InfoMap', see Rosvall and Bergstrom, 'Maps of Random Walks on Complex Networks Reveal Community Structure', 1118–1123.

[24] Girvan and Newman, 'Community Structure in Social and Biological Networks'.

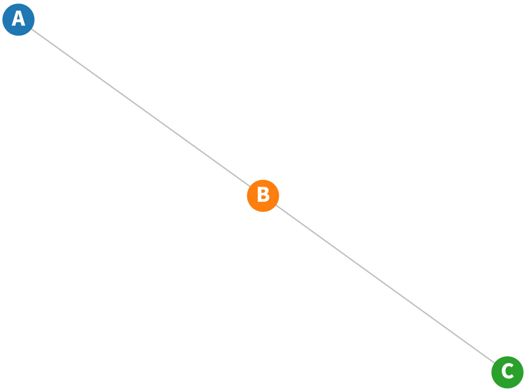

FIGURE 2
A is friends with B, and B with C, but A is not friends with C

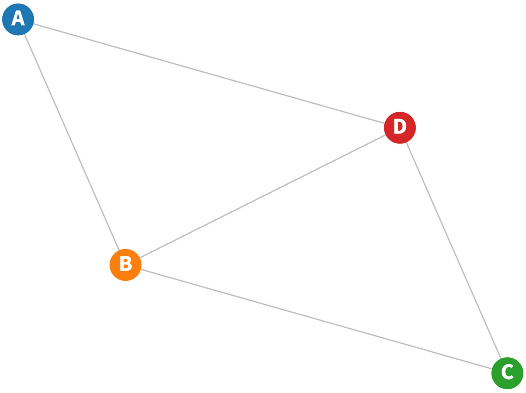

FIGURE 3
A, B, D are friends, B, C, D are friends, but A is not friends with C

Now we imagine a more complex graph, as in Figure 4, in which A, B and C are friends with each other, and additionally D is friends with A and B but not C. On the other hand, E, F and G are friends with each other, and finally E is friends with C. In such a graph, we would be unlikely to assume that everyone is friends with everyone (e.g. F with D); instead, we may find out that {A, B, C, D} are all part of a chess club, that {E, F, G} play tennis, and finally that C and E are siblings. In graph theory, we would call the ensembles {A, B, C, D} and {E, F, G} "communities". Formally, a community is defined as having more links (i.e. edges) between nodes of the community than with nodes outside the community. There can be cases where a single node belongs to several communities, e.g. if C played chess and tennis, as in Figure 5 where C is marked in orange meaning "belongs to both communities". The role of community detection algorithms then becomes to take in a full graph such as the ones of Figure 4 and Figure 5 and highlight the communities they contain.

Graph community is a concept that is useful beyond evaluating social network friendships and List has proposed to use the concept to explore rhymes of

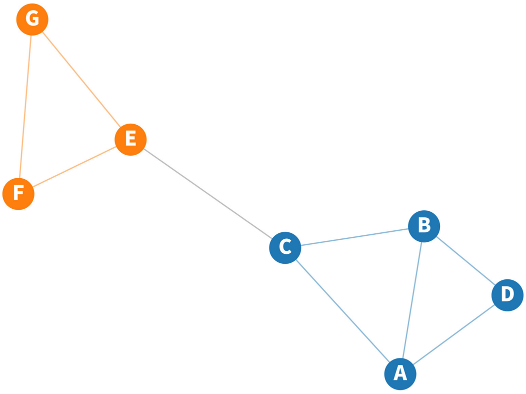

FIGURE 4
C, from the chess club (A, B, C, D), is friends with E, from the tennis club (E, F, G)

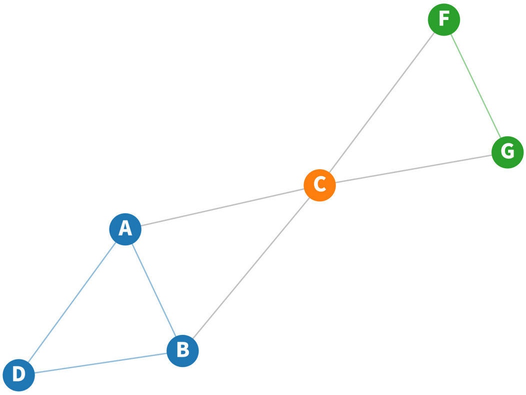

FIGURE 5
C belongs to both chess (A, B, C, D) and tennis (C, F, G) clubs

the *Shījīng* 詩經 to investigate why certain characters seem to rhyme arbitrarily with characters reconstructed in Old Chinese as *-*n* or *-*j*, and he uses communities to test Baxter's hypothesis of the existence of a separate rhyme group, *-*r*, which appears to rhyme with both *-*n* and *-*j* because of phonological changes between Old and Middle Chinese in which some *-r changed to *-j and others with *-n.[25]

4.2 *Community annotator*

By attributing the nodes of a graph to a list of communities, the community detection algorithms essentially produce a partition of the rhyming characters into sets, which suggests we can use them as the basis of a set annotator. In the rest of the article, we look at the practical details of such a construction, and evaluate the behaviour of such an annotator. A crucial advantage of the

[25] List, 'Using Network Models to Analyze Old Chinese Rhyme Data'.

approach presented in the following section over the technique introduced in List (2016) is that the annotator we build does not require any prior manual annotation nor any phonological knowledge: everything is automatic and learnt from the specific corpus to annotate.

4.2.1 Corpus

To demonstrate that computational techniques can allow us to produce deeper analyses of larger corpora than manual work would allow, we decide to make use of two collections of rhymed material:

- *Quán Táng Shī* 全唐詩 (Complete Poetry of the Tang, QTS),[26] containing 44,388 poems
- *Quán Sòng Shī* 全宋詩 (Complete Poetry of the Song, QSS),[27] containing 209,104 poems

Manually annotating such collections alone would likely require several years, not to mention the analysis of such an annotation. In addition to their size, these corpora conveniently represent a continuous[28] stretch of time spanning from the early 7th century to the late 13th century. As the ambition of their compilers was to be exhaustive, they cover poems from authors from all regions of China.

4.2.2 Rhyme community-based annotation

In contrast with the naïve approach and the *Guǎngyùn*-based approach which both have predetermined behaviours, using the community detection algorithm described earlier has the advantage that it considers the data: how do real poets write poems, and which characters do they choose to rhyme with each other. The output of such an approach could highlight practices that are not predicted by the rule-based approaches described above.

This annotator works in two steps: first, during the learning step, it takes a graph of the rhymes in a corpus, detects communities within this graph and arbitrarily attributes them a letter (community A, community B etc.): the assumption is that any two characters that are in the same community must rhyme. Then, during the annotation step, it then goes through the poems, identifies the rhymes and replace them by the community letters in the same way the *Guǎngyùn*-based annotator replaced the rhymes by their corresponding

[26] Quan Tang Shi, 25 vols (Beijing: Zhonghua Shuju, 1979).

[27] Quan Song Shi, 72 vols (Beijing: Peking University Press, 1998).

[28] Despite their name, these collections also contain poems from the interregnum period commonly referred to as the "Five Dynasties" (*wu dai*五代), so that the sum of QTS and QSS does cover a continuous time span.

rhyme category. Finally, just as in the *Guǎngyùn*-based process above, it does a second pass through the poem and produces a final annotation of the shape 'aabbcc' etc.

The community-based approach cannot work on its own: it needs a pre-existing assumption to work from: for communities to be detected, we first need to build a graph of relationships between characters. For two characters to be detected as being part of the same community, a prerequisite is that there must exist a path between these two characters; in other words, the community detection algorithm always return a graph with fewer edges than in the original graph: it can never create new ones. If we return to the social network analogy, if we have not witnessed a friendship between A and B and that none of the friends (or 'friends of friends … of friends') of A is friends with any friend (or 'friend of friend …') of B, then we would have no reason to suspect a possible connection between A and B.

In a way, the community detection annotator can only invalidate prior hypotheses, not create new ones. This suggests that in order to get a full picture, we need to make as few assumptions as possible regarding what rhymes and what does not. This means that if we used the *Guǎngyùn*-based annotator for initialisation, the community detection algorithm may actually show us sub-divisions within certain rhyme categories (i.e. rhyme splits), but the downside would be that we could *only* discover rhyme splits but never any rhyme merges: once the initial graph decides that *tuwng* 東 and *towng* 冬 do not rhyme, the community detection can only agree with that decision.

Ideally, we would use a manually annotated corpus; since we do not have one, the closest viable strategy is to use the naïve approach described earlier: we give a chance to every connection by first assuming that all characters of a poem in rhyming position rhyme, and then we use the community detection algorithm to invalidate some of these connections. The results are not guaranteed to be entirely correct and the following sections will be dedicated to evaluating the approach.

Using the naïve approach, we build a graph by repeating the following two steps for each poem:

- We list all the rhyming characters in the poem; for each character, if it is not already present in the graph, we add it to the graph with a weight of 1; if it is already present, we simply increase its weight by 1. Node and edge weights are important, as the community detection algorithm will discount rare connections as chance, and keep the common ones as being motivated by rhyme.
- For each possible pair of rhyming characters in the poem, we add an edge between those two characters, or increase its weight if it already exists. Fol-

lowing List's suggestion,[29] the weight of the edge is $\frac{1}{n_{rhymes}-1}$, with $\mathrm{n}_{\mathbf{rhymes}}$ being the number of rhyming characters in the poem. This is to limit the impact of very large poems on the algorithm: a poem with 2 rhymes contributes 1 edge to the graph, one with 3 rhymes contributes 3, and in general one with $\mathrm{n}_{\mathbf{rhymes}}$ contributes $\frac{n_{rhymes}(n_{rhymes}-1)}{2}$ edges. For a very large poem of 300 rhymes, this would represent 44,850 edges; without normalisation of the weight (i.e. assigning a weight of 1), due to the very large amount of edges that link every pair of characters of the poem, the community detection algorithm would conclude that all these form a community.

Once the graph is built, the community detection algorithm is run on it so as to produce communities of rhymes. These communities are then used to annotate poems following the same principle as the rhyme book annotator.

4.2.3 Visual interpretation of the community annotator training

To provide a more concrete understanding of the community annotator's learning step, let us visualise the process. For this purpose, we follow the process described in the "Rhyme community-based annotation" section to train the community annotator on the poems of the *Quán Táng Shī*: this produces a graph containing 6,895 nodes (i.e. distinct rhyming characters) and 401,533 edges.

The first step is to create a graph of the entire corpus and then to train the communities on it. The outcome of the detection is that each node is assigned to a community, characters belonging to the same community being considered as rhyming. As described, the approach taken is to run our community detection on top of a naively annotated corpus. Figure 6 shows a subset of the graph that results from annotating the entire corpus naively. To keep the figure legible, only the most frequent 40 characters and their corresponding edges are shown. This is purely a presentation decision, the algorithm itself was run on the entire corpus. The nodes are coloured according to their detected community, and a manual annotation of the rhymes is provided as legend.

The distribution of nodes in this graph is based on the edges that link nodes: the nodes try to stay as far as possible from each other and are kept close to each other by the edges. Since the naïve annotator considers that everything in a poem rhymes, most characters are linked to all others, but it is the frequency of those links (how often two characters are found to rhyme in the corpus) that decides whether they belong to the same rhyme community. In the figure, we

[29] List, 'Using Network Models to Analyze Old Chinese Rhyme Data', 228.

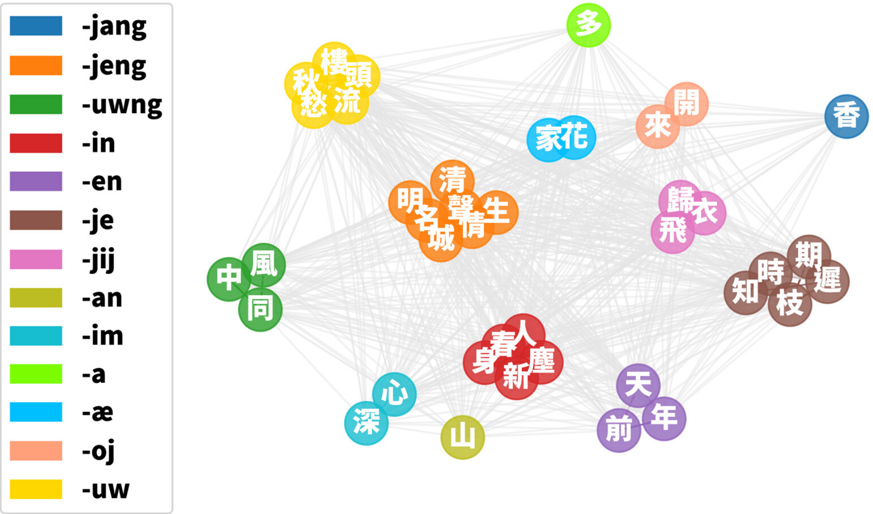

FIGURE 6 Graph of the QTS rhymes, as produced by a naive annotator (only the top 40 characters are shown); nodes are coloured based on their detected community; the legend is a manual annotation of the rhymes in the Baxter-Sagart MC system, it is not produced by the algorithm.

can see that the detected communities are broadly correct, e.g. with the orange *-jeng* cluster containing *mjæng* 明, *tshjeng* 清, *sjeng*聲, *sr(j)æng* 生, *tshjeng* 情, *dzyeng* 城 and *mjieng* 名, or the yellow *-uw* cluster containing *tshjuw* 秋, *dzrjuw* 愁, *ljuw* 流, *duw* 頭 and *luw* 樓.

This coloured graph representation shows how our community annotator analyses new poems: each of the communities / components in the graph constitutes a set of rhyming characters (in the annotator's "mind") and the annotator then uses the usual set annotator's routine to produce an annotation of the poems. In our example, this suggests that although *mjæng* 明 and *tshjeng* 清 belonged to different rhymes in the *Guǎngyùn* (*kæng* 庚 and *tshjeng* 清, respectively), they were used as rhyming often enough that the community annotator would consider them as rhyming.

4.3 *Case studies*

To illustrate how the process works in practice, we present here a few examples. Considering the three automated annotators, Naive, *Guǎngyùn* and Community, there are five possible scenarios:

1. All three produce the same output (Naive=*Guǎngyùn*=Community, N=G=C)
2. Community produces a different output from the other two (N=G, C≠N, C≠G)

3. *Guǎngyùn* produces a different output from the other two (N=C, G≠N, G≠C)
4. Naive produces a different output from the other two (G=C, N≠C, N≠G)
5. All three produce different outputs (N≠C, N≠G, G≠C)

Since the Naive annotator always produces a straight ‘aaa … aaa’ output, the case in which it disagrees with the other two is not particularly interesting: it simply suggests that the poem has a more complex rhyme structure, as we have seen in the poem presented previously (‘aaabbaa’). We will therefore ignore this outcome and focus on the remaining four cases. In general, the higher the inter-annotator agreement, the more likely the annotators are correct. For this reason, when all three produce the same output, there is generally no reason to doubt that they are correct: the poem has one rhyme throughout, as is often expected, it obeys the rules of the *Guǎngyùn*, and the empirical Community annotator confirms the pattern. We only examine one such poem, for illustration purposes, to show how the annotators work in a well-behaved case.

As for the three remaining cases, in which the *Guǎngyùn* and the Community annotators disagree, they all present a particular opportunity:

- N=C, G≠N, G≠C: following the assumption above that the higher inter-annotator agreement, the higher the chance they are correct, the case in which the *Guǎngyùn* annotator produces a different annotator from the other two is a situation in which the *Guǎngyùn* is more likely to be wrong, demonstrating the power of the Community annotator.
- N≠C, N≠G, G≠C: this case is a step up in complexity compared to the previous one: as previously, the Community and *Guǎngyùn* annotators disagree, but this time none of them produces a Naive annotation. Since the Community annotator is trained on top of the Naive annotator, a case in which it ends up disagreeing with the Naive annotator is a proof to its ability to learn complex patterns. Here, we choose a poem for which Community differs from *Guǎngyùn* and produces a regular pattern of the shape ‘aabbccddee’, the regularity of which makes it more likely to be correct.
- N=G, C≠N, C≠G: in this case, since the *Guǎngyùn* annotator agrees with the Naive one, they are likely to be correct and the Community annotator is consequently likely to be wrong; we present such a case as a demonstration of the limits of the Community annotator.

4.3.1 All annotators produce the same output (N=G=C)

Cases in which all three annotators produce the same output necessarily follow a ‘aaa … aaa’ pattern, the only pattern the Naive annotator can produce. This represents 60 % of the poems in our corpus. One such example is the 20-line poem *Hé Guǒcuì Zhào Gǔn Liángbì Píngjiāng tíng* 和果倅趙袞良弼平江

TABLE 6 Hé Guǒcuì Zhào Gǔn Liángbì Píngjiāng tíng 和果倅趙袞良弼平江亭 (In Response to the Aide for the Fruits Zhao 'Good Assistant' Gun's "The Pingjiang pavilion") by Féng Shān 馮山 (?–1094) (QSS 740.8640)

Poem	Rhyme	MC	*GY* category	*GY* annotation
牛峰絕頂凌雲閣，形勝潼江照開廓。	廓	khwak	鐸	a
涪水臺中隱凡菴，坐對西山橫碧落。	落	lak	鐸	a
頃嘗登覽放懷抱，鈎掛軒窗捲帷幕。	幕	mak	鐸	a
氣酣景熟狀不起，一片 風騷情索寞。	寞	mak	鐸	a
充城累身倦未出，無事門庭可羅雀。	雀	tsjak	鐸	a
翠筠五瑞掃塵土，澄照清風易椽薄。	薄	bak	鐸	a
平江最好最後到，隱几凌雲僅依約。	約	ʔjak	鐸	a
屏星主人蹋閒散，長句森森邀我作。	作	tsak	鐸	a
漳濱病起吟魂耗，中散慵来筆鋒弱。	弱	nyak	鐸	a
鏗金入木將奈何，手把新篇空愧怍。	怍	dzak	鐸	a

亭 (In Response to the Aide for the Fruits Zhao 'Good Assistant' Gun's "The Pingjiang pavilion") by Féng Shān 馮山 (?–1094) (*QSS* 740.8640), presented in Table 6 along with the characters that appear in rhyme positions.

Since the decision of the Guǎngyùn-based annotator is easily understood, let us focus on the Community annotator. The annotator starts from the Naive assumption ("in all poems, everything rhymes together"), compiles statistics of co-occurrences of rhymes, and decides whether or not characters actually rhyme:

- if two characters often appear together (or appear together with an intermediate, e.g. A is often seen with B and B often seen with C), they probably rhyme.
- if they rarely appear together, they probably do not rhyme.

Let us examine statistics of co-occurrences for the rhyme characters of our poem (廓落幕寞雀薄約作弱怍). First, we collect all poems in which any of these characters appear as a rhyme: we obtain a list of 2,280 poems. Then, we collect all the rhyme characters of those poems: we obtain a total of 28,861 characters, representing 2178 distinct characters (i.e. each character occurred around 13 times, on average). Of these 28,861 characters, 68% belong to the *dak* 鐸 rhyme, and the remaining 32% are spread across the 182 other rhyme categories. As a result, purely based on the data (the Community annotator has no knowledge of the *Guǎngyùn*), the annotator is able to find that the rhyme characters of our poem belong to a very well-defined community gravitating around the characters *lak* 落, *bak* 薄, *tsak* 作, *lak* 樂, *khwak* 廓, *ʔak* 惡, *xak* 壑, *kak* 閣, *ʔjak* 約, *mak* 寞, the 10 most frequent characters of that community, of which 6 actually appear in our poem. Since the annotator thinks these characters belong to a rhyming community, it annotates them as rhyming, resulting in the 'aaaaaaaaaa' annotation.

4.3.2 The *Guǎngyùn* annotator produces a different output from the other two (N=G, C≠N, C≠G)

The next case study is the situation in which the Naive and Community annotators produce the same output, but the *Guǎngyùn* annotator differs. In the extreme, some poems produce an annotation that has as many different letters as it is long (e.g. 'abcdefgh', 8 letters for 8 annotations), implying that the *Guǎngyùn* annotator thinks nothing rhymes in the poem. One such case is *Sòng tíxíng Sūn Qí shǎoqīng yí Húběi zhuǎnyùn* 送提刑孫頎少卿移湖北轉運 ("Sending off Vice-President Sun Qi, Commissary for Judicial Affairs, on the occasion of his transfer to Hubei") by Sū Zhé 蘇轍 (1039–1112) (*QSS* 856.9921), a poem containing 20 lines (10 rhyme characters) presented in Table 7.

This gives a pattern 'abcacdefgh' or 'abcadefgdh', depending on whether we choose to resolve the *hjwon*$_X$ 願 / *hjwon*$_H$ 阮 ambiguity to pair with line 3 or line 9.[30] When we look at the reconstruction, we see that the tone is not consistent between the lines, oscillating between departing tone (H) and rising tone (X). Although incorrect in theory, by the 9th century, it had become common to mix these two tones and make them rhyme.[31] To build an annotator that takes this into account, we can use a rhyme table like the *Yùnjìng* 韻鏡 (the Rhyme Mirror) and merge rising and departing tones as one category by replacing all the rising tones by their departing tone equivalent (e.g. *tuwng*$_X$ 董 by *suwng*$_H$ 送). The process is shown in Table 8.

After the merge, we obtain a significantly clearer picture, with either 'aababcdcbd' or 'aabacbdbcd' (depending on which way we want to resolve the *hjwon*$_X$ 願 / sen$_H$ 霰 ambiguity). The fact that the community annotator tells us the poem is 'aaaaaaaaaa' suggests that the data must have convinced it thusly. The ten rhyme characters of the poem fall within a single community, which means that there must have been enough co-occurrences of these characters over the corpus that looking at the graph would convince us that they are undistinguishable. Since the full graph for the corpus contains 11,505

[30] And if the ambiguity cannot be resolved, this gives 'abcadefghi'.

[31] Pulleyblank, 'The Rhyming Categories of Li Ho (791–817)', 1.

TABLE 7 Sòng tíxíng Sūn Qí shǎoqīng yí Húběi zhuǎnyùn 送提刑孫頎少卿移湖北轉運 ("Sending off Vice-President Sun Qi, Commissary for Judicial Affairs, on the occasion of his transfer to Hubei") by Sū Zhé 蘇轍 (1039–1112) (QSS 856.9921)

Poem	Rhyme	MC	*GY* category
持節憂邦刑，職業已自簡。	簡	kɛn$_{X}$	產[a]
下車攝留都，談笑事亦辦。	辦	bɛn$_{H}$	襉
開軒揖佳客，退食事書卷。	卷	kjwen$_{X/H}$[b]	線 / 獮
爲政曾幾何，清風自無限。	限	hɛn$_{X}$	產
官居歲月迫，歸念湖湘遠。	遠	hjwon$_{X/H}$	願 / 阮
依依東軒竹，凜凜故人面。	面	mjien$_{H}$	線
詔書遂公私，使節許新換。	換	hwan$_{H}$	換
舊治行當經，家山企可見。	見	ken$_{H}$	霰
宦遊得鄉國，勞苦顧猶願。	願	ngjwon$_{H}$	願
歸旆正滂洋，行輈豈容緩。	緩	hwan$_{X}$	緩

a As *kean*$_{X}$ 簡 does not appear in the *Guǎngyùn*, we take the category from its graphic variant *kean*$_{X}$ 𥳑. The *Lǐbù yùn lüè* 禮部韻略 (Ministry of Rites's abridged rhymes), which does list *kean*$_{X}$ 簡, confirms the rhyme category.

b For 卷 and 遠, both H (departing tone) and X (rising tone) are possible; as the surrounding context has a mix of both tones, disambiguating is not feasible automatically.

TABLE 8 Merging of rising and departing rhyme categories in Sòng tíxíng Sūn Qí shǎoqīng yí Húběi zhuǎnyùn 送提刑孫頎少卿移湖北轉運 (QSS 856.9921)

Character	*Guǎngyùn* category	Rising ⇒ departing merge
簡	產	襉
辦	襉	襉
卷	線 / 獮	願 / 霰
限	產	襉
遠	願 / 阮	願
面	線	霰
換	翰	翰
見	霰	霰
願	願	願
緩	旱	翰

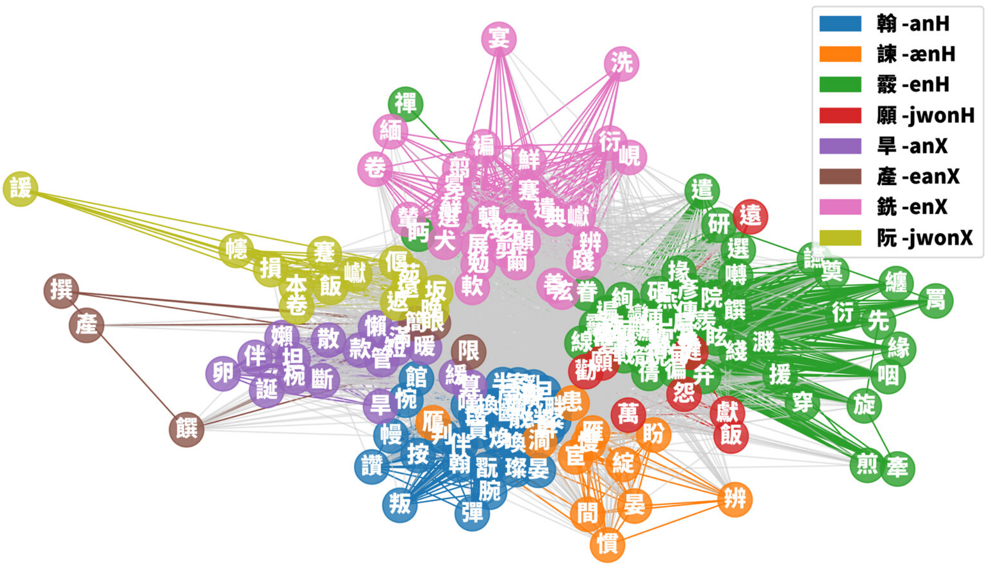

FIGURE 7 Graph of the Guǎngyùn rhymes for QTS+QSS, limited to rhyme categories for which a character appears in Sū Zhé's "Sending off Vice-President Sun Qi, Commissary for Judicial Affairs, on the occasion of his transfer to Hubei"; each colour is distinct Guǎngyùn rhyme

nodes, in Figure 7 we show a sub-graph that contains the nodes that belong to the community that the annotator used to label this poem, which contains 589 nodes.

The individual nodes are of little importance here, but they are colour-coded based on the corresponding Guǎngyùn rhyme category. On a general level, we can see that the nodes tend to cluster based on their rhyme categories—as we would expect—but these clusters are very close to each other and overlap in places, which suggests that these characters form a community: without the colours added as an aid it would appear as a single cluster.[32]

This suggests that the rhyme pattern 'aaaaaaaaaa', although surprising from the Middle Chinese reconstruction point of view, is correct. The reason for this is that the MC reconstruction is based on the early 7th century *Qièyùn* rhyme book, while the poem is from the late 11th century. Looking at Pulleyblank's Late Middle Chinese (estimated to represent an 8th century pronunciation of the Cháng'ān 長安 dialect) in Table 9, it is clear that those characters had a

[32] It is interesting to note that the graph seems to displays a near-split along the North-East / South-West diagonal: on the left, we find the han$_X$ 旱, sræn$_X$ 潸, sen$_X$ 銑 and ngwjon$_X$ 阮, all in rising tone; on the right, we find their departing tone equivalents han$_H$ 翰, kæn$_H$ 諫, sen$_H$ 霰 and ngjwon$_H$ 願. This suggests that although inter-rhyming rising and departing tone was common, enough poets kept the distinction for it to be visible in the graph.

TABLE 9 Late Middle Chinese reconstrution of the rhymes in Sòng tíxíng Sūn Qí shǎoqīng yí Húběi zhuǎnyùn 送提刑孫頎少卿移湖北轉運 (QSS 856.9921)

Rhyme	Late Middle Chinese	Rhyme	Late Middle Chinese
簡	kjaːnˊ	面	mjianˋ
辧	pɦaːnˋ	換	xɦuanˋ
卷	kyanˊ / kyanˋ	見	xɦjaːnˋ
限	xɦjaːnˋ	願	ŋyanˋ
遠	yanˊ / yanˋ	緩	xɦuanˋ

pronunciation much closer than the *Qièyùn* would suggest. Vowel length and tone aside, Pulleyblank reconstructs all of these characters with the same vowel -*a*-,[33] making it possible for them to rhyme.

To highlight the impact of language change on poems being perceived as rhyming or not, we can train one community annotator on the full Tang corpus (*QTS*) and one on the full Song corpus (*QSS*).[34] As pronunciation and rhyming practices change through time, an annotator trained on a corpus in which the change has not happened yet will produce a different annotation from one trained on a corpus in which the change *has* happened.

The results are presented in Table 10 and show a clear picture: during the Tang, actual rhyming practice—for these specific rhymes—was relatively close to the rhyme book prescription while, by the time of the Song, all of these rhymes could be considered to belong to one larger group, or at least to a set of groups that were close enough to each other to be perceived as a group. To get a visual impression, Figure 8 displays the community graph when the *QTS* annotators is used instead of the full corpus annotator.[35]

Compared to the previous graph, this one is much clearer: although there are some connections between the clusters, we can see five distinct communi-

[33] Of these characters, Pulleyblank reconstructs six as already carrying an -a- in EMC (*kwian* 卷, *wuan* 遠, *mjian* 面, *ɣwan* 換, *ŋuan* 願 and *ɣwan* 緩). He reconstructs the four others as *kɛːn* 簡, *bɛːn* 辧, *ɣɛːn* 限 (from earlier *kəɨn, baɨn and ɣəɨn*, the schwa first lowering to /a/ and the /ɨ/ fronting to /i/, now giving *kain*, *bain*, *ɣain*, before the /i/ caused fronting and lengthening of the vowel to /ɛː/) and *kɛːn* 見. The /ɛː/ then defronted to /aː/, leading to *kjaːn* 簡, *pɦaːn* 辧, *xɦjaːn* 限 and *xɦjaːn* 見.

[34] Using a range of annotators trained on specific periods can help us detect and date specific phonological changes; an article on the topic is in preparation.

[35] As the *QSS* is four times larger than the *QTS* and therefore dominates the corpus, the graph produced by the *QSS* looks similar to the full graph and will not be displayed here.

TABLE 10 Comparison of Guǎngyùn, QTS and QSS annotators for Sòng tíxíng Sūn Qí shǎoqīng yí Húběi zhuǎnyùn 送提刑孫頎少卿移湖北轉運 (QSS 856.9921)

Character	*Guǎngyùn* annotator	*QTS* annotator	*QSS* annotator
簡	a	a	a
辦	b	b	a
卷	c	a	a
限	a	a	a
遠	d	a	a
面	e	c	a
換	f	d	a
見	g	c	a
願	d	c	a
緩	h	e	a

ties,[36] which explains the five-lettered annotation produced by the *QTS* annotator on the poem. This tells us that there was a change in rhyming practice between the two periods. The community detection algorithm itself is only able to highlight a phenomenon, not to explain why. Although we have discussed the reasons for this specific poem, in the general case evolution in rhyme practice can be explained as a combination of the following factors:

- Language change: as in our example, a change in pronunciation causing a merge or split of phonological categories will similarly impact poets' choice of rhyme characters.
- A change of attitude towards the rhyming rules: in our example, inter-rhyming of the rising and departing tones meant that more characters could rhyme together, producing fewer rhyme communities.
- Corpus bias: Southern poets represent 44 % of the *QTS*, while the proportion increased greatly in *QSS*: 77 % for the Northern Song period (960–1127), and 96 % for the Southern Song (1127–1279).[37] What we perceive as diachronic language change could therefore be explained as a difference in dialect.

[36] Note that the *QTS* annotator produces a 5-lettered annotation already indicates relatively free inter-rhyming of several *Guǎngyùn* categories, as we would otherwise expect an 8-lettered annotation.

[37] For a discussion of the causes of such a shift, see Wang Zhaopeng 王兆鵬, 'Spatial Distribution and Displacement of Poetry during the Tang and the Song'.

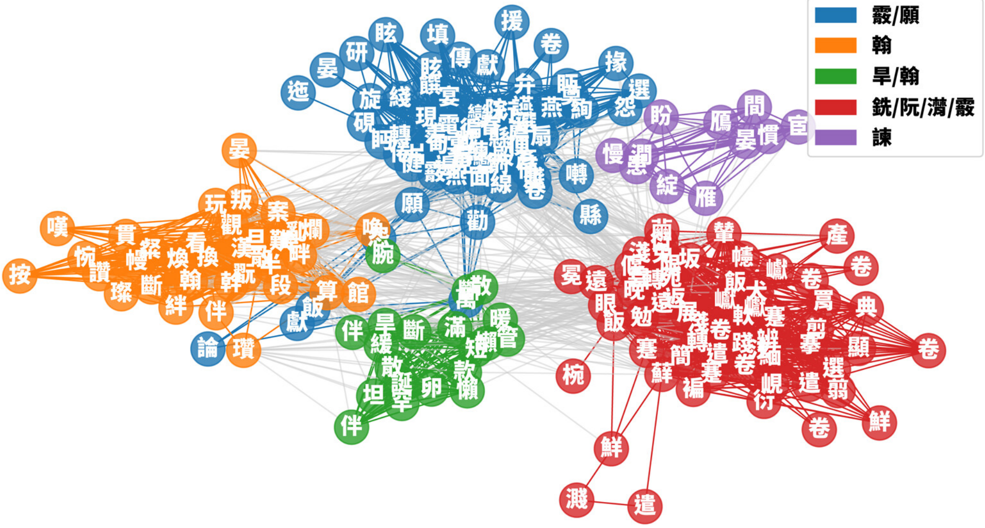

FIGURE 8 Graph of the rhyme communities for QTS only, limited to communities for which at least one character appears in Sū Zhé's "Sending off Vice-President Sun Qi, Commissary for Judicial Affairs, on the occasion of his transfer to Hubei"

4.3.3 All annotators produce a different output (N≠C, N≠G, G≠C)

Since the Community annotator starts from the naïve assumption, cases for which it produces a non-naïve annotation are a demonstration of its power, especially if the *Guǎngyùn* annotator failed to produce the same pattern. Out of all the non-naïve patterns that the annotator could produce, the ones for which the annotation looks regular (e.g. a rhyme change every quatrain) are more likely to be correct.[38]

[38] There are however cases in which an irregular pattern can be of interest, for instance if it is nearly regular (only one annotation is not an 'a') and the *Guǎngyùn* produces the same annotation. One such poem is *Shòu chéngxiāng qí èr* 壽丞相其二 (On the Prime Minister's birthday, part II) by Wú Shìqīng 吳勢卿 (fl. 1241–1260) (QSS 3331.39729), in which the characters in rhyming position are *kuwng* 功, *khuwng* 空, *dzyin* 臣, *trjuwng* 中. The third character, *dzyin* 臣, cannot possible rhyme with the rest, producing 'aaba'. On closer inspection, the third couplet, which reads『一隙暇時關國祚，十分重任賴忠臣。』("A moment of respite and the country is at risk, for this momentous duty, [His Majesty] can count on his loyal minister"): the penultimate character, *trjuwng* 忠, would fit the rhyme perfectly, and swapping the last two characters does not change the meaning ("[His Majesty] can count on his minister's loyalty"). As a confirmation of our reading and that the poem itself contains a mistake, we find the same poem in the QSS—correctly rhyming this time—attributed to another poet, Liú Zǐhuán 劉子寰 (*jìnshì* 進士 in 1217) (QSS 3086.36806). Their overlapping lifespan does not allow us to decide who actually wrote it.

TABLE 11 Yàn tái sì shǒu: qiū 燕臺四首 秋 (Four poems on the Swallow Terrace: Autumn) by Lǐ Shāngyǐn 李商隱 (c. 813–858) (QTS 541.6233)

Poem	Rhyme	MC	*GY* category	Community
月浪衝天天宇濕，涼蟾落盡疎星入。	入	nyip	緝	a
雲屏不動掩孤嚬，西樓一夜風箏急。	急	kip	緝	a
欲織相思花寄遠，終日相思却相怨。	怨	ʔwjon$_H$	元	b
但聞北斗聲迴環，不見長河水清淺。	淺	tsen$_X$	先	b
金魚鎖斷紅桂春，古時塵滿鴛鴦茵。	茵	ʔjin	真	c
堪悲小苑作長道，玉樹未憐亡國人。	人	nyin	真	c
瑤琴愔愔藏楚弄，越羅冷薄金泥重。	重	drjowng$_H$	用	d
簾鉤鸚鵡夜驚霜，喚起南雲繞雲夢。	夢	mjuwng$_H$	送	d
雙璫丁丁聯尺素，內記湘川相識處。	處	tsyho$_H$	御	e
歌唇一世銜雨看，可惜馨香手中故。	故	ku$_H$	暮	e

For instance, Table 11 presents such a poem, *Yàn tái sì shǒu: qiū* 燕臺四首 秋 (Four poems on the Swallow Terrace: Autumn) by Lǐ Shāngyǐn 李商隱 (c. 813–858) (*QTS* 541.6233), in which the *Guǎngyùn* annotator produces 'aabcddefgh'. We need to consider a few things: first, *suwng*$_H$ 送 and *ngjo*$_H$ 御 are marked in the *Guǎngyùn* as *dúyòng* 獨用 ('used singly'), an annotation which usually means that there was a large chance of confusion with the rhymes listed just after it and therefore that it was necessary to specify "use singly", i.e. "do not use interchangeably [with other rhymes]".[39] In the present case, *suwng*$_H$ 送 and *ngjo*$_H$ 御 are immediately followed by *yowng*$_H$ 用 and *mu*$_H$ 暮, precisely the rhymes found in the poem: we can therefore expect that Lǐ Shāngyǐn *did* intend to make these characters rhyme, and the pattern should therefore be 'aabcddeeff'. Second, the remaining problematic pair (*sen* 先 / *ngjwon* 元) was already used by other poets of the time, as Pulleyblank reports about Lǐ Hè,[40] Li Shangyin's quasi-contemporary, and as we have seen in Sū Zhé's poem, albeit

[39] Pulleyblank, Middle Chinese, 139. The history of the development of *dúyòng* and *tōngyòng* annotations is obscure but would stem from a memorial to the throne arguing that the rhymes of the *Qièyùn* were too narrowly defined.

[40] Pulleyblank, 'The Rhyming Categories of Li Ho (791–817)', 13. Pulleyblank's suggestion for explaining the phenomenon is that the vowel in rhyme *ngjwon* 元 was a schwa and the medial was initially the close central unrounded vowel /ɨ/. As the distinction between /ɨ/ and /j/ got lost and /ɨ/ became /j/, this led to the fronting of the schwa, making it close to the *-jen* of the *sjen* 仙 rhyme, itself rhyming freely with *sen* 先.

in a different tone, *sen_H* 霰 and *$ngjwon_H$* 願 being the respective departing tone equivalent of *sen* 先 and *ngjwon* 元.

These two aspects taken into account, we reach the pattern 'aabbccddee', i.e. the poem contains five quatrains, each with its own rhyme, a pattern followed by Lǐ Shāngyǐn in the other three season poems of his tetralogy. The fact that the Community annotator was able to discover this pattern is an achievement: after all, the Community annotator starts from the Naive assumption and works backwards; its default behaviour is to predict 'aaaaaaaaaa'. Yet, despite being taught by the Naive annotator that this poem is 'aaaaaaaaaa', the Community annotator was able to figure out that it was not the case, and that another annotation was preferable.

4.3.4 The community annotator produces a different output from the other two (N=C, G≠N, G≠C)

In the final case study, we look at situations in which the Community annotator is likely to be wrong: it is particularly likely for poems in which the two other annotators agree, i.e. when the *Guǎngyùn* annotator produces a naïve annotation. In most of those cases the Community annotator produces an irregular pattern, e.g. 'aaba' which is usually a sign that the annotator is wrong. These cases are rather rare in our corpus, but Table 12 provides one example, *Méihuā èrshí shǒu qí yībā* 梅花二十首其一八 (Twenty Poems on Plum Blossoms, part 18) by Zhāng Dàoqià 張道洽 (1205–1268) (*QSS* 3293.39249). In the poem, the characters all belong to the same Guǎngyùn category and therefore the annotator produces 'aaaa'. In addition to this, the characters rhyme in several conservative modern dialects such as Hokkien and Cantonese, and it seems clear that the intent of the poet is indeed 'aaaa'. So why does the Community annotator produce 'aaba'?

One way to understand what happens is to look at the two communities produced by the annotator (i.e. 'a' and 'b'), see what characters they contain and what are the corresponding *Guǎngyùn* rhymes. In our case, community 'a' corresponds to the group of rhymes dominated by the *kɛ* 佳 rhyme, while the 'b' community is dominated by the *mæ* 痲 rhyme. This tells us that characters from the *Guǎngyùn kɛ* 佳 rhyme seem to belong to two distinct communities.[41] To get a better idea of what is happening, Figure 9 shows the 'a' community (佳) is shown in light grey, and the 'b' community (痲) in darker grey.

[41] Although characters can appear in several rhyme categories in the *Guǎngyùn*, *kɛ* 佳 itself belongs only to one.

TABLE 12 Méihuā èrshí shǒu qí yībā 梅花二十首其一八 (Twenty Poems on Plum Blossoms, part 18) by Zhāng Dàoqià 張道洽 (1205–1268) (QSS 3293.39249)

Poem	Rhyme	MC	LMC	*GY* category	Community ann.
幾年冷樹雪封骨，一夜東風春透懷。	懷	hwɛj	xɦwa:j	佳	a
花裏清含仙韻度，人中癯似我形骸。	骸	hɛj	xɦja:j	佳	a
三點兩點淡尤好，十枝五枝疏更佳。	佳	kɛ	kja:j	佳	b
野意終多官意少，玉堂茅舍任安排。	排	bɛj	pɦa:j	佳	a

Compared to previous figures, although the two communities are clearly defined, there is a high density of inter-community edges, meaning that there is inter-rhyming between the two categories, but not enough for the algorithm to consider them as a single community. At the boundary, we see a few characters that are practically 'between' the two communities. A zoom on this boundary is displayed in Figure 10.

In the very centre, we see the following characters: 蛙佳涯鮭鞋洼. Aside from the fact[42] that they all use the phonetic component *kwej* 圭, the Ministry of Rites's rhyme book (*Lǐbù yùn*禮部韻) lists each of them except *kɛ* 佳 and *kwej* 鮭 under both rhymes *mæ* 麻 and *kɛ* 佳. This should suggest *kɛ* 佳 belonging to the light grey community, as *kwej* 鮭 does, but instead we see it in the dark grey one. Looking at the characters with which *kɛ* 佳 most often rhymes in Table 13, the list is dominated by *mæ* 麻 rhymes. This suggests that although the character *kɛ* 佳 is listed in rhyme books as belonging only to the *kɛ* 佳 rhyme, in practice poets used it as if it belonged to the *mæ* 麻 rhyme.[43] Such a phenomenon is the sign of a rhyme split: two characters that used to rhyme do not rhyme anymore, due to a diverging phonological evolution.

[42] Or, more likely, related to this fact, but a deeper study of the question will take place elsewhere.

[43] Pulleyblank notes this peculiarity of kja:j 佳 which, along with a handful of characters such as xɦwa:j 話, xɦwa:j 畫, xɦwa:j 卦 and xɦwa:j 卦, lost their final glide, giving a -a final in most modern dialects (compare with kja:j 街 which developed normally into Mandarin *jiē*). He suggests that this is the result of an EMC dialect that was not included in the *Qièyùn* but which later contributed these glide-less forms to the standard language. cf. Pulleyblank, Middle Chinese, 111 and 197.

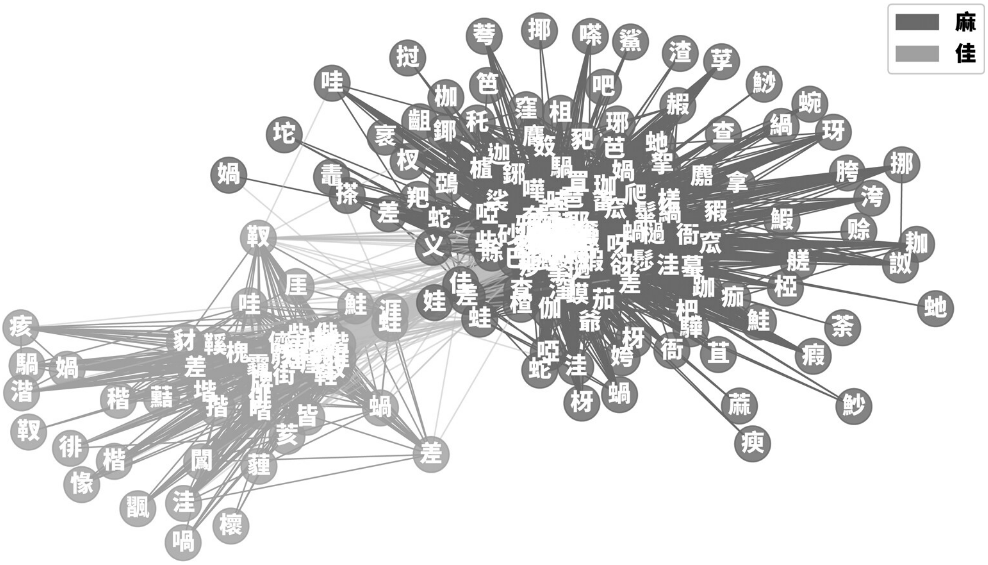

FIGURE 9 The mæ 麻 and kɛ 佳 rhyme communities

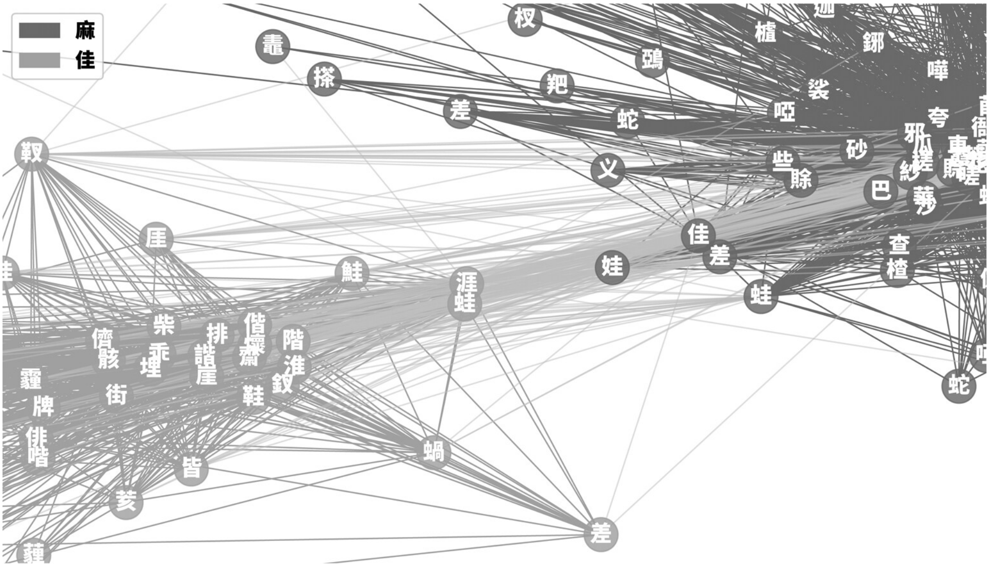

FIGURE 10 The mæ 麻 and kɛ 佳 rhyme communities, up close

TABLE 13 Rhyme categories and LMC reconstruction of the characters most often rhyming with kɛ 佳 in the QSS

Character	*Guǎngyùn* rhyme	Middle Chinese	Late Middle Chinese
差	麻 / 佳	tsrhæ / tsrhɛ(j)	tʂʰa / tʂʰaːj
蛇	麻	zyæ	ʂɦia
華	麻	xwæ	xɦwaː
涯	支 / 佳	ngɛ	ŋjaːj
蛙	麻 / 佳	‘wɛ	ʔwaː
鞋	佳	hɛ	xɦjaːj
沙	麻	sræ	ʂaː
過	戈	kwa	kua
推	脂	thwoj	tʰuaj
車	麻	tsyhæ	tʂʰia

As we conclude our survey of the properties, strengths and limitations of the community detection algorithm as a rhyme annotation strategy, this raises an interesting question: in this poem, is the Community annotator wrong? On the one hand, the intent of the poet is clear and suggests ‘aaaa’; on the other hand, although the poet followed the prescriptive rules of the *Guǎngyùn*, it is clear that his choice of rhymes would have been regarded as unusual, perhaps archaic, especially since he lived in the last days of the Song.[44] Regardless of our position on this particular poem, this suggests that when the Community annotator produces an unexpected pattern, there is usually a strong motivation behind it.

[44] My investigation suggests that, although modern Hokkien and Cantonese both have j-glides for this character, the use of *kɛ* 佳 during the Song was consistently to make it rhyme with *mæ* 麻, regardless of the geographical origin of the poet, with poems rhyming with *mæ* 麻 being 3.6 times more frequent than poems rhyming with *kɛ* 佳 in the North and 3.3 times in the South (given the small size of the samples, 23 poems in the North and 186 in the South, the difference in ratio is not statistically significant). As for the time dimension, as *kɛ* 佳 only appears 9 times in the QTS (7 times with *mæ* 麻, once with *ka* 歌 and once with kɛ 佳), the sample is too small to establish a trend, but it is clear that the change had already started, if not yet completed.

5 Conclusion

In the present work, we have answered the invitation of List et al. to work towards a standard of annotation in rhymed texts. Considering the extremely large corpus available to us, we felt the need to explore the question of automatic and semi-automatic annotation: starting from a formalisation of the concept of annotator, we have moved onto different annotation strategies—of varying complexity—that offer a speed-up in the task. It turns out that a lot of datasets—rhyme books and reconstructions—can easily be turned into automatic annotators, especially if these datasets have been digitised. Going further, we reused List's (2016) idea of using graph community detection algorithms for resolving graph ambiguities and produced an automatic rhyme annotator that relies entirely on the raw corpus and needs neither previous annotations (as opposed to List's use of the algorithm on the *Shījīng*) nor knowledge of phonology. As such, this paper's main contribution is a big step towards a solution to the problem of annotation of arbitrary rhymed corpora. In principle, the approach is language-independent (no assumption is made regarding the corpus's language or period) and could apply to any rhymed material; further work is nevertheless required to test its applicability to more varied corpora than the *shī poetry* of the Tang and Song, and we can anticipate challenges for smaller corpora such as the Shījīng or the less formally structured rhyme patterns of the *cí poetry* of the Song.

The comparison with other annotators has been useful: as the *Guǎngyùn* annotator is prescriptive and the Community annotator is descriptive, poems in which the *Guǎngyùn* annotator produces a regular annotation and the Community annotator an irregular one indicate the rule-conscious, archaising style of the poet. While the case of poems in which the Community annotator performs better than the *Guǎngyùn* annotator seems to be the norm, there are poems in which neither annotator produces the correct output: this is particularly the case with poems in which the rhyme reflects a later stage of the language than the rest of the corpus.

Through these case studies and the insights we can derive from them, we demonstrated that these annotators are not only powerful tools that can significantly speed up annotation efforts and make the analysis of large rhymed corpora finally possible, they are also able to highlight interesting linguistic phenomena from such large corpora.

References

Behr, Wolfgang. ‘Inscriptional Evidence and the Origins of Poetic Form in Early China’. Accessed 4 January 2021. https://www.academia.edu/36662287/Inscriptional_Evidence_and_the_Origins_of_Poetic_Form_in_Early_China.

Boltz, William G. ‘Origin of the Chinese Writing System’. *Encyclopedia of Chinese Language and Linguistics*.

Branner, David Prager, ed. *The Chinese Rime Tables: Linguistic Philosophy and Historical-Comparative Phonology*. Amsterdam Studies in the Theory and History of Linguistic Science, v. 271. Amsterdam; Philadelphia: J. Benjamins Pub, 2006.

Dobson, W.A.C.H. ‘Linguistic Evidence and the Dating of the “Book of Songs” ’. *T’oung Pao* 51, no. 4/5 (1964): 322–334.

Girvan, M., and M.E.J. Newman. ‘Community Structure in Social and Biological Networks’. *Proceedings of the National Academy of Sciences of the United States of America* 99, no. 12 (11 June 2002): 7821–7826. https://doi.org/10.1073/pnas.122653799.

Hú Jiājiā 胡佳佳. ‘Wǎngluò fēnxī fāngfǎ zài yīnyùnxué jiàoxué zhōng de yìngyòng——yǐ “Guǎngyùn” fǎnqiè xìlián wéi lì 网络分析方法在音韵学教学中的应用——以《广韵》反切系联为例 (The Role of Network Analysis in the Teaching of Phonology—Using the Fanqie Connections in the Guyangyun as an Example)’. *Lìyún yǔyánxué Kān* 励耘语言学刊 (*Liyun Linguistics*), no. 2018/2 (2018).

Lǐ Línqīng 李林青. *Fújiàn Liǎngguǎng Táng-Wǔdài Wénrén Shīcí Yòngyùn Bǐjiào Yánjiū* 福建两广唐五代文人诗词用韵比较研究 (*Comparative Research into Rhyme Usage in Shi and Ci Poetry during the Tang and Five Dynasties in Guangdong and Guangxi*). 1 vols. Nanjing: Nanjing Normal University, 2011.

Lín Zhèngsān 林正三. *Mǐnnányǔ Shēngyùnxué* 閩南語聲韻學 (*Southern Min Phonology*). Taipei: Wenshizhe chubanshe 文史哲出版社 (The Liberal Arts Press), 2002.

List, Johann-Mattis. *PoePy. A Python Library for the Quantitative Handling of Poetry*. Zenodo, 2019. https://doi.org/10.5281/zenodo.3252142.

List, Johann-Mattis. ‘RhyAnT: A Web-Based Tool for Interactive Rhyme Annotation’. Billet. *Computer-Assisted Language Comparison in Practice* (blog). Accessed 26 July 2020. https://calc.hypotheses.org/2380.

List, Johann-Mattis. ‘Using Network Models to Analyze Old Chinese Rhyme Data’. *Bulletin of Chinese Linguistics* 9, no. 2 (22 June 2016): 218–241. https://doi.org/10.1163/2405478X-00902004.

List, Johann-Mattis, Nathan W. Hill, and Christopher J. Foster. ‘Towards a Standardized Annotation of Rhyme Judgments in Chinese Historical Phonology (and Beyond)’. *Journal of Language Relationship* 17, no. 1–2 (1 February 2019): 26–43. https://doi.org/10.31826/jlr-2019-171-207.

Liú Xiǎonán 刘晓南. ‘Sòngdài Fújiàn Shīrén Yòngyùn Suǒ Fǎnyìng De Shí Dào Shísān Shìjì De Mǐn Fāngyán Ruògān Tèdiǎn 宋代福建诗人用韵所反映的十到十三世纪的

闽方言若干特点 (A Few Notes on the Rhyme Usage of Song Poets from Fujian and Its Reflection in the Min Dialect of the 10th to 13th Centuries)'. *Yǔyán yánjiū* 语言研究 (*Linguistics Research*) 1998, no. 1 (1998): 155–169.

Pulleyblank, E.G. *Middle Chinese: A Study in Historical Phonology*. Vancouver: University of British Columbia Press, 1984.

Pulleyblank, Edwin G. 'Late Middle Chinese (Part II)'. *Asia Major* 16 (1971): 121–168.

Pulleyblank, Edwin G. 'The Rhyming Categories of Li Ho (791–817)'. 清華學報 / *Tsinghua Journal of Chinese Studies* 7 (1968): 1–25.

Qián Yì 钱毅. 'Sòngdài Jiāng-Zhè Shīrén Yòngyùn De Tōngyǔ Yīnbiàn 宋代江浙诗人用韵的通语音变 (Tongyu Variation in Jiangsu and Zhejiang Poets' Rhyming in the Song Dynasty)'. *Hànyǔ Xuébào* 汉语学报 (*Chinese Language*) 2013, no. 4 (2013): 35–47.

Quán Sòng Shī 全宋詩 (*The Complete Shī Poetry of the Song*). 72 vols. Beijing: Peking University Press, 1998.

Quán Táng Shī 全唐詩 (*The Complete Shī Poetry of the Tang*). 25 vols. Beijing: Zhōnghuá Shūjú 中华书局, 1979.

Rosvall, Martin, and Carl T. Bergstrom. 'Maps of Random Walks on Complex Networks Reveal Community Structure'. *Proceedings of the National Academy of Sciences* 105, no. 4 (2008): 1118–1123.

Sòng Qí 宋祁. 'Qǐ Zhòngdìng Guǎngyùn Zòu 乞重定廣韻奏 (Memorial Begging for the Reinstatement of the Guǎngyùn)'. In Quán Sòng Wén 全宋文, juan 494:314. 上海辭書出版社 (Shanghai Lexicographical Publishing House), 2006.

Sòng Qí 宋祁. 'Xiángdìng Gòngjǔ Tiáozhì Zòu 詳定貢舉條制奏 (Memorial to the Redactors of the Imperial Examination System Regulations)'. In *Quán Sòng Wén* 全宋文, juan 494:316–320. 上海辭書出版社 (Shanghai Lexicographical Publishing House), 2006.

'Sòng běn Guǎngyùn 宋本廣韻 XML'. Accessed 9 February 2022. https://github.com/cjkvi/cjkvi-dict/blob/master/sbgy.xml.

Wáng Lì 王力. *Hànyǔ Shīlǜ Xué* 汉语诗律学 (*Study on Chinese Versification*). Shanghai: Shànghǎi jiàoyù chūbǎnshè 上海教育出版社 (Shanghai Education Publishing House), 1989.

Wáng Zhàopéng 王兆鹏. 'Táng-Sòng Shīgē Bǎntú de Kōngjiān Fēnbù yǔ Wèiyí 唐宋诗歌版图的空间分布与位移 (Spatial Distribution and Displacement of Poetry during the Tang and the Song)'. *Zhōngguó Rénmín Dàxué Xuébào* 中国人民大学学报 (*Journal of Renmin University of China*) 2016, no. 6 (2016): 2–9.

Yǐn Dàizhōng 尹戴忠. 'Liǎng Zhǒng Xìlián "Guǎngyùn" Fǎnqiè Xiàzì de Xīn Fāngfǎ 两种系联《广韵》反切下字的新方法 (Two New Methods of Linking Lower Fanqie Characters from the Guǎngyùn)'. *Zhōngnán Dàxué Xuébào* 中南大学学报 (*Journal of the Central South University*) 21, no. 4 (2015): 244–248.

Yú Nǎiyǒng 余迺永, ed. *Xīnjiào Hùzhù Sòngběn Guǎngyùn Dìnggǎoběn* 新校互註宋本廣韻定稿本 (*A New Critical and Cross-Referenced Definitive Edition of the Songben Guǎngyùn*). Shanghai: Shànghǎi Rénmín Dàbǎnshè 上海人民大版社, 2008.