Digital Scholarship in the Humanities, 2024, **00**, 1–22
https://doi.org/10.1093/llc/fqae001
Full Paper

OXFORD

Understanding poetry using natural language processing tools: a survey

Mirella De Sisto[1,*], Laura Hernández-Lorenzo [2], Javier De la Rosa[3], Salvador Ros[2], Elena González-Blanco[4]

[1]Tilburg University, The Netherlands
[2]National Distance Education University, Spain
[3]National Library of Norway, Norway
[4]IE University, Spain

*Corresponding author. Tilburg University, Warandelaan 2, 5037 AB Tilburg, The Netherlands. E-mail: m.desisto@tilburguniversity.edu

Abstract

Analyzing poetry with automatic tools has great potential for improving verse-related research. Over the last few decades, this field has expanded notably and a large number of tools aiming at analyzing various aspects of poetry have been developed. However, the concrete connection between these tools and traditional scholars investigating poetry and metrics is often missing. The purpose of this article is to bridge this gap by providing a comprehensive survey of the automatic poetry analysis tools available for European languages. The tools are described and classified according to the language for which they are primarily developed, and to their functionalities and purpose. Particular attention is given to those that have open-source code or provide an online version with the same functionality. Combining more traditional research with these tools has clear advantages: it provides the opportunity to address theoretical questions with the support of large amounts of data; also, it allows for the development of new and diversified approaches.

Keywords: Poetry analysis; Natural Language Processing; Computational Literary Studies.

1. Introduction

The field of automatic poetry analysis is expanding, allowing for the application of automatic tools in verse investigation. This enables new quantitative approaches to address traditional and theoretical questions.

The advancement of Natural Language Processing and text analysis technologies has significantly contributed to the growth of automatic poetry analysis. Multiple and varied tools have been developed, each addressing different aspects of verse with specific goals, such as metrical annotation or the visualization of poetic features. While most tools focus on analyzing English, an increasing number of programs now target other languages and language varieties.

These tools empower scholars to address theoretical questions with the support of large-scale data analysis. However, the development of computational tools often remains disconnected from the theoretical research that could benefit from them. The present article aims to bridge this gap by providing a survey of currently available computational tools for automatic poetry analysis. In this survey, we focus on tools for the analysis of the structural characteristics of poetry of European languages.

Since most tools are not publicly available, a comprehensive qualitative comparison was not feasible. In addition, such a comparison would require expertise in multiple poetic traditions, which goes beyond the possibilities of a single research group. Therefore, this article aims to offer an exhaustive overview of available tools for European languages, while empirical testing falls beyond its scope.

The article is structured as follows: Section 2 briefly introduces automatic poetry analysis and describes its advantages for poetry-related research. Section 3, along with its subsections, classifies the tools presented in this article based on language (3), purpose and features (4), and algorithmic nature (i.e. rule-based or data-driven) (5). A concluding section summarizes the key points of the review and highlights the importance of bridging the gap between traditional and computational versification studies. Appendix Table A1 provides a detailed overview of the classification and includes the links to the tools' source code and graphical user interface.

2. Automatic poetry analysis

In the last few decades, the use of automatic tools for analyzing poetry has grown notably. An important factor contributing to this growth is the development of new technologies and, in particular, advancements in the field of Natural Language Processing.

Automatic poetry analysis offers clear advantages for literary and linguistic research. Researchers gain access to large amounts of data that would otherwise be unavailable or require the effort of many people for a long stretch of time to compile and analyze.

Furthermore, the use of such tools also presents advantages that vary depending on the type of tool considered. For instance, they can provide a diachronic perspective on formal aspects of poetry or a stylistic analysis on often overlooked poetic language markers. These tools can be employed to empirically test claims made in previous traditional studies. Studies by Navarro-Colorado (2015), Ruiz Fabo (2020) and Hernández-Lorenzo *et al.* (Hernández-Lorenzo *et al.* 2021) have tested and corroborated the claim regarding the Spanish Renaissance hendecasyllable being predominantly *a maiore* (Henríquez Ureña 1919; Domínguez Caparrós 2020). Moreover, at the epistemological level, automatic annotations of large corpora can be viewed as performed by a single "hand," which is not possible in classical manual approaches. In fact, working with big corpora annotated by a group of people requires precise guidelines in order to avoid inter-annotator disagreement. Nevertheless, discussing ambiguous cases is needed. Given that poetry is usually open to interpretation, the annotation of one annotator might differ from that of another. On the contrary, in automatic annotation all material is annotated by a single annotator, the machine. Once the model has been evaluated, it will always make the same choices and interpretations along the whole corpus. As most systems are deterministic, this minimizes the risk of having two poems be metrically different because of the choices of its automatic annotator. Similarly, automatic tools for poetry visualization enhance close reading (Fisher 2013) or complement distant reading perspectives, as proposed by Moretti (Moretti 2013); this type of innovative approach allows for a more detailed and rich investigation of poetry. However, not all tools are easily available to scholars and only a few have a graphical user interface.

The present survey specifically focuses on tools that analyze the structural characteristics of verse, rather than its content. For this reason, we decided not to include tools that relate to the meaning of poetic texts. Furthermore, we chose tools for Indo-European languages as our area of focus, including historic languages such as Latin, Ancient Greek, Middle High German, etc. Therefore, our survey does not contain the description of tools developed for other languages and varieties. Basque and Finnish are the only non-Indo-European language considered. The reason for this exception is that the tools annotating Basque and Finnish poetry are already included in this study, since they are also used to perform annotation of English and Spanish verse, and English verse, respectively.

Our selection method aimed to be as exhaustive as possible. We started from the most recent conference on Digital Humanities 2020,[1] and selected all abstracts based on a keyword search for poetry-related terms: for example, poetry, poem, verse, meter, rhyme, metrical annotation, and poetry scansion. From there, we went backward in time reading all bibliographies from those relevant papers, in an effort to encompass every approach in a timeline.

As a result of our survey, we could infer that not all languages and poetic traditions have received equal attention in the development of automatic tools for poetry analysis. Some languages, such as English, have a large number of existing tools; other languages are underrepresented, such as Italian, for which only one tool seems to be available.

Lastly, the time range of our survey expands from the late 1980s to mid-2020, covering 40 years of advances and evolution of tool creation. As shown in Fig. 1, the earliest automatic tools for poetry analysis date back to the end of the last century—especially the nineties—whereas many of them have been created in the last decade.

In the next section, we propose a classification of the tools based on their general characteristics, such as the languages they are developed for, their purpose, and the type of technology they employ. Online documentation and code repositories are listed in Appendix Table A1.

3. The languages of poetry analysis tools

The current survey of tools covers seventeen languages and language varieties. We focused on Indo-European languages and their existing tools. The only two exceptions are Basque and Finnish. English is strongly represented with twenty-three tools. The second most represented language is Spanish, with eight tools. Table 1 displays the tools considered in this article divided by language and language variety.

4. Purpose of the tools

Three groups can be outlined based on the general purpose for which the tools were developed: (1) tools for metrical annotation, namely those that annotate poetry in terms of stress, line length, rhyme, or rhythm; (2) tools for poetry recognition, that is, tools that determine whether a given text is poetry or not; and (3)

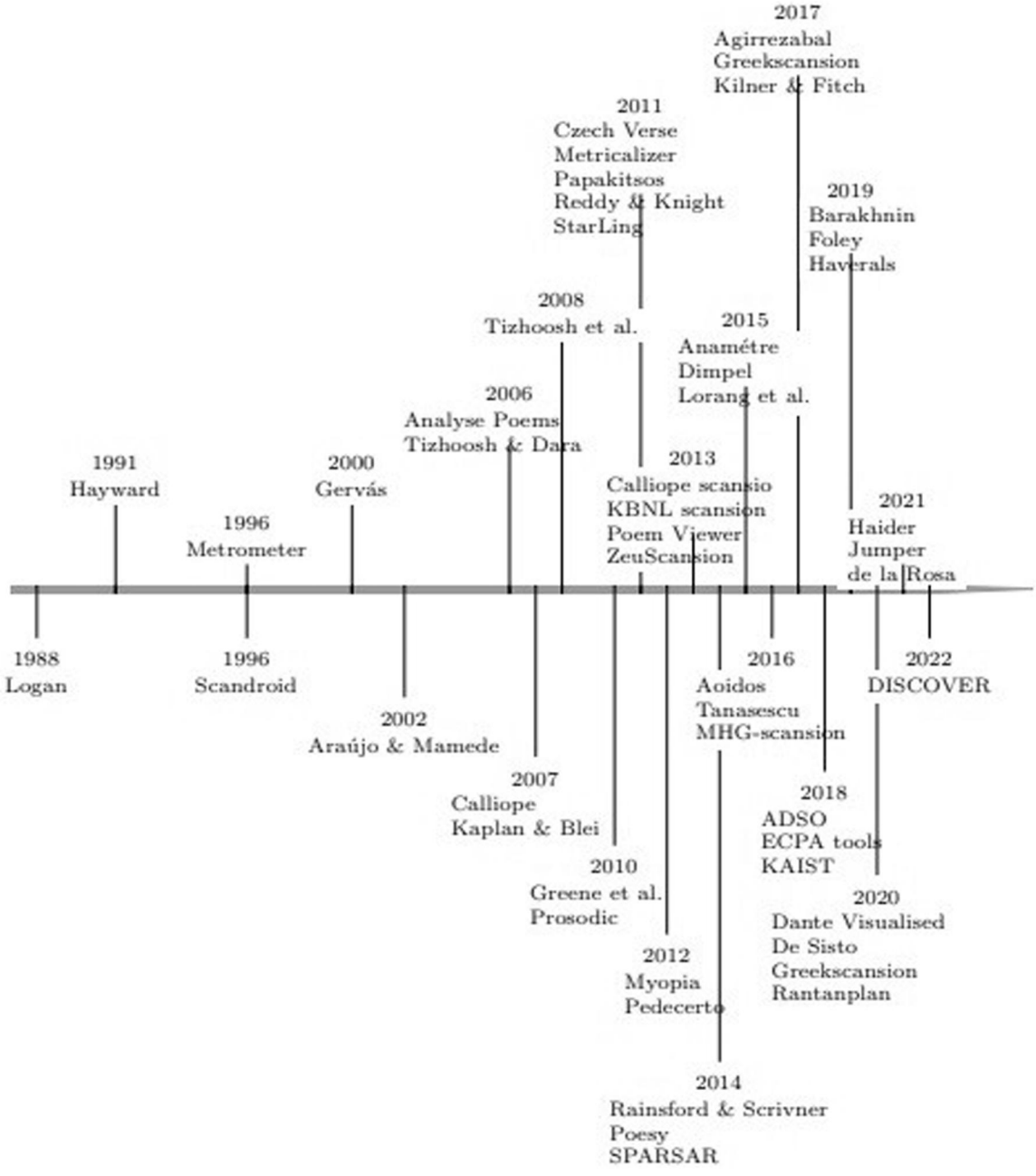

Figure 1. Timeline chart showing the development of automatic tools for poetry analysis. All the tools are classified and explained in this article.

tools for poetry visualization, which aim to enhance close-reading by highlighting various details of the line, for example, recurrent sounds in a line.

The following sections provide a survey of the tools based on this classification and offer some details about the specific focus and features of each tool, including a brief explanation of their functioning. Some tools span two groups and are classified according to their primary function.

4.1 Tools for metrical annotation

The first and the largest macro-group considered in the present survey consists of tools that focus on metrical annotation. These tools are primarily concerned

Table 1 Tools described in the current paper organised by language and language variety.

Language	Tools	
English	Logan (1988) Hayward (1991) Scandroid (Hartman 1996) AnalysePoems (Plamondon 2006) Tizhoosh, Sahba, and Dara (2008) Calliope (McAleese 2007) Kaplan and Blei (2007) Prosodic (Heuser *et al.* 2010) Reddy and Knight (2011) Myopia (Chaturvedi *et al.* 2012) Poem Viewer (Abdul-Rahman *et al.* 2013)	Poesy (Algee-Hewitt *et al.* 2014) ZeuScansion (Agirrezabal *et al.* 2013) SPARSAR (Delmonte 2014) Lorang *et al.* (2015) Tanasescu, Paget, and Inkpen (2016) Agirrezabal, Alegria, and Hulden (2017) Kilner and Fitch (2017) KAIST (Kim, Zhao, and Zelalem 2018) ECPA tools (Huber 2018) Foley (2019) Haider (2021) De la Rosa *et al.* (2021)
Spanish	Gervás (2000) ZeuScansion (Agirrezabal *et al.* 2013) ADSO (Navarro-Colorado 2018) Agirrezabal, Alegria, and Hulden (2017) Rantanplan (De la Rosa *et al.* 2020b)	Jumper (Marco *et al.* 2021) De la Rosa *et al.* (2021) DISCOVER (Bermúdez Sabel, Ruiz Fabo, and Martínez Cantón 2023)
Ancient Greek	Papakitsos (2011)	Greek_scansion (Conser, 2017) Greek_scansion (Schumann, 2020)
Basque	Agirezabal (2017)	
Czech	Automatic analysis of Czech verse (Ibrahim and Plecháč 2011)	
Dutch	KBNL-scansion (Koppelaar and Van Oostendorp 2013)	
Middle Dutch	Middle Dutch (Haverals et al. 2019; Haverals, 2020)	
Renaissance Dutch	De Sisto (2020)	
Finnish	Prosodic (Heuser *et al.* 2010)	
French	Metrometer (Beaudouin and Yvon 1996) Reddy and Knight (2011)	Anamétre (Delente and Renault 2015)
Middle High German	AMM 2.0 (Dimpel, 2015)	MHG-scansion (Estes and Hench 2016)
German	Metricalizer (Bobenhausen 2011) Haider (2021)	De la Rosa *et al.* (2021)
Italian	Dante Visualised (Ferraro 2020)	
Latin	Calliope scansio (Jacquet 2013)	Pedecerto (Colombi *et al.* 2012)
Old Occitan	Rainsford and Scrivner (Rainsford and Scrivner, 2014)	
Portuguese	Aoidos (Mittmann, von Wangenheim, and dos Santos 2016)	Araújo and Mamede (2002)
Russian	StarLing (Pilshchikov and Starostin 2011)	Barakhnin, Kozhemyakina, and Borzilova (2019)

with defining the metrical features of verse. The specific features considered may vary depending on the poetic tradition being analyzed. For example, word stress and the number of syllables per line are essential elements in accentual–syllabic verse (e.g. English iambic pentameter) and play an important role in syllabo–tonic traditions (e.g. Spanish hendecasyllable) as well. On the other hand, for syllabic poetry, which is only defined by the number of syllables per line (e.g. French alexandrine), detecting the number of syllables might be the main focus of the tool. Foot-based poetry (e.g. Ancient Greek verse) needs to be analyzed with a tool able to identify feet. Other tools annotate meters which are either quantitative (e.g. Latin hexameter) or a hybrid between quantitative and accentual verse (e.g. Middle High German trochaic tetrameter); in those cases, syllable quantity also needs to be considered. In addition, metrical features like stanza division and rhyme scheme can provide valuable insights into the characteristics of poetry.

The tools described in this section are designed to detect and annotate these features based on the specific poetic traditions they are developed for. They do so by applying different types of algorithms or different techniques. Nevertheless, they all share a common goal: to understand the structure of a verse and annotate it according to its metrical pattern and characteristics.

4.1.1 English

One of the first metrical annotation tools developed is the tool by Logan (1988), which utilizes sound pattern detection for metrical annotation. In addition to syllabification and stress assignment, this tool identifies the recurring sounds within the lines and their patterns across lines. It also considers the distinctive features of sounds (i.e. + or—labial, + or—coronal, etc.) both on a per-line basis and for the entire poem. By comparing the generated metrical annotation with the metrical pattern of an iambic pentameter, the tool calculates metrical complexity.

The connectionist model developed by Hayward (Hayward 1996, 1991) analyzes poetry in order to measure its metricality, which is a concept that has been claimed to be gradient (Hayes, Wilson, and Shisko 2012). It is still not clear at which degree of deviance from the canon a line stops being metrical. The tool aims to identify the boundary between metricality and unmetricality by using Artificial Neural Networks with a parallel distributed processing model (McClelland and Rumelhart 1988). The annotation considers stress on the intonational, lexical, prosodic, syntactic, and interpretative levels and aims to account for their interaction. The annotations for the different levels are merged into a calculation of the potential activation of metrical stress of each position. This tool is built with a bias toward iambic reading, therefore, when annotating a line, it always tries to match it to such rhythm.

Scandroid (Hartman 1996) is a tool for scansion of iambic and anapestic poetry (and has a bias for these meters as a result). If a poem is written in a different meter, the program forces its lines into one of the two possible scansions. When analyzing a line, it first tries to determine whether it is formed by iambic or anapestic feet and, subsequently, calculates line length. Initially, it divides the line into words and then syllabifies words and assigns stress to syllables based on a built-in dictionary and on general principles. The tool can be downloaded from its website, which also provides a tutorial, a manual and the source code.[2]

AnalysePoems (Plamondon 2006) is developed using the poetry of Representative Poetry online,[3] a web anthology of 4,800 poems in English and French. AnalysePoems employs syllable decomposition, stress placement, and rhyme, in order to identify the dominant meter and the rhyme scheme of a poem. Word stress assignment is checked in a lexicon database and, in the case that a word is not in the database, the user is asked to provide the information needed. AnalysePoems also verifies whether line length is stable in the poem. Even though it is not based on a specific fixed meter, the tool assumes a regular metrical form.

Built on Scandroid, Calliope (McAleese 2007) implements metrical scansion by using syntactic information, namely stress assignment derived from syntax, and phonological scansion based on syntax. It uses the Antelope Natural Language Process Parser (Chaumartin 2013) for parsing the text into syntactic elements. Stress is assigned using a lexicon and phonological grouping is performed. The output is parsed by a phonological scansion algorithm, and, subsequently, consistency of the result is assessed. The tool has been tested with iambic and trochaic meters, and with free verse.

In the second decade of the 2000s, there was a growing interest in automatic metrical annotation tools, leading to a significant increase in the variety of tools for analyzing poetry. Greene, Bodrumlu, and Knight (2010) developed a tool for metrical annotation, generation, and translation of poetic texts, with a focus on English rhythmic poetry. They extract word-stress patterns from the corpus sonnets.org, an archive of English sonnets,[4] using unsupervised learning. Since a pronunciation dictionary of contemporary English could not handle the language of part of the sonnets in the corpus (i.e. Shakespeare's 1600 English), the model extracts word stress without relying on a pronunciation dictionary. Poetry generation is based on a finite-state cascade composed of three models: a user-supplied model which generates a single string of metrical strong and weak positions; a model that selects words based on the required stress (augmented with a pronunciation dictionary); a word-trigram model built from a corpus consisting of 10,000 lines of English love poems. Finally, another finite-state cascade is employed in order to implement translation to the model with the aim of providing coherence of meaning to the generated text. To explain, the poetic input from a different language, Italian in this case, is used to train a statistical phrase-based machine translation (PBMT) system Moses (Koehn *et al.* 2007); the lattice generated by the PBMT is sent to the model trained for poetry annotation; finally, the resulting syllable sequences are filtered with a single-path deterministic iambic pentameter acceptor.

Prosodic (Heuser *et al.* 2010) is a tool that focuses on metrical and phonological parsing of poetry. Its scansion process begins with tokenization of the text into words, which are then converted into stressed, syllabified phonetic transcriptions. By default, Prosodic uses the CMU pronunciation dictionary (Weide 1998) to retrieve information about stress patterns and syllable boundaries. For the metrical parsing of words not available in the CMU pronunciation dictionary, a text-to-speech (TTS) engine[5] is used. Each line is assigned its best available metrical parsing based on a set of constraints, following the principles of Optimality Theory (Prince and Smolensky 1993), which can be

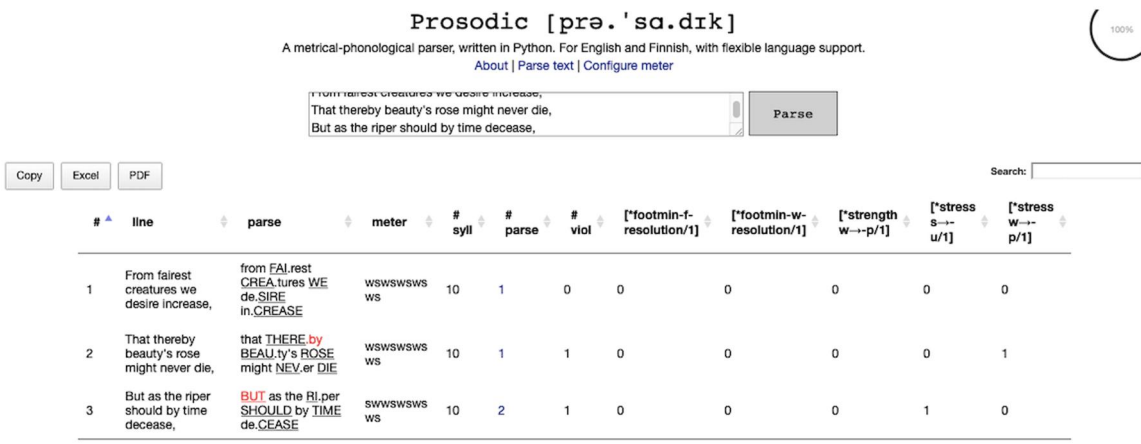

Figure 2. Metrical annotation with Prosodic.

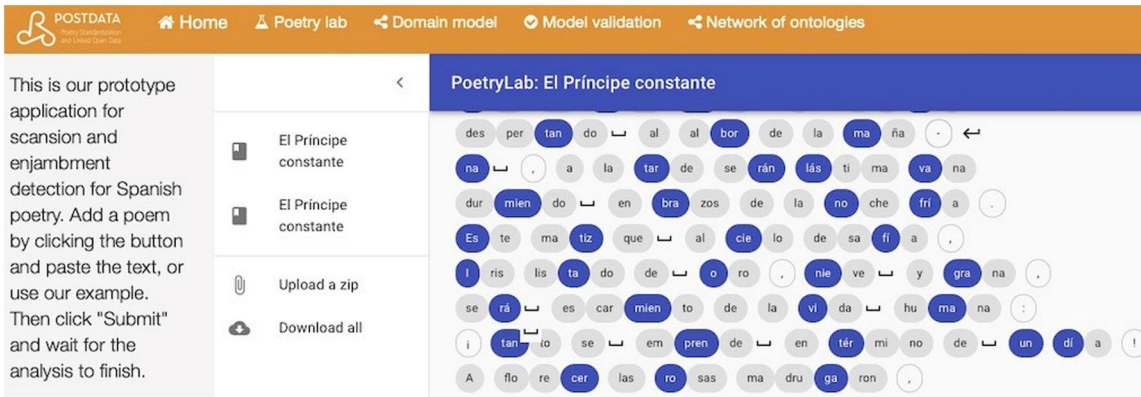

Figure 3. Metrical annotation with Rantanplan and enjambements detection with JollyJumper in PoetryLab.

defined by the user. The default metrical constraints are based on those proposed by Hanson and Kiparsky (Hanson and Kiparsky 1996), such as *strength* (a weak syllable should not be in a strong position and a strong syllable should not be in a weak position), *stress* (an unstressed syllable should not be in a strong position and, vice versa, a stressed syllable should not be in a weak position) and *minimal foot* (a disyllabic metrical position should not contain more than a minimal foot). Constraints can be adjusted to different frameworks or metrical forms. Prosodic is available in a code repository, which also contains two English and Finnish corpora.[6] Aside from its source code being available, Prosodic is available as an online platform, which gives the possibility of configuring the meter by adjusting metrical parameters and constraints. The line analysis can be downloaded as a PDF or Excel format. Figure 2 is an example of the annotation of three lines by using Prosodic.

ZeuScansion (Agirrezabal *et al.* 2013) is a rule-based tool that annotates poetry by defining line stress patterns. It also attempts to identify the dominant meter of a poem and the metrical feet constituting it. The tool performs tokenization, stress assignment via Part-of-Speech (POS)-based rules, looks up words in a pronunciation lexicon, and divides lines into feet. The evaluation of ZeuScansion relies on the gold-standard data from For Better For Verse,[7] a corpus of about 100 poems manually annotated in terms of weak and strong positions. The poems of For Better For Verse can be downloaded from a code repository.[8] ZeuScansion's source code is publicly available in a code repository, where instructions on installation and utilization are provided.

Built on Prosodic, Poesy (Algee-Hewitt *et al.* 2014) is a tool that generates a metrical annotation of a poetic text and detects rhyme patterns. This tool, apart from syllabification, also groups syllables into feet. Its

metrical parsing and annotation procedures depend on Prosodic. The tool is available in a code repository.[9]

The combination of metrical annotation and TTS technology has led to the creation of SPARSAR (Delmonte 2014). SPARSAR is a rule-based system aiming to represent and analyze poetic devices in English and Italian. However, it is only evaluated, though, with English corpora. The system utilizes a modified version of VENSES, a semantically oriented NLP pipeline for deep text understanding. SPARSAR has three modules: the first operates at the sentence level and provides syntactic and semantic analysis; the second is a rule-based system that converts each poem into phonetic characters, divides words into stressed and unstressed syllables, and computes rhyming schemes at line and stanza levels. The third module uses these data to generate a parameterized version of the poem which can be read by a TTS engine. SPARSAR's output includes a visualization and allows for the comparison of poems from different authors using the Pearson correlation coefficient. In addition, it includes a technology for TTS reading of poems. The tool can be downloaded from the project's website,[10] which, besides documentation, also provides examples of analyzed poems and of the generated graphics. However, the platform does not appear to be regularly updated.

Tanasescu , Paget, and Inkpen (2016) developed a tool for poetry classification partially based on Scandroid. The tool uses prosodic and phonetic features to identify meter and rhyme. Lexical stress detection is based on the Scandroid code. For classification, they use a decision tree algorithm (J48) and a bootstrap aggregating machine learning technique (Breiman 1996). The source data of this tool is the poetry available on the online platforms Poetry Foundation[11] and sonnets.org.

Similarly, Agirrezabal and his colleagues (Agirrezabal 2017; Agirrezabal, Alegria, and Hulden 2017) use Recurrent Neural Networks (RNN; Schmidhuber 1989) with bidirectional Long Short-Term Memory (bi-LSTM) cells (Hochreiter and Schmidhuber 1997) and Conditional Random Fields (CRF; Lafferty, Abdul-Rahman, and Pereira 2001) to automatically scan poetry in three languages, that is, English, Spanish, and Basque. The tool tokenizes words using the IXA pipeline[12] (Agerri, Bermudez, and Rigau 2014) and then tags POS. Stress is assigned according to an implementation of Groves' rules (Groves 1998) and using the CMU pronunciation dictionary (Weide 1998) in combination with NETtalk (Sejnowski and Rosenberg 1987). In the case that a word is not found in the dictionaries, its stress pattern is guessed by searching for similar entries in the NETtalk lexicon and in the CMU dictionary. Finally, syllables are grouped into feet. Like ZeuScansion (Agirrezabal *et al.* 2013), this tool also employed the For Better For Verse corpus for English for training and the Corpus se Siglo de Oro (Navarro-Colorado 2015) for Spanish.

Kim, Zhao, and Zelalem (2018) from the KAIST research group developed an automatic tool for poetry classification. Their model extracts syllables by referring to the WebCelex dataset and scans lines based on the manually annotated training corpus; aside from special characters (e.g. colons and full stops) and syllables, word boundary is encoded as a character on the syllable. The tool also determines whether a line is metrical or not. The data used for developing this tool is, as for the tool by Tasnasescu *et al.* (2016), the poetry available on Poetry Foundation. The source code is publicly available in a code repository.

Although these tools are all developed to provide metrical annotations of English poetry, they do present some significant differences. The major distinction is between tools developed based on a particular meter (e.g. iambic) and those which are not developed for a specific poetic form. Tools developed by Logan (1988), Hayward (1991), Hartman (1996), and Tanasescu, Paget, and Inkpen (2016) have a bias for iambic meter, making their annotation more accurate with iambic poetry than with other metrical patterns. The same could count for Greene *et al.* (Greene, Bodrumlu, and Knight 2010), since it was trained on Shakespeare's sonnets. On the other hand, Prosodic (Heuser *et al.* 2010) has a set of constraints which can be adjusted, and the other tools are not based on a specific meter.

Another difference concerns rhyme detection. Only Logan (1988), AnalysePoems (Plamondon 2006), Poesy (Algee-Hewitt *et al.* 2014), and Tanasescu, Paget, and Inkpen (2016) provide information about rhyme patterns.

4.1.2 Spanish

Gervás' tool (Gervás 2000) for Spanish poetry is also one of the earliest automatic annotation tools. It presents a logic programming application for the metrical annotation of Spanish poems. The system uses Definite Clause Grammars to model word syllabification, as well as additional predicates to define synaloepha, syllable count, and rhyme. The logic programming implementation, using SWI Prolog, treats verses as lists of words, and words and syllables as lists of characters. A predicate analysis retrieves from each verse the number of syllables, a list of syllables in the verse, a list of positions of stressed syllables, and the verse's rhyme (which is used to define the rhyme pattern of the stanza). It also includes a system vocabulary, referred to as 'a database of facts' (see Gervás 2000: 1335), containing word information. If data are missing for a given word, a set of rules is used to fill them. After checking if the data are available or need to

be produced, a Word Syllable Parser module is used to parse character types, character groups and syllables, locate the stressed syllables of a word, and extract the rhymes of a word. Additionally, a Metrical Syllable Counter counts the metrical syllables in a verse. The Metrical Syllable Counter consists of a set of Prolog predicates that operate recursively over the list of words in a verse, with another predicate to evaluate if the verse is a hendecasyllable. However, the author acknowledges that this application makes significant errors with the character "y" and has problems with some diphthongs.

On the rule-based side, the ADSO Scansion system (Navarro-Colorado 2018) is a tool for Spanish poetry that uses POS tags for syllabification and stress assignment. First, POS tags (using FreeLing),[13] Padró and Stanilovsky (2012) are added to the words of every line in a poem. Then, each word is split into syllables, the POS tag is used to classify syllables as stressed or unstressed. As the system is prepared for hendecasyllable detection, a syllable counter is applied. Only when the verse length is shorter or longer than eleven syllables, diaeresis and synaloephas are applied. ADSO's source code is available in a code repository. The tool was developed on the *Corpus de Siglo de Oro*[14] (Navarro-Colorado 2015), the only available manually annotated corpus of Spanish poetry, which contains sonnets from the 16th and the 17th centuries. The poems are in XML format, meeting TEI standards. Additionally, each line has its metrical pattern indicated in the annotation. The metrical annotation is obtained with the tool developed by Navarro-Colorado and manually verified by experts in some cases.

Rantanplan (De la Rosa *et al.* 2020b) is a rule-based tool for Spanish poetry. It relies on POS tags and its own syllabification algorithm to assign stress to words and lines, in a similar way to the ADSO Scansion system (Navarro-Colorado 2018). Like ADSO, it is based on the *Corpus de Siglo de Oro* (Navarro-Colorado 2015). However, Rantanplan applies all possible synaloephas and synaeresis at the syllable level before determining the metrical pattern. If the expected metrical length has a greater value than the calculated one, synaloephas and synaeresis are undone in different tests until the discrepancy is resolved. Rantanplan works with different types of verses, not only hendecasyllables, as is the case with ADSO. Rantaplan has its source code available in a code repository. In addition, its scansion can be analyzed and visualized on the web-based interface application PoetryLab (De la Rosa *et al.* 2020a) (see Figure 3). This platform also displays Named Entity Recognition provided by the tool HisMeTag[15] and enjambement detection provided by the Python library JollyJumper.[16] Rantanplan is intended to complement the data repository, Averell,[17] which contributes to reducing the current limitations of quantitative research on poetry by standardizing poetic corpora into an easily downloadable and accessible format.

A different approach is employed by Jumper (Marco *et al.* 2021), which addresses the metrical annotation of Spanish poetry without performing syllabification but by counting syllables just by counting the vowels of each word. Stress is assigned using the same method: the position of graphically accented vowels is defined by counting vowels, and, in case of no graphical stress, Spanish orthographic rules are applied. The verse level module is composed by three submodules: the first module counts syllables by counting vowels and assigns stress on the line level; the second module applies line compensation (e.g. synaloepha) to lines which are longer than eleven syllables; finally the third module addresses metric ambiguities. The poem level module analyzes the scansion performed by the line level module for each line evaluates whether only one line length is attested in the poem (in the case of a fixed-meter poem), and whether some line lengths are more frequent than others. Jumper's code is available in a code repository.

Figure 4. Metrical annotation with Metricalizer.

Finally, De la Rosa *et al.* (2021) conducted metrical pattern annotation of Spanish, English, and German poetry using transformer-based models. They utilize monolingual and multilingual BERT-base and RoBERTa models (Wolf *et al.* 2019) to predict line stress patterns. The findings demonstrate that these models effectively capture sufficient structural information for poetry scansion. Specifically, they perform well for Spanish in a monolingual setting, and the cross-lingual transfer, when training the model on the three languages, yields to particularly promising results for English and German. The training, test, and evaluation datasets consist of the following: for Spanish, a subset of the *Corpus de Siglo de Oro* (Navarro-Colorado 2015); for English, a subset of the For Better For Verse corpus; for German, the manually annotated corpus from Haider *et al.* (2020). The German corpus comprises 158 poems from 1575 to 1936 which are annotated in terms of syllable stress, foot boundaries, caesuras, and line main accent. The original lines are available on the online platform *Antikoerperchen Lyrik Datenbank*.[18]

4.1.3 Ancient Greek

A tool for Ancient Greek hexameter, developed by Papakitsos (2011) is one of the scansion machines that emerged in the early 2010s. The primary goal of the tool is to divide the line in six parts (feet or meters) and determine their length. The tool has four modules: the first module detects syllables; the second module assigns length based on the number of syllables; the third module identifies foot structure; and the final module returns a Boolean (true or false) value indicating the success of the scansion. When the Boolean is true, the 'identity of the syllables and feet is known' (Papakitsos 2011).

Another tool developed for Ancient Greek is Greek_scansion, created by Conser (2017). This tool analyzes prosody in Ancient Greek poetry and prose. It offers two versions: a generic one not tied to a specific meter, and another specifically designed for iambic trimeter. The tool provides line scansion as a list of syllable markers, namely "long," "short," and "unknown." The syllabification module is based on James Tauber's Greek accentuation Python library[19] and can potentially work with a different syllabifier. Unfortunately, the source code, available in a code repository, does not appear to be updated regularly.

Greek_scansion (Schumann 2020; Schumann *et al.* 2022), homonym of the tool by Conser (Conser 2017), was implemented based on the syllabification method proposed by Papakitsos (Papakitsos 2011). Like the tool by Papakitsos this rule-based tool also focuses on annotating Ancient Greek hexameter. It uses a class ruleset containing linguistic rules for syllabification and syllable count, and a verse class that contains the verse model, including information about the poetic text such as syllabification, syllable count resulting from hyphenation, and the hexameter scheme. The source code for this tool is available in a code repository.

4.1.4 Czech

In the expanding field of poetry scansion, the *Versologie* research group at the Institute of Czech Literature has been particularly prolific, developing a set of tools for poetry analysis. These data are available in the Database of Czech Metres (DCM; Ibrahim and Plecháč 2011), which provide metrical and stanzaic descriptions of poems contained in the Corpus of Czech Verse. Poetry is analyzed in terms of prosody and metrical structure through a set of algorithms. With DCM, poetry can be statistically evaluated in terms of verse meter, verse length, line ending, metrical pattern, stanza and rhyme scheme, and types of fixed metrical forms (e.g. sonnet). In addition to DCM, the *Versologie* research group also developed a tool called *Gunstick* for automatic analysis of rhyme. Its annotations are available in the Database of Czech rhymes.[20] The set of tools to analyze the material available in the Corpus of Czech Verse and the DCM are available on the Institute of Czech Literature's website.

4.1.5 Dutch, Middle Dutch, and Renaissance Dutch

The KBNL scansion generator (Koppelaar and Van Oostendorp 2013; Van Oostendorp 2014) is an annotation tool for Dutch poetry. It uses information about stressed syllables from the CELEX Lexical database for Dutch[21] to determine which syllables are stressed in the words that form a line. Subsequently, the algorithm attempts to match the line with an ideal template (in their example, a fixed number of iambs). An online platform is for the KBNL scansion generator (Koppelaar and Van Oostendorp 2013; Van Oostendorp 2014), offering comprehensive information about the tool, a demo for users to scan poetry, and a link to the code repository.

A Middle Dutch syllabifier (Haverals, Karsdorp, and Kestemont 2019) has been developed using a bi-LSTM RNN. The tool provides syllabification of Middle Dutch texts. Each word is inputted to the model at the character level, and the model provides the syllable grouping. In Haverals (2020) word stress is implemented. The model was trained on the Corpus Van Reenen-Mulder (Reenen and Mulder 1993). This tool can be applied to both verse and prose, and in Haverals (2020), it was used to define the metrical pattern of Dutch medieval poetry. The source code is available in a code repository. In addition, the input data for the machine learning algorithm in various

formats and a pdf-file with further information about the dataset are also publicly accessible on Zenodo.

De Sisto (2020) developed a tool similar to that by Haverals, Karsdorp, and Kestemont (2019), for the metrical annotation of a Dutch Renaissance poetry corpus. Like the Middle Dutch syllabifier and other data-driven tools, the tool for Renaissance Dutch uses a bi-LSTM RNN. The tool is based on two models: one is trained on a list of the most frequent words syllabified and annotated in terms of stress (with a focus on polysyllabic words); the other on random lines from the corpus with stress pattern annotations. These two models are then combined to extract the metrical pattern of a line based on the lexical stress assignment of the words it contains. Line annotation only considers syllabification and disregards word boundary. The Dutch Renaissance poetry corpus annotated by De Sisto (2020) and the source plain texts are available in a code repository.[22] The original source texts used to build the corpus are available at *De Digitale Bibliotheek voor de Nederlandse Letteren* (DBNL).[23]

4.1.6 French

Metrometer (Beaudouin and Yvon 1996) is a tool for French poetry which was developed in the mid-nineties. As mentioned earlier, in the case of the French poetic tradition, metrical annotation is based on the number of syllables per line. Metrometer uses phonemic transcriptions, phonological and syntactic information to calculate the number of syllables and of metrical positions per line while considering the occurrence of metrical phenomena such as synaloepha and liaison. The input is converted into phonemic transcriptions enriched with information such as syntactic components. The tool combines a syntactic analyzer with a system of grapheme-to-phoneme transcription. The rules of the tool were developed based on alexandrines by P. Corneille and J. Racine, hence it is more suitable for annotating this specific form of versification.

Reddy and Knight (2011) propose an unsupervised and language-independent model for rhyme schemes detection. The model infers rhyme schemes based on word repetition within stanzas. Their study tests this model on English and French data. English and French poetry data were used in the rhyme recognition process. The English data are from the open access part of the corpus used by Sonderegger (2011) consisting of 12,000 stanzas from the 15th to 20th centuries. The French data are from the ARTFL project[24] containing 3,000 stanzas.

Lastly, another tool for French poetry is *Anamètre* (Delente and Renault 2015). The tool is used for the analysis of French poetry and drama from the early seventeenth to the early twentieth century. The meter calculation program employs a deterministic approach based on Cornulier's metrical method (Cornulier 1982). Developed in Python, this tool identifies vowels which are part of the metrical pattern and from those calculates the meter based on the prosodic and morpho-syntactic constraints of the French language, which vary diachronically and depend on the context. Additionally, *Anamètre* uses a lexical resource of grammatical syllables and employs algorithms to address issues of ambiguity by considering left and right context.

4.1.7 German and Middle High German

Metricalizer (Bobenhausen 2011; Bobenhausen and Hammerich 2015) is a rule-based tool for metrical analysis of German poetry. It detects words, line, rhyme, and stanza structures in a text and provides word syllabification. The metrical analysis is based on a rule-based algorithm that incorporates prosodic and morphological information. In addition, the tool can identify metrical complexity by detecting discrepancies between prosodic and metrical structures. The results produced by Metricalizer can be saved in TEI-XML format. An online platform is available for accessing Metricalizer. In addition to the user interface, the website offers an analyzed poetry corpus from the *Freiburg Anthologie*.[25] The corpus analysis provides a rich overview of the data that can be generated with the tool. The poems, organized by author, are initially displayed as plain text and can be analyzed and presented with their corresponding annotations (according to metrical pattern, prosodic level, word boundary, and complexity). The database allows for other options, such as visualizing the transcription and comparing poems. An annotated sample from the corpus from the *Freiburg Anthologie* available on the Metricalizer website is shown in Fig. 4.

Haider (2021) conducted experiments to test how different models learn meter on two large corpora of English and German poetry. After preprocessing the corpora, different models were compared. The results demonstrated that bi-LSTM-CRF models with syllable embeddings outperform the baseline CRF model and BERT models. Notably, the experiments yielded interesting findings regarding caesura: the use of POS information supported the correct identification of caesura, which highlights its connection to poetic rhythm and its dependency on syntax. The bi-LSTM-CRF models also perform better in terms of emotion prediction. Specifically, emotion prediction seems to play a role in learning POS and verse measure, leading Haider to propose a relationship between meter and emotion. The two large corpora used for this study contain approximately 3 million lines of English poetry and around 2 million lines of German poetry. The English

corpus contains all poetry available in the English Project Gutenberg.[26] The German corpus contains data from digitized corpora such as the German Text Archive,[27] the Digital Library of Textgrid[28] and the German version of Project Gutenberg. The data are available in both JSON and TEI P5 XML format.

The tool *Automatische Mittelhochdeutsche Metrik* 2.0 (Dimpel 2015) provides metrical annotation for Middle High German. Originally designed for authorship detection to identify an author's metrical profile, the tool annotates and statistically evaluates line features, such as stress placement, by using the ErMaStat package, which follows the *Kadenz-Taxonomie* "rhythm taxonomy" by Heusler (1927). Not all features were statistically significant in improving the performance of the tool. This was the case, for instance, of elision, which did not significantly improve line annotation. Consequently, these features have been excluded from the tool. However, they can be activated manually by the user in the user interface in case they are of interest to the researcher.

Estes and Hench (2016) use machine learning technology for annotating Middle High German poetry. Their supervised learning model, namely a bi-LSTM with CRF neural network, predicts the metrical value of syllables. The meter analyzed is both stress-based and quantity sensitive, therefore both stress- and weight-related features are predicted by the tool. Other features considered are syllable position within the line, word boundary, syllable length in characters, syllable constitution in characters, and elision. To obtain the most likely scansion of a line, additional metrical rules are considered and marginal probabilities calculated. The tool was trained on material from the *Mittelhochdeutsche Begriffsdatenbank*.[29] The database can be consulted online. However, most of its material still has copyright, hence, the whole texts are not publicly available. The Middle High German scansion tool (MHG-scansion model) by Estes and Hench (2016) has a graphical user interface and its source code available in a code repository.

4.1.8 Latin

Also dating from the early 2010s, Pedecerto (Colombi *et al.* 2012) has been developed within the project *Traditio patrum* FIRB of the *Università di Udine*. It is used to annotate approximately 244,000 dactylic lines from the *Musisque Deoque* digital archive. Unfortunately, neither the source code nor the algorithm used for developing the tool seems to be available. The website presenting Pedecerto offers a user-friendly online demo of the tool which can be used to scan verse of various meters. In addition, the platform provides a corpus of annotated poetry[30] and the possibility of doing multiple types of searches, such as searching for word or verse type, based on line metrical pattern or based on specific metrical sequences. The plain text can be instead browsed at the *Musisque Deoque*[31] archive.

Calliope scansio (Jacquet 2013) dates back to the same year. This tool for Latin poetry detects the number of syllables per line and groups them into feet. Syllables are looked up in a dictionary, and syllabification occurs on a character-by-character basis, following specific rules. Once the syllables are defined, their quantity is assigned based on their position in the words. The resulting line annotation is matched to a metrical template, for example hexameter, by considering all possible combinations and groupings of the material, until the best combination is found. The code repository of Calliope scansio contains some annotated and non-annotated material, however, the folder contains a limited amount of material and, just like the whole tool, has not been updated since 2014.[32] The code repository of Calliope scansio mentions a version of the tool with graphical user interface, but it does not appear to be currently available. In addition, its repository does not seem to be regularly updated.

4.1.9 Old Occitan

Slightly different from the other tools introduced in this section, the metrical annotation machine by Rainsford and Scrivner (2014) serves the purpose of adding metrical annotation to treebanks (i.e. syntactically annotated corpora). The process of annotation of Old Occitan follows three steps: first, a syllable-level tokenizer algorithm syllabifies words in a line; second, the results are manually corrected, and lexical stress is added; third, a scansion algorithm, equipped with accepted rules of syllabification, assigns metrical positions to each syllable. The material used by Rainsford and Scrivner is available—although without annotation—on the Old Occitan corpus *Le Roman de Flamenca Corpus*.[33] The online platform appears to be under construction; therefore, while the plain text of the poems can be browsed, the metrical annotations do not seem to be available.

4.1.10 Portuguese

The first tool for metrical annotation of Portuguese verse also dates back to the early 2000s. It was developed by Araújo and Mamede (2002) and analyzes poems in terms of line length, rhyme, and the number of lines in each stanza. Its architecture is inspired by Galaxy-II (Seneff *et al.* 1998) with a central process that coordinates the various modules: the external interface, which generates phonetic transcriptions and syllabic division; the lexical interface, which connects the information generated by the external interface to the lexicon; the system interface, which includes

information such as poem edition; the concept identification module, which assigns rhyme scheme, strophic, and metrical structure; and the rule identification module, which provides the final classification of the poem. In addition, Mamede *et al.* (2004) implemented another module to suggest the ending word of a verse given an incomplete poem as input.

Aoidos (Mittmann, von Wangenheim, and dos Santos 2016) is the second and, to the best of our knowledge, the only other tool which focuses on Portuguese verse. This tool annotates the metrical pattern of the line and detects rhyme schemes. In addition, the tool detects the number of syllables per line and which poetic licenses, for example, synaloepha, are employed in the analyzed text. A trial version of Aoidos is available on its website.[34] By creating an account, users gain access to the entire tool and can analyze a large amount of data.

4.1.11 Russian

An example of a tool for the metrical annotation of Russian syllabo-tonic meter is that developed by Pilshchikov and Starostin (2011). They use a program based on "Treeton" (developed by Starostin for morpho-syntactic analysis, which incorporates elements of StarLing).[35] The tool calculates the proximity to various hypotheses instead of favoring one hypothesis over another, based on theoretical considerations. The tool consists of six modules: 1) tokenizer; 2) syllabifier (which follows the Sonority Sequencing Principle); 3) morphological analyzer (developed by Starostin, generating morphological interpretations at the word level using Andrei Zalizniak's Grammatical Dictionary of the Russian Language, which includes stress placement for every word in the inflection paradigm; 4) fragmentator, identifying the elements composing the poetic text, for example, line, stanzas, etc.; 5) accentuation form generator, defining multiple possible ways of accentuating a line; and 6) meter analyzer, calculating the distance of each verse to each of the five Russian syllabo-tonic meters (iamb, trochee, dactyl, amphibrach and anapaest). StarLing has been an essential tool for the analysis of Russian poetry (see for instance Kozmin 2006).

Some principles proposed by Pilshchikov and Starostin for syllabic–accentual meters (e.g. the formal definition of the degree of metricality, indicated as "1," divided by the number of deviations from the pattern) have also been employed to annotate poetic traditions other than the Russian. For instance, they were the base for the program of verse annotation of The Corpus of Czech verse.[36]

An ongoing project on metrical analysis of Russian poetry has been launched at the Institute of Computational Technologies SBRAS (Barakhnin, Kozhemyakina, and Borzilova 2019). The tool comprises three levels: structural, semantic, and pragmatic. The structural analysis defines the characteristics related to meter and degree of metricality, while the semantic level extracts the semantic structure of the poetic text. Finally, the pragmatic level analyzes genre and style of the text. The tool can be used for analyzing poems on an online platform with a graphical interface, although the platform is currently only available in Russian.

A substantial and publicly available source for Russian poetry is the Poetic subcorpus of the Russian National Corpus,[37] which contains a vast amount of data (12.8 million word forms). This corpus enables searching for texts based on various parameters (meter, line length, rhyme, and stanza). The metrical annotation in this corpus was initially done manually and without providing a metrical pattern for each line, but now such information is available. The corpus contains the meter and the rhythmic pattern of each line (which were obtained with automatic analysis).[38] Recent work as that by Plecháč, Zelenkov, and Šela (Plecháč, Zelenkov, and Šela 2020), has utilized the poetry in this corpus for analysis, employing a machine-assisted accentuation tool developed by Yuri Zelenkov.[39]

4.2 Tools for poetry recognition

Poetry recognition is a recent development in the field of text classification[40] and automatic text recognition with the aid of Optical Character Recognition (OCR). The purpose of poetry recognition tools is to detect poetic text and to automatically distinguish it from prose. All tools reviewed in this section are for English poetry.

The first approach to poetry recognition was introduced by Tizhoosh and Dara (2006). Their main goal was to suggest a way to "measure poetry" and provide an exploratory study to lay the foundations for the research field of poetry recognition. They define a measure of poetry based on the textual features which can distinguish poetry from prose, such as the number of lines in the text, since poems generally have less lines than prose. In addition, they consider the average number of words per line, which is generally smaller for poetry, as well as the number of words that rhyme, which is expected to be greater. Finally, they calculate the poetic significance of words and the density of poetic words in the text. They conducted a preliminary experiment aimed at defining a set of poetic features relating to the structure and organization of poetic text, and three case studies. Each case study focused on a different set of poetic features and machine learning techniques. Bayesian classifiers (Maron 1961; Minsky 1961) are used for the first two case studies, while decision tree classifiers (Quinlan 1986; Duda, Hart, and Stork 2001) are employed in the third. The

first case study considers three types of features, namely rhyme, rhythm (the number of words and the number of syllables in a line and throughout the poem), and frequency of adjectives in comparison to prose. Their results suggest that rhythmical features, followed by rhyme, were the most accurate in distinguishing. The frequency of adjectives, on the other hand, was found to be not significant for poetry recognition. In the second case study, the classifier was trained on multiple classes representing poetic forms and one representing prose; nine features were considered, such as the number of lines and the number of words in a line, and in the whole texts, the number of stanzas, the number of alliteration and rhymes. This method performs considerably well, albeit using a small dataset for training. In the third case study, the Festival Text-to-Speech Synthesis System (Black, Taylor, and Caley 2002) was used to extract various text features, particularly those relevant to semantics and prosody, that were then employed to train a decision tree classifier. As in the previous case studies, the employment of the extracted features gave high-accuracy results.

Subsequently, Tizhoosh, Sahba, and Dara (2008) explored various techniques of poetry recognition. They focused on extracting genre-specific poetic features, for example, rhyme, meter, in order to define new poem classification techniques. Two classifiers were used, namely naïve Bayesian and Multilayer Perceptron. They evaluated five different approaches: (1) a traditional text classification method employing Information gain and Mutual Information (Yang and Pederesen 1997); (2) a method that considered shape, that is, structural characteristics of poetry such as line length; (3) a method which combined shape and rhyme features; (4) a method which considered shape, rhyme, and metrical structure; and (5) and a method which combined poetic features of method 3 and word frequency. Despite all approaches performing well, the highest accuracy was achieved with method 5.

A further advancement for poetry recognition was represented by the tool developed by Lorang and *et al.* (2015). This tool used an image-based approach to identify poetry in historical newspapers. The purpose of the tool was to detect and make available for further investigation poetry which appears in non-poetic historical texts. In order to do so, supervised machine learning was used to create a classifier able to distinguish snippets that contain poetry from those that do not. A preprocessing phase is followed by feature extraction. The preprocessing algorithm performs blurring on the snippet from a newspaper, then, the blurred image undergoes bi-Gaussian binarization and pixel consolidation. The feature extraction algorithm saves a computed set of attributes to the data structure representing the processed image.

Similarly, the project by Kilner and Fitch (2017) addressed poetry detection in digitized historical Australian newspapers from the AusLit database (Kilner 2002). The authors trained a naïve Bayesian classifier adapting the code by Shiffman (Bayesian Filtering,[41] 2008) and that by Paul Graham (A Plan for Spam,[42] Graham 2002), and by using the overProof (Evershed and Fitch 2014) correction algorithm to improve the quality of OCR. The features considered for recognizing poetry were based on Tizhoosh *et al.* (Tizhoosh, Sahba, and Dara 2008).

Finally, the most recent tool for poetry identification was developed by Foley (2019, 2020). The tool is an automatic, learned model for poetry extraction from digitally scanned books. Its purpose is to identify poetry in scanned documents and in particular in scanned pages which do not necessarily only contain poetry. The model (using bi-LTSM RNN) has been trained on a few thousand labeled pages and detects anomalous formatting in the page, such as differences in punctuation, margins, and capitalization. This tool has its code, model, and datasets available and downloadable.

4.3 Tools for poetry visualization[43]

4.3.1 English

In this section, we describe a range of tools capable of visualizing different aspects of a poem. Most of these tools have emerged in recent years, indicating an increasing interest in poetry from the perspective of distant reading (Moretti 2013), macroanalysis (Jockers 2013), and quantitative approaches.

The first research-oriented method was presented by Kaplan and Blei (2007). It visualizes stylistic similarities and relationships between poems by embedding them in a vector space and using Principal Component Analysis. Style detection is used to distinguish poetic texts and cluster poems by poet. A set of features is used to differentiate different styles and classify poems. These features are categorized as orthographic (e.g. word count, average line length), syntactic (e.g. frequencies of POS), and phonemic (e.g. sound patterns, assonance, and consonance). Each feature was assigned a weight. The CMU Pronouncing Dictionary for North American English (Weide 1998) is used for word-to-phoneme translation. The method is applied to a miscellaneous corpus of American poets from various sources and different centuries. The authors conclude that their algorithm is able to distinguish between poems from poets with different styles, verifying whether a single long poem follows the same style, and providing evidence of known mentoring relationship between some poets.

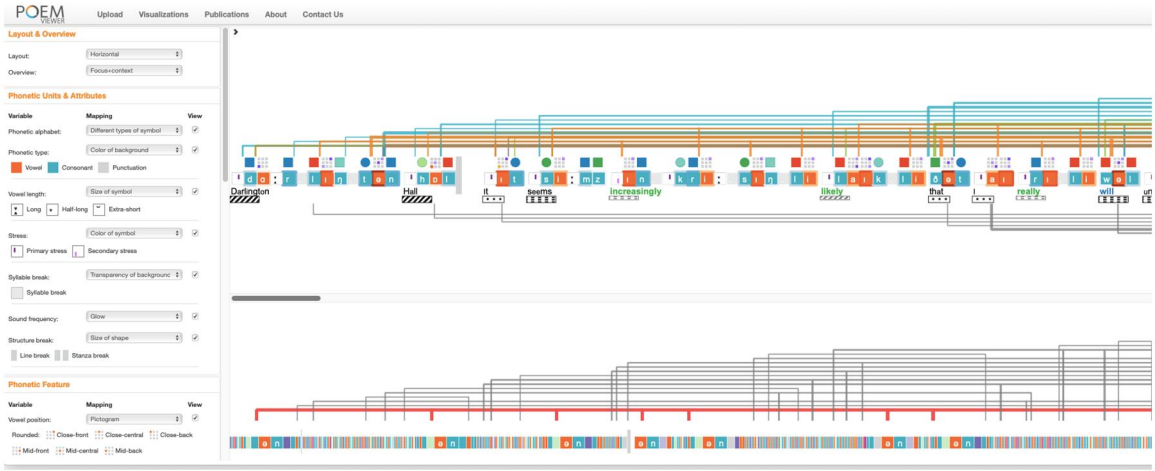

Figure 5. Sample of poetry visualization with PoemViewer.

Myopia (Chaturvedi *et al.* 2012) is a software tool implemented in Python for poetry visualization in the context of close reading. It was developed based on the poems available in the Poetess Archive,[44] which contains 4,000 entries of poetry written by British and American poets between 1750 and 1900. Myopia uses a schema extended from TEI, where a poem's characteristics are structured as tags and attributes according to XML standards. It is designed for comparing literary attributes of different poems and to improve the display of a multidimensional representation of poetry encoded in TEI. The tool has a visual interface and currently exists as a desktop tool.

Poem Viewer (Abdul-Rahman *et al.* 2013) is part of the Visual Informatics Lab at Oxford and is a web-based tool for visualizing poems which aims of enhancing close reading.[45] It was developed to create a user-centered tool that allows the visualization of different features. The features that can be displayed pertain to the phonic aspect of the poem: the tool shows consonants, vowels, their phonetic features, and phonetic relations (rhyme, assonance, and alliteration). It also visualizes the vocalic movement within the vocalic triangle, depicting the transition from one vowel to another while reading the line. In addition, word repetition, word sentiments, and semantic relations can be visualized. Poem Viewer has an online graphical user interface platform where poetry can be analyzed. The platform gives many customizable options regarding the visualization, such as displaying vowel length or word repetition. An example is given in Fig. 5.

The Eighteenth-Century Poetry Archive (ECPA; Huber 2018) provides the community with a range of new web-based poetry visualization tools for this corpus. For each digital edition of a particular poem, users can apply analysis tools to extract information on the phonological, morphological, syntactic, semantic, and pragmatic (mainly, named entities) layers. Additionally, it integrates some original and adapted visualization tools, including the aforementioned Poem Viewer, as well as Phonemia—which visualizes the phonemic makeup of a poem—and DoubleTreesJS—which aims to provide a more compact view of words in their context of use than traditional concordances (Key Word in Context). These tools are available on the ECPA's graphical user interface platform and are connected to the digital edition site of each poem. The analysis tools are displayed by default, while Phonemia, Poem Viewer, and DoubleTreeJS must be launched by the user.

4.3.2 Spanish

One tool for visualization of Spanish poetry is the project DISCOVER (Bermúdez-Sabel, Ruiz Fabo, and Martínez Cantón 2023). This tool adapts existing resources for the analysis of rhymes in the DISCO corpus, featuring Spanish sonnets (Ruiz Fabo *et al.* 2017a). This corpus was annotated using ADSO Scansion system (Navarro-Colorado 2018), ANJA by Pablo Ruiz Fabo (Ruiz Fabo *et al.* 2017b), and RhymeTagger[46] by Petr Plecháč (2018) for rhyme detection. The interface, implemented in PHP and Javascript, allows users to filter results by corpus and prosodic elements and view rhyme trends. DISCOVER has an online graphical user platform and allows the user to limit the corpus by origin, century, and even gender of the author. It also provides filters for stress, rhyme, the presence of enjambment, and even visualize rhyme trends. However, the code is not available.

4.3.3 Italian

Dante Visualized (Ferraro 2020) was developed to enhance distant reading of Dante's *Divina Commedia*,

and has the potential to be applied to other texts in future research. The tool analyzes both stylistic and semantic features, offering a schematic representation of the structure and rhythm, a visual representation of the sentiment analysis, and the distribution of keywords in the three parts of the work. The tool was developed by using machine learning techniques and the Naive Bayes Classifier (Perkins 2010). The tool was implemented in Python. A JavaScript library (Bostock D3.js[47]) was used for producing the visualizations. The code repository of the Dante Visualized (Ferraro 2020) includes the tool itself as well as the results of the case study on the *Divina Commedia* by Dante.

5. The role of rules and data

Poetry analysis tools can be classified based on the type of technology they are built upon and, more specifically, the role played by rules and data in their structure.

A large number of tools are rule-based, especially, those for metrical annotation. These tools are constructed based on a set of predefined rules that govern the annotation process. For metrical annotation, explicit annotation rules are outlined, and the tool follows these rules to annotate new material.

With the advancements of machine and deep learning techniques, the use of these technologies has spread in the field of automatic poetry analysis, leading to different a type of systems. In these systems, rules are not explicitly provided to the tool; instead, the model learns implicit generalizations from large amounts of data. Machine learning algorithms are employed to extract information and create a model, which is then used to analyze new data (Patternson and Gibson 2017). This process can be either supervised or unsupervised. In supervised learning, the model is trained on manually annotated data to make generalizations for annotating new data, while unsupervised learning involves using raw data and fine-tuning the model with manually annotated material. This type of tool, where the model makes generalizations from data, is called data-driven or inference-based. Although the terms may seem contradictory, they refer to the same aspect: data-driven refers to the fact that the model receives data and learns from them; inference-based refers to the process of making generalizations and making predictions from the data.

In the present article, both rule-based and inference-based tools are considered. As mentioned above, within the group of metrical annotation tools, most tools are rule-based. Defining metrical and prosodic rules for verse annotation has proven to be an effective technique. These tools generally rely on rules related to linguistic features, such as syllabification, stress assignment, phonetic representation, as well as metrical features of the line, such as line length, number of stressed syllables per line, rhyme, and the number of lines in a stanza.

Nevertheless, in the last few years, the use of machine learning technologies in the field of automatic metrical annotation has significantly increased. The most commonly employed approach involves RNNs (Schmidhuber 1989) with bi-LSTM cells (Hochreiter and Schmidhuber 1997) and CRF (Lafferty, Abdul-Rahman, and Pereira 2001). These techniques have demonstrated high accuracy in metrical annotation. Some of the metrical annotation tools developed using machine learning techniques include the MHG-scansion model (Estes and Hench 2016), the tool by Tanasescu, Paget, and Inkpen (2016), the tool by Agirrezabal (Agirrezabal 2017; Agirrezabal, Alegria, and Hulden 2017), the tool by KAIST (Kim, Zhao, and Zelalem 2018), the Middle Dutch Syllabifier (Haverals, Karsdorp, and Kestemont 2019) and Stresser (Haverals 2020), the tool by De Sisto (2020) and that by Haider (2021). The manually annotated material used as training data of these models is generally annotated according to syllable structure, stress, rhyme, word boundary, and, in the case of quantitative verse (i.e. Middle High German for MHG-scansion model), in terms of syllable quantity. These tools typically employ a similar architecture based on RNN with bi-LSTM. A different method is proposed in the study by De la Rosa *et al.* (2021), where transformer-based models (Wolf *et al.* 2019) are employed for detecting line stress patterns.

Similar to data-driven metrical annotation tools and, actually, a predecessor of them, the tool by Hayward (1991, 1996) also uses Artificial Neural Networks for evaluating line metricality. In addition, Greene, Bodrumlu, and Knight (2010) use unsupervised learning for extracting word stress from a corpus as part of their tool which annotates, translates, and generates poetry.

Machine learning techniques are also utilized in the field of poetry visualization, as demonstrated by the ongoing project Dante Visualized (Ferraro 2020). This tool uses a Naive Bayes Classifier (Perkins 2010) trained on a manually created dataset of subsets of texts from other authors that are linguistically and temporally similar (Ferraro 2020).

Finally, inference-based methods are highly common among tools of poetry recognition; in fact, all tools considered in the present survey employ them: Tizhoosh and Dara (2006), Tizhoosh, Sahba, and Dara (2008), Lorang *et al.* (2015), and Foley (2019, 2020).

6. Conclusion

In this article, we have presented a survey of automatic tools for poetry analysis, with a focus on those

analyzing structural characteristics of verse. To this end, we provided various classifications of the tools based on language, general purpose, and the role of rule-based or data-driven technology. The purpose of this article was to carry out a survey of the available tools for automatic analysis of poetry and provide information about their applicability for future research. Although many traditional poetry scholars might not be familiar with these automatic tools, our work highlights the significant number of automatic tools developed to date for poetry analysis. This overview aims to bridge the gap between more traditional poetry studies and available computational versification approaches, enabling researchers to scale up their data for testing theoretical questions and supporting them in terms of generalizability and verifiability.

We hope poetry scholars will find the resources presented in this article useful, particularly for research focusing on European poetic traditions. In addition, we believe that the availability of many of the tools as online platforms or open-source code will encourage researchers to use them—or their open annotated corpora—in their studies.

Both Natural Language Processing and automatic analysis of poetry are fields in expansion and, as a result, we are witnessing new and important advances, methods, and applications every day. Being aware of the current developments that could enhance our research is crucial for preserving the quality of our studies. Automatic tools present both challenges and opportunities for poetry researchers. They pose challenges due to their interdisciplinary nature, which requires researchers to acquire knowledge from various disciplines. However, they also offer opportunities to address unanswered questions and explore new avenues of research in the field of poetry, which, as Emily Dickinson[48] stated, is full of possibilities.

Author contributions

Mirella De Sisto (Conceptualization, Formal analysis, Investigation, Methodology, Writing—original draft, and Writing—review & editing), Laura Hernández-Lorenzo (Conceptualization, Formal analysis, Investigation, Methodology, and Writing—original draft), Javier de la Rosa (Data curation, Validation, and Writing—original draft), Salvador Ros (Conceptualization and Project administration), and Elena González-Blanco (Conceptualization, Funding acquisition, and Project administration)

Funding

Research for this paper was made possible thanks to the Starting Grant researchproject Poetry Standardization and Linked Open Data: POSTDATA (ERC-2015-STG-679528) http://postdata.linhd.uned.es/ led by Elena González-Blanco. Thisproject is funded by the European Research Council (https://erc.europa.eu) (ERC)as part of the European Union's Horizon2020 research and innovation program.

Conflict of interest statement

None declared.

Notes

1. See the Book of Abstracts of this Conference: https://dh2020.adho.org/abstracts/, accessed 22 Feb. 2022.
2. Links to tools' online platforms and source codes are provided in Appendix Table A1.
3. See at: https://rpo.library.utoronto.ca, accessed 22 Feb. 2022.
4. See its website at: http://sonnets.org, accessed 22 Feb. 2022.
5. eSpeak: http://espeak.sourceforge.net, accessed 7 Oct. 2021 .
6. Available at: https://github.com/quadrismegistus/prosodic/tree/master/corpora, accessed 22 Feb. 2022.
7. See the website platform at: https://prosody.lib.virginia.edu, accessed 22 Feb. 2022.
8. Available at: https://github.com/waynegraham/for_better_for_verse/tree/master/poems, accessed 22 Feb. 2022.
9. Available at: https://github.com/quadrismegistus/poesy/tree/master/poems/shakespeare_sonnets, accessed 18 Feb. 2022.
10. See the project's website at: https://sparsar.wordpress.com/, accessed 18 Feb. 2022.
11. See at: http://poetryfoundation.org, accessed 22 Feb. 2022.
12. Available at https://github.com/ixa-ehu/ixa-pipe-tok, accessed 19 Feb. 2022.
13. https://nlp.lsi.upc.edu/freeling/index.php/node/1, accessed 21 June 2023.
14. Available at: https://github.com/bncolorado/CorpusSonetosSigloDeOro, accessed 18 Oct. 2021.
15. Available at: https://github.com/linhd-postdata/HisMetag, accessed 30 Oct. 2021.
16. Available at: https://github.com/linhd-postdata/jollyjumper, accessed 30 Oct. 2021.
17. Available at: https://github.com/linhd-postdata/averell, accessed 30 Oct. 2021.
18. Available at: https://lyrik.antikoerperchen.de/, accessed 15 Feb. 2022.
19. Available at: https://github.com/jtauber/greek-accentuation, accessed 15 Feb. 2022.
20. The Versologie research group has also been one of first groups to use metrical information in Digital Literary Stylistics. See, for example, their studies on the interest of metrical information for authorship attribution (Plecháč *et al.* 2019).
21. See at: https://catalog.ldc.upenn.edu/LDC96L14, accessed 14 Jan. 2022.
22. Available at: https://github.com/mirsdev/Dutch_Renaissance_poetry_corpus, accessed 9 Sept. 2021.
23. See the website platform at: https://www.dbnl.org, accessed 9 Sept. 2021.
24. See at: ARTFL-FRANTEXT | The ARTFL Project (uchicago.edu), accessed 12 Nov. 2021.
25. See the website platform at: https://metricalizer.de/en/corpus/, accessed 18 Nov. 2021.
26. See the website platform at: http://www.gutenberg.net, accessed 18 Oct. 2021.
27. Available at: https://www.deutschestextarchiv.de/, accessed 15 Feb. 2022.
28. Available at: http://textgrid.de [accessed 18 Oct. 2021].
29. See at: http://mhdbdb.sbg.ac.at [accessed 18 Oct. 2021].
30. See at: http://www.pedecerto.eu/public/pagine/autori [accessed 31 Oct. 2023].
31. See at: http://mizar.unive.it/mqdq/public/[accessed 18 Feb. 2022].

32. Available at: https://github.com/ChristopheJacquet/Calliope/tree/master/test [accessed 18 Jan. 2022].
33. See at: https://cl.indiana.edu/~obscrivn/F1_495.html [accessed 21 Jun. 2023].
34. https://aoidos.ufsc.br [accessed 21 Jun. 2023].
35. StarLing is a software package for various types of linguistic text and database analyses. For more details: https://starlingdb.org/program.php?lan=en [accessed 21 Jun. 2023].
36. We would like to thank an anonymous reviewer for pointing this out to us.
37. Available at: https://ruscorpora.ru/new/search-poetic.html [accessed 6 Jan. 2022].
38. We would like to thank an anonymous reviewer for providing us information about this corpus and about research based on it. Given that the website and other resources are only available in Russian, it would not have been possible for us to include it in our review without the reviewer's contribution.
39. See Plecháč, Zelenkov, and Šela (2020), p. 136, footnote 7.
40. For relevant literature, see: Manevitz and Yousef 2007; McCallum and Nigam 1998; Ruiz and Srinivasan, 1998; Yang 1999.
41. Unfortunately, the link to this blog post does not exist anymore: http://shiffman.net/teaching/a2z_2008/bayesian/[last accessed 27 Oct. 2015]
42. http://www.paulgraham.com/spam.html [accessed 22 Feb. 2022].
43. Apart from the tools mentioned in this section, there are some studies of rhyme through network analysis. In this sense, we would like to highlight the panel "Understanding Rhyme Through Network Analysis" presented at Digital Humanities 2020 Conference (Houston *et al.* 2020).
44. See its website at: http://www.poetessarchive.org [accessed 18 Oct. 2021].
45. According to its authors, their goal was to produce "a suitable tool to help poets to explore the information spaces of poems. Following the nested design process" (Abdul-Rahman *et al.* 2013).
46. https://github.com/versotym/rhymetagger
47. https://d3js.org/
48. See the poem 'I dwell in Possibility' (https://www.poetryfoundation.org/poems/52197/i-dwell-in-possibility-466).

References

Abdul-Rahman, A. *et al.* (2013) 'Rule-based Visual Mappings—with a Case Study on Poetry Visualization', *Computer Graphics Forum*, **32**: 381–90. https://doi.org/10.1111/cgf.12125

Agerri, R., Bermudez, J., and Rigau, G. (2014) 'Ixa Pipeline: Efficient and Ready to Use Multilingual nlp Tools', *LREC*, **2014**: 3823–8.

Agirrezabal, M. (2017) *Automatic Scansion of Poetry*. Donostia, Spain: University of Basque Country.

Agirrezabal, M., Alegria, I., and Hulden, M. (2017) *A Comparison of Feature-Based and Neural Scansion of Poetry*. in: *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP* 2017, pp. 18–23. Varna, Bulgaria. INCOMA Ltd.

Agirrezabal, M. *et al.* (2013) 'ZeuScansion: A Tool for Scansion of English Poetry', *Journal of Language Modelling*, **4**: 3. https://doi.org/10.15398/jlm.v4i1.102

Algee-Hewitt, M.R. *et al.* (2014) *The Stanford Literary Lab Transhistorical Poetry Project Phase ii: Metrical Form*. in: *9th Annual International Conference of the Alliance of Digital Humanities Organizations, {DH} 2014, Lausanne, Switzerland, 8-12 July 2014, Conference Abstracts*. Alliance of Digital Humanities Organizations (ADHO). http://dharchive.org/paper/DH2014/Paper-788.xml.

Araújo, J. P., and Mamede, N.J. (2002) *Classificador de Poemas*. Lisbon: CCTE.

Barakhnin, V.B., Kozhemyakina, O. Y., and Borzilova, Y. S. (2019) 'The Development of the Information System of the Representation of the Complex Analysis Results for the Poetic Texts', *Vestnik NSU: Information. Technologies*, **17**: 5–17. https://doi.org/10.25205/1818-7900-2019-17-1-5-17

Beaudouin, V., and Yvon, F. (1996) 'The Metrometer: A Tool for Analysing French Verse', *Literary and Linguistics Computing*, **11**: 23–31. https://doi.org/10.1093/llc/11.1.23

Bermúdez Sabel, H., Ruiz Fabo, P., and Martínez Cantón, C. (2023) 'DISCOvering Spanish Sonnets: A Circular Reading Experience', in: A. S. Bories, P. Plecháč, and P. Ruiz Fabo (eds.) *Computational Stylistics in Poetry, Prose and Drama*, pp. 67–86. Berlin, Germany: De Gruyter. https://doi.org/10.1515/9783110781502-004

Black, A.W., Taylor, P., and Caley, R. (2002) *The Festival Speech Synthesis System: System Documentation*. Edinburgh, Scotland: University of Edinburgh, The Centre for Speech Technology Research.

Bobenhausen, K. (2011) 'The Metricalizer2–automated Metrical Markup of German Poetry', *Current Trends in Metrical Analysis*. Bern, Switzerland: Peter Lang.

Bobenhausen, K., and Hammerich, B. (2015) 'Métrique Littéraire, Métrique Linguistique et Métrique Algorithmique de l'allemand Mises en Jeu Dans le Programme Metricalizer2', *Langages*, **3**(199): 67–88.

Breiman, L. (1996) 'Bagging Predictors', *Machine Learning*, **24**: 123–40.

Chaturvedi, M. *et al.* (2012) *Myopia: A Visualization Tool in Support of Close Reading*. in: *Digital Humanities Conference*. https://www-archiv.fdm.uni-hamburg.de/dh2012/conference/programme/abstracts/myopia-a-visualization-tool-in-support-of-close-reading.1.html.

Chaumartin, F. R. (2013) *Antelope, Une Plate-forme de TAL Permettant d'Extraire les Sens du Texte: Théorie et Applications de l'interface Syntaxe-sémantique 228*. Université Paris-Diderot - Paris VII. PhD Dissertation. https://theses.hal.science/tel-00803531/file/TheseFRC.pdf.

Colombi, E. *et al.* (2012) 'Pedecerto [WWW Document]', http://www.pedecerto.eu/public/index. Date accessed 21 June 2023.

Conser, A. (2017) 'Greek_scansion [WWW Document]', https://github.com/aconser/greek_scansion. Date accessed 24 September 2020.

Cornulier, B. (1982) *Théorie du Vers: Rimbaud, Verlaine, Mallarmé*. Paris: Seuil.

De la Rosa, J. *et al.* (2021) 'Transformers Analyzing Poetry: Multilingual Metrical Pattern Prediction with Transfomer-based Language Models', *Neural Computing and Applications*, **35**: 1–6.https://doi.org/10.1007/s00521-021-06692-2

De la Rosa, J. *et al.* (2020a) 'PoetryLab as infrastructure for the analysis of Spanish poetry' in: Navarretta, C. and Eskevich, M. (Eds.), *Selected Papers from the CLARIN Annual Conference 2020. Presented at the CLARIN Annual Conference 2020*. Linköping Electronic Conference Proceedings 180 pp. 82–7. https://doi.org/10.3384/ecp1809.

De la Rosa, J. *et al.* (2020b) 'Fast and Accurate Syllabification and Scansion of Spanish Poetry', *Review Process Length Natural*, **65**: 83–90.

De Sisto, M. (2020) *The Interaction between Phonology and Metre. Approaches to Romance and West-Germanic Metre*. Amsterdam, the Netherlands: Radboud University Nijmegen. LOT Publications.

Delente, E., and Renault, R. (2015) Traitement automatique des formes métriques des textes versifiés, in: *Actes de la 22e conférence sur le Traitement Automatique des Langues Naturelles*. Articles courts, pp. 116–122, Caen, France: ATALA.

Delmonte, R. (2014) A Computational Approach to Poetic Structure, *Rhythm and Rhyme 7*. in: *Archivio istituzionale della ricerca - Università degli Studi di Venezia Ca' Foscari*. https://core.ac.uk/download/pdf/223168461.pdf.

Dimpel, F.M. (2015) 'Automatische Mittelhochdeutsche Metrik 2.0 [WWW Document]', http://web.fu-berlin.de/phin/phin73/p73i.htm. Date accessed 10 August 2020.

Domínguez Caparrós, J. (2020) *Métrica Española*. Madrid: UNED.

Duda, R.O., Hart, P.E., and Stork, D.G. (2001) *Pattern Classification*. New York, NY: Wiley.

Estes, A., and Hench, C. (2016) 'Supervised machine learning for hybrid meter', *Proceedings of the Fifth Workshop on Computational Linguistics for Literature*, pp. 1–8. San Diego, CA: Association for Computational Linguistics. https://doi.org/10.18653/v1/W16-0201

Evershed, J., and Fitch, K. (2014) 'Correcting Noisy OCR: Context beats Confusion', in: *DATeCH'14: Proceedings of the First International Conference on Digital Access to Textual Cultural Heritage*, New York: Association for Computing Machinery, pp. 45–51. https://doi.org/10.1145/2595188.2595200.

Ferraro, G. (2020) From data to visualisation: Dante's Divine Comedy as a case study. *Digital Humanities 2020*. Ottawa, Canada. http://dx.doi.org/10.17613/0g8f-x945.

Fisher, F., and Frey N. (2013) 'Close Reading as Part of a Comprehensive Literacy Framework', *Colorado Reading Journal - Winter 2014*, 30–4.

Foley, J. (2019) *Poetry: Identification, Entity Recognition, and Retrieval*. Boston, MA: University of Massachusetts.

Foley, J. (2020) Automatic Extracion of Poetry from Digitally Scanned Books. *Digital Humanities 2020*. Ottawa, Canada. https://hcommons.org/deposits/item/hc:31737/.

Gervás, P. (2000) 'A Logic Programming Application for the Analysis of Spanish Verse', in: J. Lloyd, V. Dahl, U. Furbach, M. Kerber, K. K., Lau, C. Palamidessi, L.M. Pereira, Y. Sagiv, and P.J. Stuckey (eds.) *Computational Logic–CL 2000, Lecture Notes in Computer Science*, pp. 1330–44. Berlin, Heidelberg, Germamny: Springer Berlin Heidelberg. Https://doi.org/10.1007/3-540-44957-4_89

Graham, P. (2002) 'A Plan for Spam', Paul Graham. [WWW Document], http://www.paulgraham.com/spam.html. Date accessed 10 September 2020.

Greene, E., Bodrumlu, T., and Knight, K. (2010) 'Automatic analysis of rhythmic poetry with applications to generation and translation', *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pp. 524–33.Cambridge, MA: Association for Computational Linguistics.

Groves, P.L. (1998) 'Strange Music: The Metre of the English Heroic Line', *English Literary Studies*, **74**.

Haider, T. (2021) *Metrical Tagging in the Wild: Building and Annotating Poetry Corpora with Rhythmic Features*. in: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, pp. 3715–3725, *Online:* Association for Computational Linguistics.

Haider, T. *et al.* (2020) 'PO-EMO: Conceptualization, Annotation, and Modeling of Aesthetic Emotions in German and English Poetry', in: *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pp. 1652–1663, Marseille, France. European Language Resources Association. https://doi.org/10.48550/ARXIV.2003.07723

Hanson, K., and Kiparsky, P. (1996) 'A Parametric Theory of Poetic Meter', *Language*, **72**: 287–335.

Hartman, C. (1996) 'Scandroid [WWW Document]', http://oak.conncoll.edu/cohar/Programs.htm. Date accessed 20 June 2023.

Haverals, W. (2020) *De Maat van het Middelnederlands: EEN Digitaal Onderzoek Naar de Prosodische en Ritmische Kenmerken van Middelnederlandse Berijmde Literatuur*. Antwerp, Belgium: Universiteit Antwerpen, Faculteit Letteren & Wijsbegeerte.

Haverals, W., Karsdorp, F., and Kestemont, M. (2019) 'Data-Driven Syllabification for Middle Dutch', *Digital Medievalist*, **12**: 2. https://doi.org/10.16995/dm.83

Hayes, B., Wilson, C., and Shisko, A. (2012) 'Maxent Grammars for the Metrics of Shakespeare and Milton', *Language*, **88**: 691–731. https://doi.org/10.1353/lan.2012.0089

Hayward, M. (1996) 'Analysis of a Corpus of Poetry by a Connectionist Model of Poetic Meter', *Poetics*, **24**: 1–11.

Hayward, M. (1991) 'A Connectionist Model of Poetic Meter', *Poetics*, **4**: 303–17.

Henríquez Ureña, P. (1919) 'El Endecasílabo Castellano', *Revista de Filologia Espanola*, **6**(2), 132–57.

Hernández-Lorenzo, L. *et al.* (2021) 'The automatic quantitative metrical analysis of Spanish poetry with rantanplan: A preliminary approach', *Tackling the Toolkit. Plotting Poetry through Computational Literary Studies*. Prague: Institute of Czech Literature of the Czech Academy of Sciences.

Heuser, R. *et al.* (2010) 'Prosodic [WWW Document]', http://prosodic.stanford.edu. Date accessed 8 March 2020.

Heusler, A. (1927) *Deutsche Versgeschichte. Mit Einschluß des Altenglischen und Altnordischen Stabreimverses, Bd. 2/III: Der Altdeutsche Vers (= Grundriß der Germanischen Philologie 8/2)*. Berlin, Leipzig: De Gruyter.

Hochreiter, S., and Schmidhuber, J. (1997) 'Long Short-Term Memory', *Neural Computation 9(8)*, 1735–80.

Houston, N.M. *et al.* (2020) 'Understanding Rhyme Through Network Analysis', *Digital Humanities 2020*. Ottawa, Canada. https://hal.science/hal-02661449.

Huber, A. (2018) *Eighteenth-Century Poetry Archive*. Oxford: University of Oxford (Eighteenth-Century Poetry Archive.

Ibrahim, R., and Plecháč, P. (2011) 'Towards the automatic analysis of Czech verse', *Formal Methods in Poetics*, pp. 295–305. Lüdenscheid, Germany: RAM.

Jacquet, C. (2013) 'Calliope Scansio [WWW Document]', https://github.com/ChristopheJacquet/Calliope Date accessed 10 August 2020.

Jockers, M.L. (2013) *Theme, in: Macroanalysis: Digital Methods and Literary History*, pp. 118–53. Urbana: University of Illinois Press.

Kaplan, D.M., and Blei, D.M. (2007) 'A computational approach to style in American poetry', *Seventh IEEE International Conference on Data Mining (ICDM 2007). Presented at the Seventh IEEE International Conference on Data Mining (ICDM 2007)*, pp. 553–8. Omaha, NE: IEEE. https://doi.org/10.1109/ICDM.2007.76

Kilner, K. (2002) 'AustLit: The Australian Literature Resource [WWW Document]', www.austlit.edu.au. Date accessed 9 January 2020.

Kilner, K., and Fitch, K. (2017) 'Searching for My Lady's Bonnet: Discovering Poetry in the National Library of Australia's Newspapers Database,' *Digital Scholarship in the Humanities*, **32**: i69–83. https://doi.org/10.1093/llc/fqw062

Kim, H., Zhao, Y., and Zelalem, B. (2018) *Automatic Classiffication of Poetry by Neural Scansion*. School of Electrical Engineering and Computer Science, Faculty of Engineering, University of Ottawa. PhD Dissertation. https://ruor.uottawa.ca/bitstream/10393/37309/1/Kesarwani_Vaibhav_2018_thesis.pdf.

Koehn, P. *et al.* (2007) 'Moses: Open source toolkit for statistical machine translation', *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions. Association for Computational Linguistics*, Prague, Czech Republic. Association for Computational Linguistics., pp. 177–80.

Koppelaar, H., and Van Oostendorp, M. (2013) 'The Scansion Generator [WWW Document]', KB Lab, http://lab.kb.nl/tool/scansion-generator. Date accessed 31 October 2023.

Kozmin, A.V. (2006) *Automated Analysis of Verse with Starling Software Package. Automatic Verse Analysis in Starling*. in: *Computsyional Linguistics and Intellectural Technologies. International Conference "Dialogue 2006" Proceedings*, pp. 265–268.

Lafferty, J., Abdul-Rahman, A., and Pereira, F.C. (2001) *Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data*. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001)*, Williams College, Williamstown, MA, USA, June 28 - July 1, 2001. 282–289.

Logan, H.M. (1988) 'Computer Analysis of Sound and Meter in Poetry', *College Literature*, **15**: 19–24.

Lorang, M.L. *et al.* (2015) 'Developing an Image-Based Classifier for Detecting Poetic Content in Historic Newspaper Collections', *D-Lib Magazine*, **21**, 1–24.

Mamede, N. *et al.* (2004) 'An electronic assistant for poetry writing', *Proceedings of the 9th Ibero-American Conf. on Artificial Intelligence. Presented at the LNCS*. Berlin, Heidelberg: Springer. https://doi.org/10.1007/978-3-540-30498-2_29.

Manevitz, L., and Yousef, M. (2007) 'One-class Document Classification via Neural Networks', *Neurocomputing*, **70**: 1466–81.

Marco, G. *et al.* (2021) 'Automated Metric Analysis of Spanish Poetry: Two Complementary Approaches', *IEEE Access*, **9**: 51734–46. https://doi.org/10.1109/ACCESS.2021.3069635

Maron, M. E. (1961) 'Automatic Indexing; an Experimental Inquiry', *Journal of the ACM*, **8**: 404–17.

McAleese, W. (2007) *Improving Scansion with Syntax: An Investigation into the Effectiveness of a Syntactic Analysis of Poetry by Computer Using Phonological Scansion Theory*. Technical Report, Department of Computing, The Open University. https://doi.org/10.21954/ou.ro.00016055.

McCallum, A., and Nigam, K. (1998) *A comparison of Event Models for Naive Bayes Text Classification. in:* AAAI Conference on Artificial Intelligence. https://courses.washington.edu/ling572/papers/mccallum1998_AAAI.pdf.

McClelland, J.L., and Rumelhart, D.E. (1988) *Explorations in Parallel Distributed Processing: A Handbook of Models, Programs, and Exercises*. Cambridge, MA: The MIT Press.

Minsky, M. (1961) 'Steps towards Artificial Intelligence', *Proceedings of the IRE*, **49**: 8–30.

Mittmann, A., von Wangenheim, A., and dos Santos, A.L. (2016) 'Aoidos: A system for the automatic scansion of poetry written in Portuguese', *Computational Linguistics and Intelligent Text Processing. Presented at the 17th International Conference*. Konya, Turkey: CICLing.

Moretti, F. (2013) *Distant Reading*. London: Verso.

Navarro-Colorado, B. (2015) 'A computational linguistic approach to Spanish Golden Age Sonnets: metrical and semantic aspects', *Proceedings of the Fourth Workshop on Computational Linguistics for Literature*, pp. 105–13. Denver, CO: Association for Computational Linguistics. https://doi.org/10.3115/v1/W15-0712

Navarro-Colorado, B. (2018) 'A Metrical Scansion System for Fixed-metre Spanish Poetry', *Digital Scholarship in the Humanities*, **33, 112–127**. https://doi.org/10.1093/llc/fqx009

Padró L. l., and Stanilovsky, E. (2012) *FreeLing 3.0: Towards Wider Multilinguality Proceedings of the Language Resources and Evaluation Conference (LREC 2012) ELRA*. Istanbul, Turkey. European Language Resources Association (ELRA). http://www.lrec-conf.org/proceedings/lrec2012/pdf/430_Paper.pdf.

Papakitsos, E.C. (2011) 'Computerized Scansion of Ancient Greek Hexameter', *Literary and Linguistics Computing*, **26**: 57–69. https://doi.org/10.1093/llc/fqq015

Patternson, J., and Gibson, A. (2017) *Deep Learning*. Sebastopol, CA: O'Reilly Media.

Perkins, J. (2010) 'Text Classifier for Sentiment Analysis—Naive Bayes Classifier', SteamHacker, https://streamhacker.com/2010/05/10/text-classification-sentiment-analysis-naive-bayes-classifier/. Date accessed 10 September 2020.

Pilshchikov, I., and Starostin, A. (2011) 'Automated Analysis of Poetic Texts and the Problem of Verse Meter', in *Current Trends in Metrical Analysis*. Lausanne, Switzerland: Peter Lang.

Plamondon, M.R. (2006) 'Virtual Verse Analysis: Analysing Patterns in Poetry', *Literary and Linguistics Computing*, **21**: 127–41. https://doi.org/10.1093/llc/fql011

Plecháč, P. (2018) 'A Collocation-Driven Method of Discovering Rhymes (in Czech, English, and French Poetry)', *Taming the Corpus: From Inflection and Lexis to Interpretation*, pp. 79–95. Cham, Switzerland: Springer.

Plecháč, P. *et al.* eds. (2019) *Quantitative Approaches to Versification*. Prague: Institute of Czech Literature of the Czech Academy of Sciences.

Plecháč, P., Zelenkov, J., and Šela, A. (2020) 'The Batenkov Phenomenon and the Problem of Authorship Verification: A

Multivariate Statistical Approach to an Unsolved Question', *Acta Slavica Estonica*, **12**: 131–65.

Prince, A., and Smolensky, P. (1993) *Optimality Theory: Constraint Interaction in Generative Grammar*. Manuscript, Rutgers University and University of Colorado, Boulder.

Quinlan, J.R. (1986) 'Induction of Decision Trees', *Machine Learning*, **1**: 81–106. https://doi.org/10.1007/BF00116251

Rainsford, T.M., and Scrivner, O. (2014) 'Metrical annotation for a verse treebank', *Proceedings of the Thirtheenth International Workshop on Treebanks and Linguistic Theories* (TLT13), Tübingen, Germany. University of Tübingen. https://doi.org/10.5281/zenodo.10054982

Reddy, S., and Knight, K. (2011) 'Unsupervised discovery of rhyme schemes', *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Portland, Oregon, USA. Association for Computational Linguistics, 77–82.

van Reenen, P., and Mulder, M. (1993) 'Een Gegevensbank van 14de-eeuwse Middelnederlandse Dialecten op Computer', *Lexikos*, **3**: 259–81. https://doi.org/10.5788/3-1-1110

Ruiz Fabo, P. (2020) 'The Diachronic Spanish Sonnet Corpus: TEI and Linked Open Data Encoding, Data Distribution, and Metrical Findings', *Digital Scholarship in the Humanities*, **36**: i68–80. https://doi.org/10.1093/llc/fqaa035.

Ruiz Fabo, P. *et al.* (2017a) *Diachronic Spanish Sonnet Corpus (DISCO)*. Madrid: UNED.

Ruiz Fabo, P. *et al.* (2017b) 'Enjambment detection in a large diachronic corpus of Spanish sonnets', *Proceedings of the Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage*. in: *Social Sciences, Humanities and Literature*, Vancouver, Canada. Association for Computational Linguistics.. 27–32.

Ruiz, M.E., and Srinivasan, P. (1998) 'Automatic text categorization using neural networks', *Proceedings of the 8th Workshop on Advances in Classification Research*, University of Washington, 59–72.

Schmidhuber, J. (1989) 'The Neural Bucket Brigade: A Local Learning Algorithm for Dynamic Feedforward and Recurrent Networks', *Connection Science*, **403**: 412.

Schumann, A. K. (2020) *Automatische Versmaßannotation in Werken der Homerischen Hexameter-Dichtung*. Berlin, Germany: Freie Universität Berlin.

Schumann, A. K. *et al.* (2022) 'Using Finite-state Machines to Automatically Scan Ancient Greek Hexameter', *Digital Scholarship in the Humanities*, **37**: 242–53. https://doi.org/10.1093/llc/fqab071

Sejnowski, T.J., and Rosenberg, C.R. (1987) 'Parallel Networks that Learn to Pronounce English Text', *Complex Systs*, **1**: 145–68.

Seneff, S. *et al.* (1998) 'Galaxy-II: A reference architecture for conversational system development', in: *The 5th International Conference on Spoken Language Processing, Incorporating The 7th Australian International Speech Science and Technology Conference, Sydney Convention Centre, Sydney, Australia, 30th November - 4th December 1998. ISCA*. 10.21437/ICSLP.1998-478.

Sonderegger, M. (2011) 'Application to Graph Theory to an English Rhyming Corpus', *Computer Speech and Language*, **25**: 655–78.

Tanasescu, C., Paget, B., and Inkpen, D. (2016) *Automatic Classification of Poetry by Meter and Rhyme 6. Proceedings of the twenty-Ninth International Florida Artificial Intelligence Research Society Conference (FLAIRS 2026)*. Association for the Advacement of Artificial Intelligence. https://cdn.aaai.org/ocs/12923/12923-60593-1-PB.pdf

Tizhoosh, H.R., and Dara, R.A. (2006) 'On Poem Recognition', *Pattern Analysis and Applications*, **9**: 325–38. https://doi.org/10.1007/s10044-006-0044-8

Tizhoosh, H.R., Sahba, F., and Dara, R. (2008) 'Poetic Features for Poem Recognition: A Comparative Study', *Journal of Pattern Recognition Research*, **3**: 24–39. https://doi.org/10.13176/11.62

Van Oostendorp, M. (2014) 'Introducing a Scansion Machine for Dutch Poetry and Prose', *Loquens*, **1**: e002. https://doi.org/10.3989/loquens.2014.002

Weide, R. (1998) *The CMU Pronunciation Dictionary*. http://www.speech.cs.cmu.edu/cgi-bin/cmudict, (accessed 17 Jan. 2024).

Wolf, T. *et al.* (2019) 'HuggingFace's Transformers: State-of-the-art Natural Language Processing'. https://doi.org/10.48550/ARXIV.1910.03771, (accessed 17 Jan. 2024)

Yang, Y. (1999) 'An Evaluation of Statistical Approaches to Text Categorization', *Information Retrieval*, **1**: 67–88.

Yang, Y., and Pederesen, J.O. (1997) 'A comparative study on feature selection in text categorization', *Proceedings of the Fourteenth International Conference on Machine Learning. Presented at the ICML'97, Nashville, 8-12 July 1997*, 412–20.

Appendix

Table A1 Summary of the reviewed tools including availability of source code, GUI, websites and code repositories.

Tool	Language	Type	Source code	GUI	Websites and code repositories[a]
Metrical annotation					
Logan (1988)	English	Rule-based	No	No	Not available
Hayward (1991)	English	Data-driven	No	No	Not available
Scandroid (Hartman 1996)	English	Rule-based	Yes	Yes	http://oak.conncoll.edu/cohar/Programs.htm
AnalysePoems (Plamondon 2006)	English	Rule-based	No	No	
Calliope (McAleese 2007)	English	Rule-based	No	No	
Greene, Bodrumlu, and Knight (2010)	English	Data-driven	No	No	
Prosodic (Heuser *et al.* 2010)	English and Finnish	Rule-based	Yes	Yes	http://prosodic.stanford.edu https://github.com/quadrismegistus/prosodic
ZeuScansion (Agirrezabal *et al.* 2013)	English (Spanish)	Rule-based	Yes	No	https://github.com/manexagirrezabal/zeuscansion
Poesy (Algee-Hewitt *et al.* 2014)	English	Rule-based	Yes	No	https://github.com/quadrismegistus/poesy
SPARSAR (Delmonte 2014)	English	Rule-based	Yes	No	https://sparsar.wordpress.com/
Tanasescu, Paget, and Inkpen (2016)	English	Data-driven	No	No	
Agirrezabal (2017)	English, Spanish and Basque	Data-driven	No	No	
KAIST (Kim, Zhao, and Zelalem 2018)	English	Data-driven	Yes	No	https://github.com/rokrokss/Automatic_Classification_of_Poetry_by_Neural_Scansion
Gervás (2000)	Spanish	Rule-based	No	No	
ADSO (Navarro-Colorado 2018)	Spanish	Rule-based	Yes	No	https://github.com/bncolorado/adsoScansionSystem
Rantanplan (De la Rosa *et al.* 2020b)	Spanish	Rule-based	Yes	No	https://github.com/linhd-postdata/rantanplan http://postdata-prototype.linhd.uned.es/poetry-lab.php
Jumper (Marco *et al.* 2021)	Spanish	Rule-based	Yes	No	https://github.com/grmarco/jumper
De la Rosa *et al.* (2021)	Spanish English and German	Data-driven	Yes	No	https://github.com/linhd-postdata/alberti
Papakitsos (2011)	Ancient Greek	Rule-based	No	No	
Greek_scansion (Conser, 2017)	Ancient Greek	Data-driven	Yes	No	https://github.com/aconser/greek_scansion
Greek_scansion (Schumann, 2020)	Ancient Greek	Rule-based	Yes	No	https://github.com/anetschka/greek_scansion
Corpus of Czech Verse (Ibrahim and Plecháč 2011)	Czech	Rule-based	No	Yes	http://www.versologie.cz/en/nastroje.html http://www.versologie.cz/en/kcv.html http://www.versologie.cz/dcm/index_en.html http://www.versologie.cz/gunstick/index_en.php
KBNL scansion (Koppelaar and Van Oostendorp 2013)	Dutch	Rule-based	Yes	Yes	https://lab.kb.nl/tool/scansion-generator
De Sisto (2020)	Renaissance Dutch	Data-driven	No	No	
Middle Dutch Syllabifier (Haverals et al. 2019)	Middle Dutch	Data-driven	Yes	No	https://github.com/WHaverals/syllabify https://zenodo.org/record/2582976#.XylqBC2w3OR
Metrometer (Beaudouin and Yvon 1996)	French	Rule-based	No	No	

(continued)

Table A1. (continued)

Tool	Language	Type	Source code	GUI	Websites and code repositories[a]
Reddy and Knight (2011)	French and English	Data-driven	No	No	
Anamétre (Delente and Renault 2015)	French	Rule-based	No	No	
Metricalizer (Bobenhausen 2011)	German	Rule-based	Yes	Yes	https://metricalizer.de/en/
Haider (2021)	German and English	Data-driven	Yes	No	https://github.com/tnhaider/metrical-tagging-in-the-wild
Automatische MittelHochdeutsche Metrik 2.0 (Dimpel, 2015)	Middle High German	Rule-based	Yes	Yes	http://www.archiv.mediaevistik.germanistik.phil.uni-erlangen.de/cgi-bin/metrik/metrik2.pl
MHG-scansion model (Estes and Hench 2016)	Middle High German	Data-driven	Yes	Yes	https://github.com/henchc/MHG-scansion https://henchc.github.io/MHG-Scansion-WebApp/
Pedecerto (Colombi *et al.* 2012)	Latin	?[b]	No	Yes	http://www.pedecerto.eu http://www.pedecerto.eu/public/scansioni/scansioni
Calliope scansio (Jacquet 2013)	Latin	Rule-based	Yes	No[c]	https://github.com/ChristopheJacquet/Calliope
Rainsford and Scrivner (2014)	Old Occitan	Rule-based	No	No	
Aoidos (Mittmann, von Wangenheim, and dos Santos 2016)	Portuguese	Rule-based	No	Yes	https://aoidos.ufsc.br
Araújo and Mamede (2002)	Portuguese	Rule-based	No	No	
StarLing (Pilshchikov and Starostin 2011)	Russian	Rule-based	No	No	
Barakhnin, Kozhemyakina, and Borzilova (2019)	Russian	Rule-based	No	Yes	http://poem.ict.nsc.ru/
Poetry recognition					
Tizhoosh and Dara (2006)	English	Data-driven	No	No	
Tizhoosh, Sahba, and Dara (2008)	English	Data-driven	No	No	
Lorang *et al.* (2015)	English	Data-driven	No	No	
Kilner and Fitch (2017)	English	Data-driven	No	No	
Foley (2019)	English	Data-driven	Yes	Yes	https://github.com/jjfiv/poetry-identification http://ciir.cs.umass.edu/downloads/poetry
Poetry visualization					
Kaplan and Blei (2007)	English	Rule-based	No	No	
Myopia (Chaturvedi *et al.* 2012)	English	Rule-based	No	Yes	
Poem Viewer (Abdul-Rahman *et al.* 2013)	English	Rule-based	No	Yes	https://nms.kcl.ac.uk:8443/poemvis/index.html
ECPA tools (Huber 2018)	English	Rule-based	No	Yes	https://www.eighteenthcenturypoetry.org/ https://www.eighteenthcenturypoetry.org/works/peh89-w0080.shtml#visualization
Dante Visualized (Ferraro 2020)	Italian	Data-driven	Yes	No	https://github.com/ginestra/dante-visualised
DISCOVER (Bermúdez-Sabel, Ruiz Fabo, and Martínez Cantón 2023)	Spanish	Rule-based	No	Yes	http://prf1.org/discover/

[a] All webpages were accessed on 14 January 2022.
[b] No information is provided about the structure of the tool.
[c] An interface version is mentioned in the code repository; however, the link does not seem to work.