Studia Metrica et Poetica 11.1, 2024, 44–61

Automatic Poetic Metre Detection for Czech Verse

Kristýna Klesnilová, Karel Klouda, Magda Friedjungová, Petr Plecháč*

Abstract: Metrical analysis of verse is an essential and challenging task in the research on versification consisting of analysing a poem and deciding which metre it is written in. Thanks to existing corpora, we can take advantage of data-driven approaches, which can be better suited to the specific versification problems at hand than rule-based systems.

This work analyses the Czech accentual-syllabic verse and automatic metre assignment using the vast and annotated Corpus of Czech Verse. We define the problem as a sequence tagging task and approach it using a machine learning model and many different input data configurations. In comparison to this approach, we reimplement the existing data-driven system KVĚTA.

Our results demonstrate that the bidirectional LSTM-CRF sequence tagging model, enhanced with syllable embeddings, significantly outperforms the existing KVĚTA system, with predictions achieving 99.61% syllable accuracy, 98.86% line accuracy, and 90.40% poem accuracy. The model also achieved competitive results with token embeddings. One of the most interesting findings is that the best results are obtained by inputting sequences representing whole poems instead of individual poem lines.

Keywords: metrical analysis, metre detection, poetry, Czech accentual-syllabic verse, Corpus of Czech Verse, KVĚTA, BiLSTM-CRF, Word2Vec

Introduction

Metrical analysis of verse is a challenging task not only because of the complexity of its subtasks, such as splitting a word into syllables, deciding

* Authors' addresses: Kristýna Klesnilová, Institute of Formal and Applied Linguistics, Faculty of Mathematics and Physics, Charles University, Malostranské náměstí 25, 118 00, Prague, Czech Republic, email: kristyna.klesnil@gmail.com; Karel Klouda, Faculty of Information Technology, Czech Technical University in Prague, Thákurova 9, 160 00 Prague 6 Czech Republic, email: karel.klouda@fit.cvut.cz; Magda Friedjungová, Faculty of Information Technology, Czech Technical University in Prague, Thákurova 9, 160 00 Prague 6 Czech Republic, email: magda.friedjungova@fit.cvut.cz; Petr Plecháč, Institute of Czech Literature, Czech Academy of sciences, Na Florenci 1420/3, 110 00 Prague, Czech Republic, email: plechac@ucl.cas.cz.

https://doi.org/10.12697/smp.2024.11.1.02

whether a syllable is accented or long[1], etc., but also because poets tend to be creative and do not always follow the metrical norms precisely. Therefore, all algorithms (as well as human researchers) and proposed approaches must consider these deviations from the norm.

Algorithmic metrical analysis used to be performed using rule-based approaches; for example, see (Navarro-Colorado 2018, Anttila, Heuser 2016, Plecháč 2016, Bobenhausen 2011). These approaches often require the presence of applied linguists to define constraints and rules. Rule-based systems cannot often generalize well and are tailored to the language and domain by their nature.

To overcome the limitations of rule-based approaches and explore the possibilities of machine learning methods, we use a state-of-the-art data-driven model for sequence tagging called BiLSTM-CRF (Reimers, Gurevych 2017a) that models the prosodic features of syllables. The use of this model has been recently proposed by researchers in the domain of versification (Agirrezabal et al. 2017), but it has not yet been adapted and tested on the Czech accentual-syllabic poems included in the Corpus of Czech Verse (CCV[2], henceforth) (Plecháč 2016) which contains 1305 books of poetry. The current best model applicable to the Czech language is KVĚTA (Plecháč 2016). KVĚTA employs a statistical approach to the metrical analysis. We reimplement KVĚTA to be able to compare and assess the results of our experiments. Additionally, we evaluate our model not only on the usual syllable level but also on the line and poem levels.

Our main contributions are:

- Based on (Plecháč 2016) we reimplement data-driven approach KVĚTA.
- We train the bidirectional long short-term memory neural network with conditional random fields (BiLSTM-CRF) as a sequence tagging model (Huang et al. 2015) and propose various input configurations to show their benefits using this model.
- We prove the success of BiLSTM-CRF model for the metrical tagging of Czech accentual-syllabic verse and its benefits over using the KVĚTA approach.
- The code of whole pipeline including KVĚTA and BiLSTM-CRF model is publicly available on GitHub[3].

[1] The Czech language differentiates between long and short vowels; this is similar to the difference between the English words "dip" and "deep". Vowel length is independent of lexical stress. This distinction is important for accurately analyzing and assigning metre, especially in quantitative metres.

[2] The description is available at https://versologie.cz/en/kcv.html, and the complete corpus is available at https://github.com/versotym/corpusCzechVerse.

[3] https://github.com/magdafriedjungova/metre_detection

Metrical Theory

The poems included in the CCV are mostly written in the Czech accentual-syllabic verse. Accentual-syllabic versification combines syllabic and tonic versification, considering the number of syllables and whether they are accented or not (Ibrahim et al. 2013). Below, we introduce terms and notations needed for metrical analysis.

The basic metrical unit of a verse is called a foot. In the accentual-syllabic verse, it represents a group of at least two syllables that are repeated regularly throughout the verse. One foot consists of strong and weak positions. We label strong positions with the letter S and weak positions with W. If there are two weak positions within a foot, the first is labelled using V. Table 1 presents all types of feet encountered within the Czech accentual-syllabic verse. (Ibrahim et al. 2013)

Table 1: Czech accentual-syllabic verse feet (positions inside brackets can be omitted)

Foot	Feet pattern
Iamb	$W_0\ S_1\ W_1\ S_2 \dots S_n\ (W_n)$
Trochee	$S_1\ W_1\ S_2\ W_2 \dots S_n\ (W_n)$
Dactyl	$S_1\ V_1\ W_1\ S_2\ V_2\ W_2 \dots S_n\ ((V_n)\ W_n)$
Dactyl with anacrusis (Amphibrach)	$W_0\ S_1\ V_1\ W_1\ S_2\ V_2\ W_2 \dots S_n\ ((V_n)\ W_n)$
Dactylotrochee	$S_1\ (V_1)\ W_1\ S_2\ (V_2)\ W_2 \dots S_n\ ((V_n)\ W_n)$
Dactylotrochee with anacrusis	$W_0\ S_1\ (V_1)\ W_1\ S_2\ (V_2)\ W_2 \dots S_n\ ((V_n)\ W_n)$

All standard Czech accentual-syllabic metrical patterns can be expressed by the following regular expression4:

^W?(SWW?)*(SW?)?$,

where V and W weak positions are annotated with the same symbol (Plecháč 2016). It is important to note that foot and word are two different concepts. Their boundaries do not have to overlap.

The repetition of metrical feet in a verse forms a metre – the abstract outline of a verse (Ibrahim et al. 2013). The main complexity of the metre

4 The regular expression says that after the beginning (denoted by ^), there might be one weak position (W?) followed by zero or more repetitions of SW or SWW. The pattern might end (the end is represented by $) with S or SW.

assignment task lies in the difference between a rhythm and a metre. When talking about the accentual-syllabic verse, metre is expressed by the regular alternation of strong and weak positions. On the other hand, rhythm is the poet's actual implementation of the metre using the alternation of accented and non-accented syllables.

For the accentual-syllabic verse, the underlying concept is that S-positions correspond to accented syllables and V-positions and W-positions to non-accented ones. However, in reality, all positions can correspond to both accented and non-accented ones. In many situations, the poet has the freedom to choose whether to use an accent. As a result, one metre can be expressed by multiple rhythmical patterns.

For the Czech accentual-syllabic verse, complex rules exist that determine in which situations it is possible to use accented or non-accented syllables. The rules were obtained through a thorough analysis of many poems. Naturally, not all poems obey these rules. (Ibrahim et al. 2013)

In the accentual-syllabic verse, three types of line endings (called clauses) are distinguished based on the last position of a verse: masculine, feminine, and acatelectic. Verses with masculine endings end with the S-position. When the verse ends with a W-position, it is either feminine or acatalectic. The acatalectic verses end with the SVW position pattern, and the feminine verses end with the SW pattern.

As the last note on the verse theory, let us mention the verse multimetry and poem polymetry. A verse is labelled multimetric when its rhythmical pattern can correspond to more metres. The correct metre of such a verse is then selected based on the surrounding context. A similar concept to multimetry is polymetry, but this time regarding a whole poem. A poem is polymetric when some of its verses have different metres assigned than others, and the occurrences of such metres are more or less predictable. (Ibrahim et al. 2013)

See Figure 1 for a metrically tagged example of Czech accentual-syllabic poetic text. The first verse represents a dactyl with four feet and a masculine clause (D4m). The second verse is a dactyl with three feet and an acatalectic clause (D3a). A dactyl (with anacrusis) verse with three feet and a feminine clause (Da3f) follows, and the last verse stands for dactylotrochee with anacrusis with three feet and a feminine clause (DTa3f). Although accented, the first syllable of the fourth verse represents a weak position. (Ibrahim et al. 2013)

S V W S V W S V W S
Pra|mé|nek zaz|vo|nil ti|še a rád. (D4m)

S V W S V W S V W
V srd|ci mém poz|dil se lis|to|pad (D3a)

W S V W S V W S W
a stu|du|ju vla|stní své ry|sy (Da3f)

W S V W S W S W
já za|po|mněl um|řít kdy|si. (DTa3f)

Figure 1: Accentual-syllabic metrical tagging. Accented syllables are underlined. Strong (S) and weak (V or W) positions and line tags containing metre, number of feet, and clauses are annotated.

In addition to standard metres presented in Table 1, some particular types of verse (for example, hexameter, pentameter, and ghazal poems) can also be found in the Czech accentual-syllabic tradition. However, we do not consider them here. For more information, see (Ibrahim et al. 2013; Plecháč 2016).

Corpus of Czech Verse

The Corpus of Czech Verse (CCV), used in this work, is lemmatised, phonetically, morphologically, metrically, and strophically annotated corpus of Czech poetry from the 19th century and the beginning of the 20th century. Each lexical unit is provided with information about its basic word form (lemma), phonetic transcription and grammatical categories; each verse line is provided with information about its type of metre (iamb, trochee, etc.), length (n-foot), type of the end of a line (masculine, feminine, etc.) and the metrical pattern (currently, only accentual-syllabic verse lines are annotated in terms of metrics). There is also a higher-level annotation available on rhyme pairs or n-some and fixed forms (e.g., sonnet, rondel). In the metrical and strophical descriptions, searching using the Database of Czech Metres is possible[5].

This corpus is among the world's most extensive poetic databases, with public (not copyright protected) part containing 66 428 poems, 2 310 917 lines, and 12 636 867 words (Plecháč, Kolár 2015; Plecháč 2021).

[5] https://versologie.cz/dcm/index_en.html

Metrical Analysis of Czech Accentual-syllabic Verse

Syllabification

When performing a metrical verse analysis, the first step is to split the text into syllables. Inside every syllable, there is a sonority peak. In Czech language, the sonority peak can be expressed by:

- vocal or diphthong,
- sonorant r or l (when positioned between two consonants or at the end of a word after at least one consonant, e.g. *krk* (neck), *vlk* (wolf))
- (very exceptionally) nasals or sibilants, e.g. *osm* (eight), *pst* (shhh).

Words without a sonority peak, e.g. non-syllabic prepositions v, k, s, and z, form one syllable with the first syllable of the following word.

Determining syllable boundaries is a challenge, even for a native speaker. For example, the word *houska* (bun) consists of two syllables, but there are two correct ways to split it (*hous-ka* or *hou-ska*?) (Ibrahim et al. 2013). Despite its complexity, two ways exist to perform syllabification using a computer – create a phonetic transcription of text or by hyphenation tools.

Using Phonetic Transcription

The phonetic transcription represents speech sounds, i.e. phones, where phoneme is the smallest sound unit. These units are then clustered into syllables. For the Czech language, we can use KVĚTA program (Plecháč 2016), which applies a sequence of rules to the input words and obtains their phonetic transcriptions. This approach is highly effective because the Czech orthography is highly phonemic. The only words that cannot be transcribed using a set of rules are words containing bigrams *au*, *ou*, *eu* and foreign words. However, these bigrams are transcribed using a manually built token-diphthong library and applying a few additional rules.

Using Hyphenation Tools

Another way to perform syllabification is to use some tools to hyphenate text. Hyphenation is used in every document preparation system (e.g. TEX or web browsers) to decide where a word can be split to continue on the following line. The approach of using hyphenation tools for syllabification has already been tested in (Haider 2021) when performing a metrical analysis of English and German verse using machine learning. For English verse, they decided to train BiLSTM-CRF syllabification model instead; for German verse, they used an ensemble of hyphenation tools and heuristic corrections.

Metre Assignment

In the past, there were presented two approaches to metre assignment of Czech accentual-syllabic verse: a rule-based approach that the KVĚTA program used previously (Ibrahim, Plecháč 2011) and a data-driven approach that the current version of the KVĚTA program is using (Plecháč 2016). We will describe the later version here briefly.

A data-driven system KVĚTA, presented and described in (Plecháč 2016), is based on syllable representation when every syllable is represented by "syllable class". The "syllable class" is a data class containing Boolean parameters extracted from the Czech accent rules (e.g., word-initial syllable, word-final syllable, content/function word). A poem is then internally represented as a list of lists of "syllable classes". The division into stanzas is not considered.

The next step of the algorithm is to take for each poem only the number of syllables in each line and generate all possible metres that could fit it. Not only standard accentual-syllabic metres are generated, but also accentual-syllabic imitations of quantitative syllabic strophes and accentual-syllabic imitations of the quantitative hexameter, pentameter, and elegiac couplet. Ghazal poems are generated as well. The V-positions are not distinguished and are represented as W-positions.

To be able to assign metres, the algorithm must pre-calculate the values of "metrical coefficient" from the corpus data for every generated metre. For every "syllable class", the "metrical coefficient" value is the conditional probability that a strong or weak position in the corresponding metrical pattern realizes the syllable class. For details on computation, see (Plecháč 2016). Finally, the metre with the highest "metrical coefficient" is selected.

The described data-driven system is more suitable for metrical tagging Czech accentual-syllabic verse than the rule-based one. It can distinguish whether a monosyllable is accented or not. It allows for more advanced rules as, for instance, the monosyllabic prepositions propers are also considered. It also tags hexameters and pentameters, which are tagged within the CCV. Again, polymetric tagging of poems is not possible.

Machine Learning Approach

Metrical analysis can also be performed as a sequence tagging task solved using machine learning. Sequence tagging is an important natural language processing task consisting of receiving a text sequence on input and outputting it tagged. The sequence tagging tasks represent, e.g. part of speech (POS) tagging or named entity recognition (NER) tagging, which aims to identify named entities within a text (people, locations, organizations, etc.). In this field, statistical models such as conditional random fields (CRF) or deep neural network-based models such as Long Short-term Memory (LSTM) and Bidirectional Encoder Representations from Transformers (BERT) are very popular. Statistical models can solve the tasks based on data such as the word itself, its position in the sentence, and the words around it. In addition, deep learning models can capture complex patterns and dependencies in the text.

In Haider 2021, CRF, BERT model and BiLSTM-CRF model are tested on English and German verse corpora. (Agirrezabal et al. 2017) experimented using perceptron, hidden Markov model, conditional random fields, and BiLSTM-CRF on English and Spanish verse corpora. Both articles evaluate syllable-level and line-level accuracies and obtain the best results using BiLSTM-CRF.

In Haider 2021, individual syllables are sent to the input and custom syllable embeddings are pre-trained from verse corpora using the Word2Vec algorithm (Mikolov et al. 2013). BiLSTM-CRF uses three BiLSTM layers with 100 recurrent units in each layer and uses a linear-chain CRF classifier. Variational dropout is used, with a 25% drop in output and recurrent connections. No character-based representation of syllables is used, as it, according to the authors, hurts both speed and accuracy.

In contrary to Haider 2021, in Agirrezabal et al. 2017, BiLSTM-CRF is used with BiLSTM character-based representation of input tokens. Three different input types are tested: individual syllables, word tokens, and individual syllables with word boundaries. For English, the best results are obtained for individual syllables with word boundaries in the input. For Spanish, the best results are obtained when word tokens are inputted.

Due to the results presented in the mentioned papers, we decided to use a similar approach described in the following section. This machine learning approach will be compared to our implementation of data-driven KVĚTA.

Implementation and Experiments

Used Datasets

Two different datasets obtained from the CCV are used. The first dataset of 57 339 poems contains no polymetric poems, unrecognised metrical positions, multimetric verses, and annotation errors.

The second dataset of 59 661 poems contains no unrecognised metrical positions, multimetric verses, and annotation errors. However, the polymetric poems are included.

Both datasets are divided into training, validation, and testing data using a ratio of 70:15:15. The KVĚTA implementation is trained on the training and validation data and tested on the testing data. The BiLSTM-CRF model is trained on the training data, fine-tuned during training on the validation data, and final results are obtained for the previously unseen testing data.

Data-driven KVĚTA Implementation

In our implementation of the KVĚTA data-driven approach, a phonetic transcription algorithm is not implemented. Instead, phonetic transcriptions already generated by KVĚTA and published within the CCV are used. These transcriptions use the X-SAMPA[6] format, which is parsed to obtain the number of sonority peaks and, therefore, the number of syllables for each word.

In the paper by Petr Plecháč (2016) explaining the KVĚTA data-driven approach, the final "syllable class" count is reduced to only 12 different "syllable classes". Not all the reductions applied are explained in the paper, so our implementation got to 15 different "syllable classes".

Our implementation generates the same metres as KVĚTA – the standard accentual-syllabic metres; imitations of quantitative syllabic strophes; imitations of hexameter, pentameter, and elegiac couplet; and ghazal poems.

For metre assignment, the conditional probability of a "syllable class" being realized by a strong or weak position is not counted using Bayes' Theorem as it is inside KVĚTA (for details see Plecháč 2016) but is counted directly. In our opinion, counting the probability using Bayes' theorem is, in this case,

---

[6] X-SAMPA is a machine-readable phonetic alphabet frequently used in technical applications like speech recognition or speech synthesis. For more details see SAMPA / X-SAMPA Transcription of Czech, available from: https://fonetika.ff.cuni.cz/o-fonetice/foneticka-transkripce/czech- sampa/.

unnecessary and does not add any benefit compared to counting it directly. Moreover, the probabilities of strong and weak positions should not be considered equal, as the weak position is more likely to occur depending on the structure of all the accentual-syllabic feet. The probabilities are estimated from training and validation datasets.

When selecting the metre, the "metrical coefficients" of the individual line patterns and the overall "metrical coefficients" are calculated the same as in KVĚTA. When the same metrical pattern is generated with different metre names, standard accentual-syllabic metres are selected with priority as in KVĚTA.

BiLSTM-CRF Implementation

The BiLSTM-CRF model combines the bidirectional LSTM (simple LSTM was presented in Hochreiter, Schmidhuber 1997) and the CRF (stands for conditional random fields) network into one model. It can use information about past and future inputs when predicting the current tag. Thanks to a CRF, the model can also use sentence level tag information. It represents the current state-of-the-art model for standard sequence tagging tasks such as POS or NER tagging (Huang et al. 2015). For the details on BiLSTM-CRF architecture, see Huang et al. 2015. We choose the publicly available implementations of BiLSTM-CRF for sequence tagging, where BiLSTM-CRF uses three BiLSTM layers with 100 recurrent units in each layer and a linear-chain CRF classifier. This implementation is optimized for speed; it groups sentences with the same lengths together during training, and thanks to that, it is much faster than other BiLSTM-CRF implementations (UKPLab, 2018). The same implementation is used in Haider 2021.

Input Data Format

When training the BiLSTM-CRF model, two different input data formats are tested: token-level (token is, roughly, a word) and syllable-level. Both formats are inputted into the model using two approaches: one input sequence represents a line in a poem, and one represents a whole poem.

The model is inputted a sequence of individual tokens (words) from the poem for the token-level input data format. For each token, a sequence of metrical positions is predicted. This sequence can be empty as well when the token is non-accented.

As for the syllable-level input data format, a sequence of individual syllables from the poem is fed into the model. Because the CCV lacks annotation of the syllable boundaries, we use X-SAMPA phonetic transcriptions to obtain syllable representations with correct syllable count. Each word's X-SAMPA transcription is divided into X-SAMPA syllables using transcriptions of sonority peaks. For non-syllabic words without a sonority peak, no X-SAMPA syllables are generated.

As an example, for the token *blýskajícím* with X-SAMPA transcription bli:skaji:t_si:m the following four X-SAMPA syllables are obtained: [bli:, ska, ji:, t_si:m].

For every syllable, exactly one metrical position is predicted as an output of the model. It means the model decides for every syllable based on its position within the word if the syllable is weak or strong.

Input Features

Representation of a word inputted into the BiLSTM-CRF network can consist of multiple concatenated vectors: word embedding, capitalisation feature, and character-based representation (Reimers, Gurevych 2017b).

Following Haider 2021, no character-based representation of the input is used. The decision is made not to use the capitalisation feature, as the capitalisation of words is of minor importance for this task, and words inside poems usually do not contain numerals.

As word embeddings, Word2Vec embeddings are pre-trained on the training data.

The implementation supports inputting other features in addition to the input tokens. Following intuition from Plecháč 2016, the author of a poem, year of publication, POS tag, or lemma is also inputted. An embedding of size ten is assigned automatically to every input value and concatenated with other input embeddings.

Training and Hyperparameters

Even though the implementation is highly configurable, the hyperparameters[7] are not further fine-tuned, as training of one input configuration takes 1 to 2 days (CPU is used, training on GPU is not tested). The hyperparameters are set according to the recommendations in Haider 2021 and Reimers, Gurevych 2017b.

Following the findings of Reimers, Gurevych 2017b, the Nadam optimizer (Dozat 2016) is utilized. Thirty-two sentences are used for mini-batch training, as the training set is relatively large. The gradient normalization is performed with threshold value 1 to overcome the exploding gradient problem. All models are trained for 15 epochs. Training is stopped earlier if the best validation score does not increase for more than five training epochs.

Evaluation

We used standard classification accuracy as usual in supervised learning tasks to evaluate the metrical analysis approaches implemented. The following three accuracies are taken into account:

- Syllable-level accuracy: the percentage of syllables for which the metrical position assigned by the model is the same as the referential.
- Line-level accuracy: The percentage of lines for which the metrical pattern assigned by the model is the same as the referential.
- Poem-level accuracy: The percentage of poems for which the metrical pattern assigned by the model is the same as the referential.

For BiLSTM-CRF models with token input, syllable accuracies cannot be evaluated because the reference and predicted syllable counts tend to differ.

[7] In neural networks, each neuron is represented by a set of parameters (or weights). Such parameters are determined based on the data during the training process. However, some parameters of the model, e.g., the number of neurons, must be set before the training starts. These parameters are called hyperparameters. Experimenting with hyperparameters may be computationally exhausting, as each new hyperparameter means completely rerunning the training process.

Results and Discussion

As mentioned above, we performed exhaustive experiments using different combinations of inputted data to the BiLSTM-CRF model. For the results on both datasets used, see Tables 2 and 3.

With the best input configurations, BiLSTM-CRF returns better results than the KVĚTA implementation regarding all three evaluated accuracies. The KVĚTA approach was reimplemented with minor changes to the original functionality, so the obtained predictions are probably slightly different than they would be with the original program (which is not publicly available or open-sourced).

The syllable input returns better results than the token input, especially when comparing the poem-level accuracy. However, considering only the line-level accuracy, the token input results are promising. Note that for token input, prior syllabification is not necessary during preprocessing. An interesting observation is that inputting a lemma significantly improves token input results. Generally, the more additional features inputted (author of a poem, year of publication, POS tag, lemma), the better (or at least not worse) the results.

One of the most interesting findings of this work is how much inputting sequences representing whole poems rather than poem lines improves the results (especially the poem-level accuracy and, perhaps even more surprisingly, the syllable-level accuracy). The best results are obtained from inputting sequences representing whole poems with all additional features and poem line indices on the input. Poem line indices allow the model to distinguish different lines of a poem. To our knowledge, the approach of inputting sequences representing entire poems may not have been previously tested.

One may also notice that we did not use "syllable class" as used by KVĚTA implementation. We decided to keep the data preparation as simple as possible, and the "syllable class" construction required human editing. Even without it, we achieved better results than the KVĚTA system.

Table 2: Results for the first dataset (no polymetric verses, no verse multimetry). The syll. stands for syllable, acc. for accuracy, and idxs for indexes. Numbers are shown in percentages.

Approach	Syll. acc.	Line acc.	Poem acc.
Our KVĚTA implementation	97.02	81.88	69.83
BiLSTM-CRF with X-SAMPA syll. embeddings Input: X-SAMPA syllables	97.93	95.67	66.78
BiLSTM-CRF with X-SAMPA syll. embeddings Input: X-SAMPA syllables, authors, years	98.14	95.97	66.29
BiLSTM-CRF with X-SAMPA syll. embeddings Input: X-SAMPA syllables, authors, years, POS tags	98.69	96.85	72.34
BiLSTM-CRF with X-SAMPA syll. embeddings Input: X-SAMPA syllables, authors, years, POS tags, lemmas	98.74	96.94	72.98
BiLSTM-CRF with X-SAMPA syll. embeddings, poem as line Input: line idxs, X-SAMPA syllables, authors, years, POS tags, lemmas	**99.71**	**99.17**	**92.51**
BiLSTM-CRF with tokens embeddings Input: tokens	-	74.38	8.69
BiLSTM-CRF with tokens embeddings Input: tokens, authors, years	-	75.91	10.29
BiLSTM-CRF with tokens embeddings Input: tokens, authors, years, POS tags	-	76.94	11.17
BiLSTM-CRF with tokens embeddings Input: tokens, authors, years, POS tags, lemmas	-	90.01	31.83
BiLSTM-CRF with tokens embeddings, poem as line Input: line idxs, tokens, authors, years, POS tags, lemmas	-	95.47	55.74

Table 3: Results for the second dataset (polymetric verses, no verse multimetry). The syll. stands for syllable, acc. for accuracy, and idxs for indexes. Numbers are shown in percentages.

Approach	Syll. acc.	Line acc.	Poem acc.
Our KVĚTA implementation	96.08	80.64	68.26
BiLSTM-CRF with X-SAMPA syll. embeddings Input: X-SAMPA syllables	97.78	95.31	62.91
BiLSTM-CRF with X-SAMPA syll. embeddings Input: X-SAMPA syllables, authors, years	98.21	96.09	67.20
BiLSTM-CRF with X-SAMPA syll. embeddings Input: X-SAMPA syllables, authors, years, POS tags	98.72	96.96	73.41
BiLSTM-CRF with X-SAMPA syll. embeddings Input: X-SAMPA syllables, authors, years, POS tags, lemmas	98.83	97.12	73.06
BiLSTM-CRF with X-SAMPA syll. embeddings, poem as line Input: line idxs, X-SAMPA syllables, authors, years, POS tags, lemmas	**99.61**	**98.86**	**90.40**
BiLSTM-CRF with tokens embeddings Input: tokens	-	72.14	7.56
BiLSTM-CRF with tokens embeddings Input: tokens, authors, years	-	76.60	9.64
BiLSTM-CRF with tokens embeddings Input: tokens, authors, years, POS tags	-	78.40	11.27
BiLSTM-CRF with tokens embeddings Input: tokens, authors, years, POS tags, lemmas	-	91.42	35.39
BiLSTM-CRF with tokens embeddings, poem as line Input: line idxs, tokens, authors, years, POS tags, lemmas	-	95.91	58.98

In future work, further experiments could be performed with the BiLSTM-CRF model, such as fine-tuning the model hyperparameters (e.g., optimizer with its settings, number of BiLSTM layers, number of recurrent units in one BiLSTM layer), using character-based representations of the word embeddings, or training and using different word embeddings (e.g., GloVe, FastText, ELMo). Syllabification without the need for phonetic transcription could be tried either by using hyphenation approaches or by training a machine learning model. Another experiment could be multitask learning (e.g. predicting the metrical pattern and the POS tag together), predicting the name of the metre (e.g., dactyl, trochee) instead of or alongside the metrical pattern in a multitask setup. Another natural continuation is the experiments using different poetic corpora to verify the model usability in the task of metre detection; however, some comparisons have already been made (Haider 2021). Alternatively, transfer learning between poetic corpora in different languages can be performed. Furthermore, other machine-learning architectures (e.g. transformers) could be tested.

Conclusion

Our main objective was to try a machine-learning approach to metrical analysis using Czech accentual-syllabic verse, specifically Corpus of Czech Verse, and compare it to the standard KVĚTA approach. The BiLSTM-CRF model was chosen and successfully trained on many input configurations. The best input configurations yield better results than our KVĚTA implementation for all evaluated accuracies (syllable-level, line-level, and poem-level). The most interesting finding is that the best results are obtained by inputting sequences representing whole poems with line indices on the input, allowing a model to distinguish different lines (especially the poem-level accuracy significantly improved). The approach of inputting sequences representing whole poems may never have been tested before. Overall, using the BiLSTM-CRF model for metrical tagging of Czech accentual-syllabic verse represents a great success and has many benefits over using the KVĚTA approach. It does not need to encode any complicated expert knowledge, as everything is learnt automatically by the machine learning model. Furthermore, it is completely automatic, unlike the KVĚTA approach, which sometimes needs human assistance.

BiLSTM-CRF proved to be a powerful model for various sequence tagging tasks, including metre detection. It can be used even for other languages; however, some steps of the process described in this paper are very language-specific.

References

Agirrezabal, Manex; Alegria, Iñaki; Hulden, Måns 2017. A Comparison of Feature-Based and Neural Scansion of Poetry. In: Mitkov, Ruslan; Angelova, Galia (eds.), *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*. Varna: INCOMA Ltd., 18–23. https://doi.org/10.26615/978-954-452-049-6_003

Anttila, Arto; Heuser, Ryan 2016. Phonological and metrical variation across genres. In: Hansson, Gunnar Ólafur; Farris-Trimble, Ashley; McMullin, Kevin; Pulleyblank, Douglas (eds.), *Proceedings of the 2015 Annual Meetings on Phonology, AMP 2015*, 12 pages. https://doi.org/10.3765/amp.v3i0.3679

Bobenhausen, Klemens 2011. The Metricalizer2 – Automated Metrical Markup of German Poetry. In: Küper, Christoph (ed.), *Current Trends in Metrical Analysis*. (Littera: Studies in Language and Literature 2). Berlin: Peter Lang, 119–131.

Dozat, Timothy 2016. Incorporating Nesterov Momentum into Adam. In: *Proceedings of the 4th International Conference on Learning Representations, ICLR 2016*, 4 pages. https://openreview.net/pdf/OM0jvwB8jIp57ZJjtNEZ.pdf

Haider, Thomas 2021. Metrical Tagging in the Wild: Building and Annotating Poetry Corpora with Rhythmic Features. In: Merlo, Paola; Tiedemann, Jorg; Tsarfaty, Reut (eds.), *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. S.l. (Online): Association for Computational Linguistics, 3715–3725. https://doi.org/10.18653/v1/2021.eacl-main.325

Hochreiter, Sepp; Schmidhuber, Jürgen 1997. Long Short-Term Memory. In: *Neural Computation* 9(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Huang, Zhiheng; Xu, Wei; Yu, Kai 2015. Bidirectional LSTM-CRF Models for Sequence Tagging. https://doi.org/10.48550/arXiv.1508.01991

Ibrahim, Robert; Plecháč, Petr 2011. Toward Automatic Analysis of Czech Verse. In: Scherr, Barry P.; Bailey, James; Kazartsev, Evgeny V. (eds.), *Formal Methods in Poetics*. Lüdenscheid: RAM, 295–305.

Ibrahim, Robert; Plecháč, Petr; Říha, Jakub 2013. *Úvod do teorie verše*. Praha: Akropolis.

Liang, Franklin Mark 1983. Word Hy-Phen-a-Tion by Com-Put-Er (Hyphenation, Computer). PhD thesis, Stanford University. https://tug.org/docs/liang/liang-thesis.pdf

Mikolov, Tomáš; Chen, Kai; Corrado, Greg; Dean, Jeffrey 2013. Efficient Estimation of Word Representations in Vector Space. https://arXiv.org/abs/1301.3781

Navarro-Colorado, Borja 2018. A Metrical Scansion System for Fixed-metre Spanish Poetry. *Digital Scholarship in the Humanities* 33(1), 112–127. https://doi.org/10.1093/llc/fqx009

Plecháč, Petr 2016. Czech Verse Processing System KVĚTA – Phonetic and Metrical Components. In: *Glottotheory* 7(2), 159–174. https://doi.org/10.1515/glot-2016-0013

Plecháč, Petr 2021. *Versification and Authorship Attribution*. Institute of Czech Literature. Prague: Karolinum. https://doi.org/10.14712/9788024648903

Plecháč, Petr; Kolár, Robert 2015. Korpus českého verše. In: Hlaváčová, Jaroslava (ed.), *Sborník semináře o digitálních zdrojích a službách ve společenských a humanitních vědách*. (WDH 2015). Praha: Univerzita Karlova, 74–77.

Reimers, Nils; Gurevych, Iryna 2017a. Reporting Score Distributions Makes a Difference: Performance Study of LSTM-networks for Sequence Tagging. In: Palmer, Martha; Hwa, Rebecca; Riedelm Sebastian (eds.), *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen: Association for Computational Linguistics, 338–348. https://doi.org/10.18653/v1/D17-1035

Reimers, Nils; Gurevych, Iryna 2017b. Optimal Hyperparameters for Deep LSTM-Networks for Sequence Labeling Tasks. https://doi.org/10.48550/arXiv.1707.06799