Computer-Assisted Text Analysis

Johann-Mattis List (University of Passau)

1 Interlinear-Glossed Text

Linguists create linguistic resources with very specific purposes in mind. As a result, a resource can often only be used to address one specific question, although – if it had been carefully designed – it could be used for many additional analyses as well. We have seen enough examples of this lack of interest in the extensibility of resources, or the lack of *integration* as well as the efforts by the CLDF initiative (Forkel et al. 2018) to address these problems. This problem can also be found when dealing with *interlinear-glossed text*.

> Although annotation tools exist [...] their application is difficult due to a lack of cross-platform support [...], but also by a large degree of freedom offered by the respective software. Since the majority of IGT is still produced in research articles, and not in the form of standardized databases, errors in the glossing procedure are still rather common [...]. (List et al. 2021: 4/15)

We have tried to address these problems by providing a first framework that shows how interlinear-glossed text can be handled in standardized CLDF formats and how a consistent integration of interlinear-glossed text resources can help us to retrieve many additional aspects of language data that the original interlinear-glossed text collection might not have been created for initially (ibid.).

Die	**Katze**	**sitz-t**	**auf**	**den**	**Matratze-n.**
ARTIC.NM.SGL.F	cat	sit-3.SG.IND	on	ARTIC.DT.PLR.F	mattress-PLR

The cat sits on the mattresses.

(1)

Word	Gloss
Die	ARTIC.NM.SGL.F
Katze	cat
sitz-t	sit-3.SGL
auf	on
den	ARTIC.DT.PLR.F
Matratze-n	mattress-PLR

(2)

Morpheme	Lexical Gloss	Grammatical Gloss
Die		ART.NOM.SG.F
Katze	cat	
sitz	sit	
t		3.SG
auf	on	
den		ART.DAT.PL.F
Matratze	mattress	
n		PL

(3a)

Lex. Concept	Concepticon
cat	1208 CAT
sit	1416 SIT
on	1741 ABOVE
matress	105 MATTRESS

(3b)

Gram. Concept	Leipzig Glossing Rules
ARTIC	ART
NM	NOM
SGL	SG
PLR	PL
...	...

(4)

Word	CLTS Transcription
Die	d i:
Katze	k a ts ə
sitz-t	s ɪ ts + t
auf	au f
den	d e: n
Matratze-n	m a t r a ts ə + n

(5)

Word	Cognacy
d i:	1
k a ts ə	2
s ɪ ts + t	3 4
au f	5
d e: n	1
m a t r a ts ə + n	6 7

Our workflow for the creation of integrated resources from interlinear-glossed text is shown above. Is it realistic to achieve all of the workflow steps in a completely automatic manner?

2 Rhyme Analysis

Having seen that it pays off, in general, to work on computer-assisted approaches in those cases where large amounts of data have to be handled, we might want to step back a bit from the very specific question of Chinese Historical Phonology and the rhyming practice (discussed in the previous session), and rather ask what we could do if we had a large database of poetry, and what questions we would like to ask. Once we have determined this sufficiently, we should decide what kind of data we want to have. In fact, most of the questions have already been discussed in the first section of this session. The question that now remains for us is how we can actually handle world-wide data on poetry in such a way that we can address these questions? In the following, we will first look at how it is actually being done, and then develop an alternative framework from the problems we observe in the current practice.

What specific questions would you like to ask about the evolution and typology of poetry?

Current Annotation Practice

When analyzing rhymes in poetry, one of the most crucial questions is what rhymes with what and where it rhymes. We can call such an analysis (which is a true analysis, since we may assume that experts commit errors in their assessment of either what the majority of language users think or what the author intended) a *rhyme judgment analysis*, similar to the term *cognate judgment*, which reflects the identification of potential cognate words by experts or algorithms. The ways in which scholars share their respective rhyme judgments in the literature is very diverse and makes a formal comparison of different rhyme analyses difficult. The problem here lies only to some degree in missing digital versions of important contributions, which would be merely a problem for pure computational approaches. A more significant problem is that many authors report their rhyme judgments in a form that is insufficiently explicit to infer the individual judgments made on individual poems and stanzas. Apart from scholars who presented only the results of their analyses, without providing the evidence, we also often find analyses that are extremely difficult to inspect, due to the way they present their judgments. In this sense, only a small amount of rhyme analyses is truly explicit. Among the few explicit rhyme analyses, we again face the problem that scholars differ widely in the formats they use for annotation, and also in the depth of annotation provided.

We have seen before that one can roughly distinguish between *inline* and *stand-off* annotation (Eckart 2012).[1] As an example illustrating the difference between the two annotation styles, consider the rhyme annotation employed by Baxter (Baxter 1992) as compared to the one by Wáng Lì Wáng 1980 [2006], for poem 109 (second part of stanza 2 in the Book of Odes). While Wáng Lì provides the rhyme judgements inline, Baxter (p. 625) basically uses a stand-off annotation by listing all relevant data in tabular form:

[1] While inline annotation manipulates the original data directly, for example, by adding tags, stand-off annotation only references the original data, without directly modifying it. Most annotation frameworks, however, typically use a mixture between the two types, although it is clear that stand-off annotation has the advantage of allowing for far more flexibility, especially if adding multiple layers of annotation to a given resource.

Character	Pīnyīn	MCH	OCH	Rhyme
哉	zāi	tsoj	*tsɨ	B
其	jī	ki	*k(r)jɨ	B
之	zhī	tsyi	*tjɨ	B
之	zhī	tsyi	*tjɨ	B
思	sī	si	*sjɨ	B

彼人是哉(tzə)! 子曰何其(giə)!
心之憂矣,其誰知之(tjiə)?
其誰知之(tjiə)?
蓋亦勿思(siə)! (*Shījīng*, 109.2)

In order to test their algorithm on automated rhyme detection, Haider and Kuhn (2018) uses a corpus in which poems are separated into stanzas, and stanzas are separated into lines, and rhyming is annotated by providing an attribute for each stanza, which reflects which line rhyme with which line, similar to the practice in school, using letters of the alphabet. What huge disadvantage has this system?

Preliminary Framework for Rhyme Annotation

Based on the discussions of the desiderata and past experiments which proved the particular insufficiency of certain annotation forms, the core annotation of a poem or a poem collection, as proposed in (List et al. 2017) now contains the following main components:

- ID: the identifier, which is a numerical ID.
- POEM: a name for the given poem.
- STANZA: the stanza of the poem (usually a numeric value, preceded by the name of the poem).
- LINE_IN_SOURCE: the line of the poem as we find it in the source from which the data is taken (especially containing original punctuation etc.).
- LINE: a double-segmented version of the line, in which words are separated with help of + as a separator, and spaces can be used to represent segments of phonetic values (similar to the format adopted by the LingPy software package to represent phonetic sequences and alignments).
- LINE_ORDER: A numerical value that provides the order of the lines of a poem in a given stanza.
- RHYMEIDS: A list of numerical identifiers, indicating which words in a the LINE rhyme by assigning the same ID to different words, using 0 to indicate that a given word does not rhyme.
- ALIGNMENT: A double-segmented version of the line that can, however, store aligned content, differing from the data in LINE, as well. This data comes in handy when trying to check questions of phonetic similarity of rhyme words, or of vowel purity, which would greatly facilitate automatic analyses as the one presented in List et al. (ibid.).

With these eight columns provided, poems can be annotated in a very straightforward way, regardless of the language in which they were written. One can, of course, add many more columns, depending on specific characteristics of the datasets, but for the general rhyme annotation, we think that these fields will be sufficient for most of the cases; it substantially exceeds rhyme annotation frameworks that have been proposed so far in terms of detail.

What is the obvious drawback of this annotation schema?

PoePy: Python Library for Quantitative Handling of Rhymes

We have developed a software API, called PoePy (`https://github.com/lingpy/poepy`), that allows one to parse, manipulate, and convert files following our new rhyme annotation schema in a convenient way, with help of the Python language. The framework builds heavily on LingPy, a Python library for quantitative tasks in historical linguistics (List and Forkel 2022), as well as SinoPy, a Python library for specialized tasks in Chinese historical lin- guistics (List 2018). The GitHub site of our API offers additional information for installing and using our software library. PoePy can read datasets in our general format mentioned above, it can also be used to align rhyme words, provided they are readily assigned to the data, and it can convert the data to different formats, that ease rhyme pattern inspection. Our stanza 2 from Ode 109 of the Shī- jīng, for example, can be rendered directly in the following tabular form, that greatly facilitates seeing the rhyme structure of the poem.

ID	STANZA	LINE	R:467	R:468
1733	109.2	園 有 **棘**	kiək	
1734	109.2	其 實 之 **食**	djiək	
1735	109.2	心 之 憂 矣		
1736	109.2	聊 以 行 **國**	kuək	
1737	109.2	不 我 知 者		
1738	109.2	謂 我 士 也 罔 **極**	qiək	
1739	109.2	彼 人 是 **哉**		tzə
1740	109.2	子 曰 何 **其**		giə
1742	109.2	其 誰 知 **之**		tjiə
1744	109.2	蓋 亦 勿 **思**		siə

How could the display be further enhanced?

Examples for Annotated Rhymes

As a first example, consider the first stanza of Bob Dylan's song "I want you" (from the album Blonde on Blonde, 1966). Here the rhyme patterns are more complex than in many other poems, but rhyming is in parts also more lax, with more imperfect rhymes, reflecting the typical style of Dylan's poetry.

ID	ST	LINE	R:1	R:2	R:3
1	1.1	The guilty undertaker *sighs*	s - ai s		
2	1.1	The lonesome organ grinder *cries*	k r ai s		
3	1.1	The silver saxophones *say*	s - æi -		
4	1.1	I should *refuse_you*		r i f j u: s j u:	
5	1.1	The cracked bells and washed-out *horns*			h - ɔ r n s
6	1.1	Blow into my face with *scorn,*			s k ɔ r n -
7	1.1	but it's not that way, I wasn't *born*			b - ɔ r n -
8	1.1	to *lose_you*		- - - l u: s j u:	

A further example is the song "Te doy una canción" by Silvio Rodriguez (from the album Mujeres, 1978), in which none of the three rhyme pairs which we have annotated in stanza 1.2 rhymes perfectly. One might thus assume that rhyming was generally not intended in this song, but we find a very similar pattern in stanza 1.4., and songs in which the words *tú* "you" and *luz* "light" co-occur in potential rhyming

position are very frequent in Spanish songs. Our hope is, that with a growing body of datasets in this form, we may learn more about the difference between rhymes which are intended and rhymes which might occur simply by chance.

ID	ST	LINE	R:1	R:2	R:3
7	1.2	Te doy una canción si abro una *puerta*	puer ta		
8	1.2	Y de las sombras sales *tú*		tú	
9	1.2	Te doy una canción de *madrugada,*	madruga da		
10	1.2	Cuando más quiero tu *luz*		luz	
11	1.2	Te doy una canción cuando apareces			
12	1.2	El misterio del *amor*			a mor
13	1.2	Y si no apareces, no me importa:			
14	1.2	Yo te doy una *canción*			can ción

What complicates the problem of finding rhymes that occur by chance and rhymes that were intended by the authors?

RhyAnt: An Interactive Tool for Rhyme Annotation

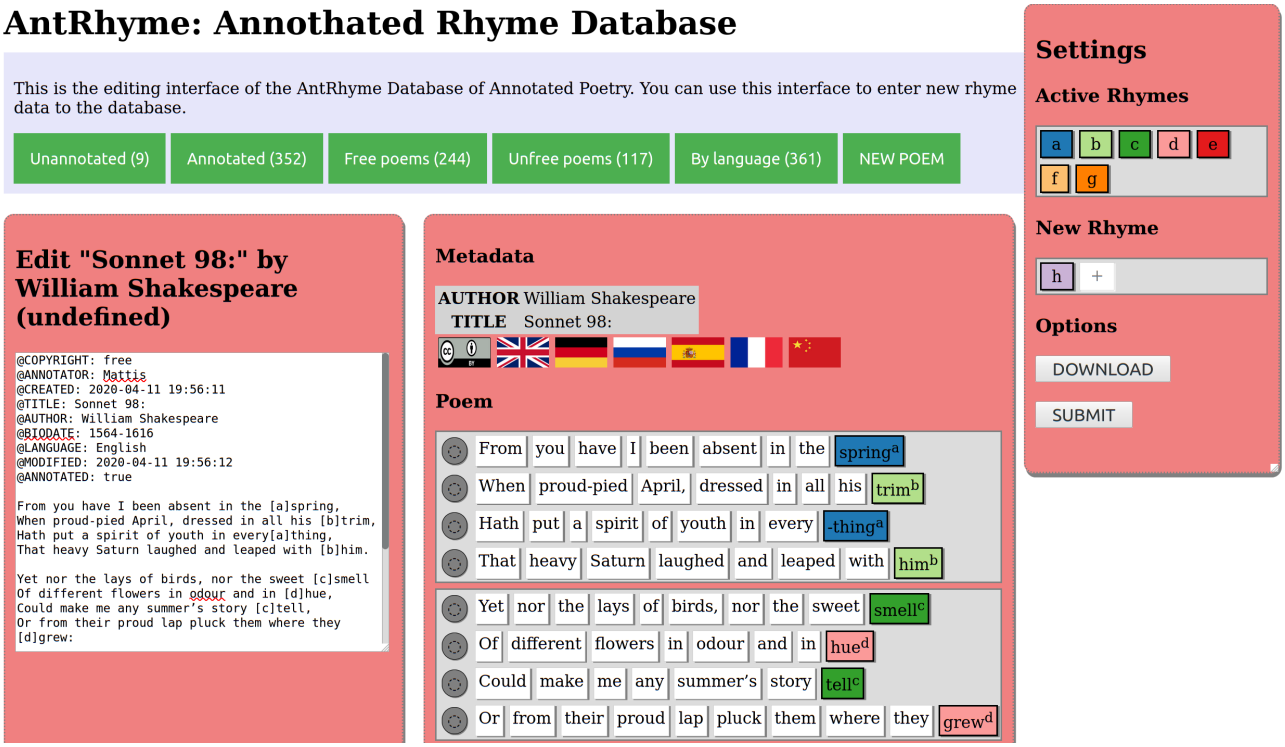
While I consider the inline-annotation format as rather complete by now (with all limitations resulting from inline-annotation), I realized, when trying to annotate poems by using the format, that it is no fun to edit text files in this way. I do not talk about small edits, like one stanza, or typing in some metadata, but annotating a whole rap song can become very tedious and even problematic, as one may easily forget which rhyme tags one already used, or oversee which words have been annotated as rhyming, or forget brackets and the like.

As a result I decided to write an interactive rhyme annotation tool which supports the inline-annotation format and can be edited both in the text and interactively at the same time, a bit similar to the text processing programs in blogging software which allow to write both in the HTML source and in a more convenient version that shows you what you will get.

This tool is now already online available (`https://digling.org/rhyant`, List 2020d). I call it RhyAnT, which is short for *Rhyme Annotation Tool*, and I have been using the tool in combination with a small server to populate a first database with rhymes in different languages that contains by now already more than 400 annotated poems (*AnTRhyme*, `https://digling.org/antrhyme`). This database can be accessed and inspected by everybody interested from its URL, but copyrighted texts from modern songs can – unfortunately – not be rendered by now (as I am not sure how many lines of them I would be allowed to share).

RhyAnt is freely available in the form of an interactive web application written in HTML/CSS and JavaScript. It can be used by opening the website `https://digling.org/calc/rhyant/`, or by downloading the code and opening the website offline with the help of a web browser. The tool is curated on GitHub, where it can also be downloaded (*https://github.com/digling/rhyant/*).

Is it a good idea to work with text that is not phonetically transcribed here?

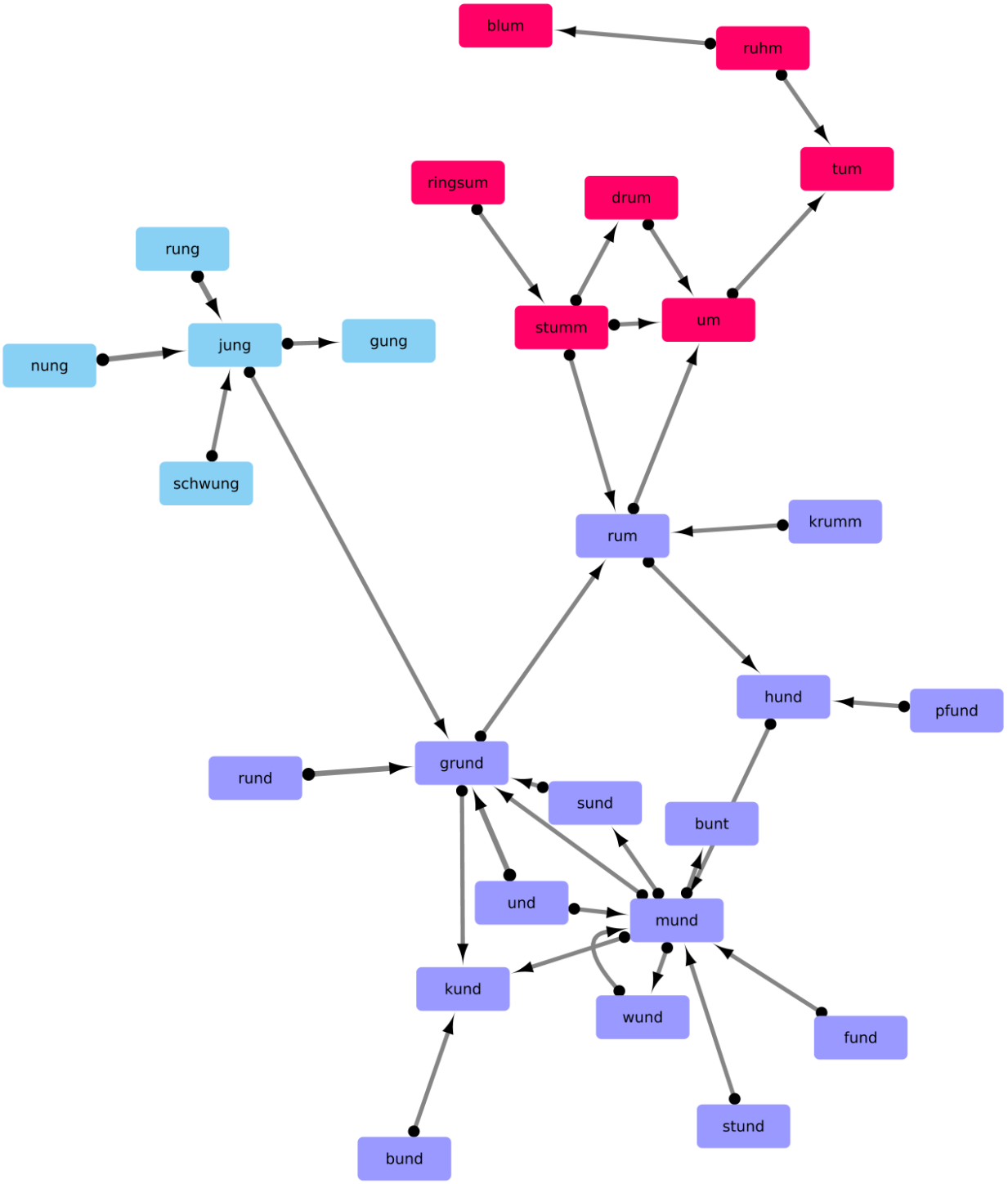


An Initial Analysis of Rhymes in AntRhyme

While the analysis of rhyme networks in Chinese can be considered as quite advanced now, with quite a few examples having been published by now and active and in part ongoing discussions (see Wáng 2020 and List 2020c), network approaches to rhyming in languages other than Chinese are lacking. The reason can be found not only in the lack of data (rhymes are abundantly available for many languages in the world), but rather in the fact that the modeling of rhyme patterns needs to be advanced and that the inference of patterns cannot be done in the same straightforward way as it is done in Chinese, where one can naively assume that the last word of a row in a stanza always rhymes (BALEY 2022).

However, with the help of RhyAnT as an annotation tool for rhymes and with the extended more detailed schemas for rhyme annotation introduced along with RhyAnT and before (List et al. 2019), initial analyses of rhymes in German can be made. As an example and a proof-of-concept, I published a concrete example on rhyming in German, based on the small AntRhyme corpus, in which rhyme patterns are manually annotated (List 2020a, List 2020b).

The rhyme network below is taken from the analysis of the AntRhyme corpus. What normalizations have been carried out in order to make the data comparable?

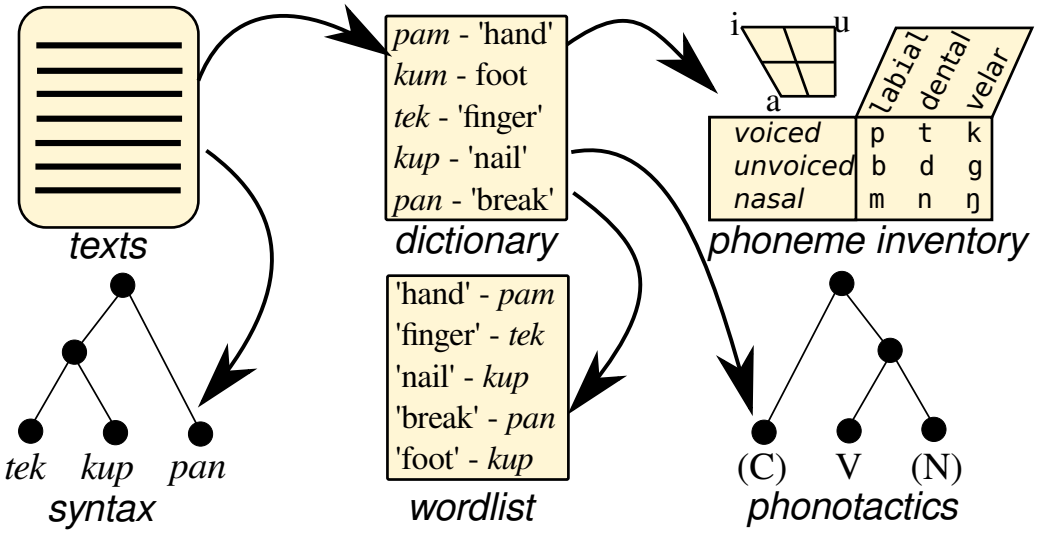


3 Outlook

There are many more topics that can be investigated in the future. For both the topic of interlinear-glossed text and the topic of general rhyme analysis, there is some hope that we can further advance them by increasing the availability of corpus data in standardized form. For interlinear-glossed text, concrete examples are currently being developed as part of CLDFViz (Forkel 2021, with new modules

handling various forms of text in MarkDown, developed by R. Forkel, see `https://github.com/cldf/cldfviz`) and `pylingdocs` (Matter 2022, see `https://github.com/fmatter/pylingdocs`). Apart from this, I hope that we will manage to provide many more examples for *integrated* data, that is, data that does not only provide one type of information, but includes multiple types of information on various aspects of the same language variety which are interdependent and interlinked or even automatically derived from each other. In addition, we hope to be able to provide CLDF examples for all kinds of data used in rhyme analysis and in Chinese Historical Phonology.

Our integration goal for linguistic data is described below in the graphic. Is integration also important for your specific research topic?



References

BALEY, J. (2022). “Leveraging graph algorithms to speed up the annotation of large rhymed corpora.” *Cahiers de Linguistique Asie Orientale* 51.1, 46 –80.

Baxter, W. H. (1992). *A handbook of Old Chinese phonology*. Berlin: de Gruyter.

Eckart, K. (2012). “Resource annotations.” In: ed. by A. Clarin-D. Berlin: DWDS, 30–42.

Forkel, R. (2021). “CLDFViz. A python library providing tools to visualize data from CLDF datasets [Software Library, Version 0.5.0].”

Forkel, R., J.-M. List, S. J. Greenhill, C. Rzymski, S. Bank, M. Cysouw, H. Hammarström, M. Haspelmath, G. A. Kaiping, and R. D. Gray (2018). “Cross-Linguistic Data Formats, advancing data sharing and re-use in comparative linguistics.” *Scientific Data* 5.180205, 1–10.

Haider, T. and J. Kuhn (2018). “Supervised rhyme detection with Siamese recurrent networks.” In: *Proceedings of Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*. (Santa Fe, 08/25/2018), 81–86.

List, J.-M. (2018). *SinoPy: A Python library for quantitative tasks in Chinese historical linguistics*. Version 0.3.1. URL: `https://github.com/lingpy/sinopy`.

— (08/24/2020a). “Analyzing rhyme networks (From rhymes to networks 6).” *The Genealogical World of Phylogenetic Networks* 9.9.

— (08/24/2020b). “Constructing rhyme networks (From rhymes to networks 5).” *The Genealogical World of Phylogenetic Networks* 9.8.

— (2020c). “Improving data handling and analysis in the study of rhyme patterns.” *Cahiers de Linguistique Asie Orientale* 49.1, 43–57.

— (2020d). *RhyAnT. A tool for interactive rhyme annotation*. Jena: Max Planck Institute for the Science of Human History.

List, J.-M. and R. Forkel (2022). *LingPy. A Python library for quantitative tasks in historical linguistics [Software Library, Version 2.6.9]*. Leipzig: Max Planck Institute for Evolutionary Anthropology.

List, J.-M., S. J. Greenhill, and R. D. Gray (2017). “The potential of automatic word comparison for historical linguistics.” *PLOS ONE* 12.1, 1–18.

List, J.-M., N. W. Hill, and C. J. Foster (2019). “Towards a standardized annotation of rhyme judgments in Chinese historical phonology (and beyond).” *Journal of Language Relationship* 17.1, 26–43.

List, J.-M., N. A. Sims, and R. Forkel (2021). “Towards a sustainable handling of interlinear-glossed text in language documentation.” *ACM Transactions on Asian and Low-Resource Language Information Processing* 20.2, 1–15.

Matter, F. (10/2022). *pylingdocs [Version 0.0.10]*. Version 0.0.10.

Wáng, L. 王力. (1980 [2006]). *Hànyǔ shǐgǎo* 漢語史稿 [History of the Chinese language]. Repr. Běijīng 北京: Zhōnghuá Shūjú 中华书局.

Wáng, Z. (2020). “A linguistic study on rhyming in the Beijing dialect.” *Cahiers Linguistiques Asie Orientale* 49.1, 21–42.