Identification of Concrete Poetry within a Modern-Poetry Corpus

Hussein Hussein
Department of Literary Studies
Free University of Berlin
Berlin, Germany
hussein@zedat.fu-berlin.de

Burkhard Meyer-Sickendiek
Department of Literary Studies
Free University of Berlin
Berlin, Germany
bumesi@zedat.fu-berlin.de

Timo Baumann
Department of Informatics
University of Hamburg
Hamburg, Germany
baumann@informatik.uni-hamburg.de

Abstract

This work aims to discern the poetics of concrete poetry by using a corpus-based classification focusing on the two most important techniques used within concrete poetry: semantic decomposition and syntactic permutation. We demonstrate how to identify concrete poetry in modern and postmodern free verse. A class contrasting to concrete poetry is defined on the basis of poems with complete and correct sentences. We used the data from *lyrikline*, which contain both the written as well as the spoken form of poems as read by the original author. We explored two approaches for the identification of concrete poetry. The first is based on the definition of concrete poetry in literary theory by the extraction of various types of features derived from a parser, such as verb, noun, comma, sentence ending, conjunction, and asemantic material. The second is a neural network-based approach, which is theoretically less informed by human insight, as it does not have access to features established by scholars. This approach used the following inputs: textual information and the spoken recitation of poetic lines as well as information about pauses between lines. The results based on the neural network are more accurate than the feature-based approach. The best results, calculated by the weighted F-measure, for the classification of concrete poetry vis-à-vis the contrasting class is 0.96.

1 Introduction

In the mid-1950s, there emerged the genre of “concrete poetry”, which, like Futurism and Dadaism, attempted to prevail over traditional poetry. For the founder, Eugen Gomringer, traditional poetry was based on semantic references in language, while concrete poetry understood language as pure material. Concrete poetry focuses the linguistic material of language by using segmentations and collages of everyday language in order to draw attention to the smallest particles of language—for example, individual letters, words, or word groups. Through new arrangements beyond the usual syntax, words and letters are freed from their accustomed context and are experienced anew.

“Concrete poetry” could be seen as a specific type of the “experimental poetry” that developed in the 20th century (Hartung 1975). The former is based on two principal techniques: The ‘decomposition’ of the semantic material of words, the poem’s vocabulary, or its syntactic connections on the one hand; and the reordering of syntactic structures of sentences, so-called ‘permutation’, on the other hand. Decomposition focuses on the words, reducing them either to syllables or even to letters. Permutation is based on a certain technique of repetition, varying the position of the words within the poem. Thus concrete poetry can be identified by three different devices: syllabic decomposition, lettristic decomposition, and syntactic permutation.

All three techniques are very typical for concrete poetry, although two of them—lettristic and syllabic decomposition—are already familiar from pre-war German poetry. The two forms of decomposition had originated in dadaistic “sound poems” during the 1920s, and were reused after 1945 by authors such as Isidore Isou, Ernst Jandl, Valerie Scherstjanoi, Franz Mon, Gerhard Rühm or Michael Lentz (Emanuely 2013). All these concrete poets used dadaistic techniques to reduce the poem’s semantics to syllables or letters, leaving only a few normal words (Mon 2012). A typical example for such dadaistic decomposition is the famous “Ursonate” by Kurt Schwitters, written between 1922 and 1932. The phonetic material of the Ursonate varies from the lettristic sound poem entitled “fmsbwtözäu / pggiv-...?mü” by Raoul Hausmann from 1921. Schwitters translated it into abstract syllable sequences which create one of the four ‘themes’ of the Ursonate: “Fümms bö wö tää zää Uu, / pögiff, / kwii Ee” (Mittelmeier 2016). Hausmann’s model was based on a lettristic decomposition in which the letters are isolated as the smallest elements of the written language, without reassembling them into words and sentences. In the syllabic decomposition of the Ursonate, on the other hand, the words are decomposed into syllables. Both these types of decompositions, i.e. lettristic and syllabic, can also be found after 1945 in concrete poetry and the Viennese group, which also limited the linguistic and semantic material of poems to syllables or letters—for example, in the poems of Valeri Scherstjanoi. An additional technique in concrete poetry is permutation, which is a conversion or exchange of words or parts of sentences, or a progressive combination and rearrangement of linguistic-semantic elements in a poem (Ernst 1992). This technique was originated in early modernism by Gertrude Stein. In Germany, the principle was made famous by the ‘concrete poet’ Eugen Gomringer, who explained it in his essay “vom vers zur konstellation” using the example of his poem ‘avenidas’:

“avenidas / avenidas y flores // flores / flores y mujeres // avenidas / avenidas y mujeres // avenidas y flores y mujeres y / un admirador” (Gomringer 1969). In the poem ‘irish’, quoted below, Gomringer also uses the principles of reduction and combination preceding the permutation. In the first step, Gomringer reduces the vocabulary – ‘irish’ consists of only seven different words – and syntax of the poem, insofar as the words are only linked by ‘and’. In the second step, the words in each line are combined differently than in the previous line, which in turn constitutes the permutation: the repeated words always have a different position in the poem. The following three examples by Ernst Jandl (Jandl 1966) and Eugen Gomringer illustrate these techniques:

Ernst Jandl: **schtzngrmm** (lettristic decomposition)	**Ernst Jandl:** **auf dem land** (syllabic decomposition)	**Eugen Gomringer:** **irish** (permutation)
schtzngrmm	rinininininininininDER	green and
schtzngrmm	brüllüllüllüllüllüllüllüllEN	sheep
t-t-t-t	schweineineineineineineineinE	sheep and
t-t-t-t	grununununununununZEN	cow
grrrmmmmm	hunununununununununDE	cow and
t-t-t-t	bellellellellellellellellEN	green
s---------c---------h	katatatatatatatatZEN	green and
tzngrmm	miauiauiauiauiauiauiauEN	cow
tzngrmm	katatatatatatatatER	cow and
tzngrmm	schnurrurrurrurrurrurrurrurrEN	sheep
grrrmmmmm	gänänänänänänänänSE	sheep and
schtzn	schnattattattattattattattattERN	green
schtzn	ziegiegiegiegiegiegiegiegEN	have been
t-t-t-t	meckeckeckeckeckeckeckeckERN	seen
…	…	

Poems in the contrasting class are also written by modern and postmodern poets, but they are based on syntactically regular sentences. In addition, the words of these poems are complete and correct, not decomposed as in the examples given above.

The basic aim of this paper is to identify the features of these techniques in concrete poetry, using the world’s largest corpus of recited poetry (*lyrikline*). In this work we developed a method to identify poetic features that are related to the delineation of rhythmical patterns in concrete poetry. This method is compared with an approach based on neural networks for the classification of concrete poetry and the contrasting class of normal poetry. The paper is organized as follows: Section 2 provides an overview of the database. The processing tools, and feature engineering (rule-based) as well as neural networks (NNs) based approaches are described in Section 3. The experimental results are given in Section 4. Finally, conclusions and future works are presented in Section 5.

2 Data

We used data from our partner *lyrikline* (http://www.lyrikline.org) in the project *Rhythmicalizer* (http://www.rhythmicalizer.net). *Lyrikline* was initiated by the Literaturwerkstatt Berlin and houses contemporary international poetry as texts (original versions and translations) and the corresponding audio files. All the poems are read by the original authors. There are 232 German-speaking poets (from Germany, Switzerland, and Austria) reading 2,571 German poems out of a total of 1,346 poets and 12,077 poems on *lyrikline*.

The philologist on our project (the second author) listened to the audio recordings of poems and classified them as belonging to to one of the three rhythmical patterns of concrete poetry (syllabic decomposition, lettristic decomposition, or permutation) or to the contrasting class. The amount of material examined in this work is small. There are a total of 133 poems (68 poems in the first group, “concrete poetry”, and 65 poems in the second group, the “contrasting class” or “normal poetry”). The number of poetic lines in the concrete poetry and the contrasting class is 1,913 and 2,090, respectively. The rhythmical patterns (syllabic decomposition, lettristic decomposition, and permutation) of concrete poetry are found in 21, 17, and 30 poems as well as 422, 612, and 879 poetic lines, respectively.

3 Method

Two approaches have been developed for the task of classification. Traditional feature extraction and classification with machine learning algorithms are employed in the first approach. The second approach uses a neural network that encodes the poem into a multi-dimensional representation.

3.1 Processing tools

The following tools are utilized for the analysis and feature extraction:

- **Text-Speech Aligner**: We perform forced-alignment of text and speech for poems using a text-speech aligner (Baumann et al. 2018b) which employs a variation of the SailAlign algorithm (Katsamanis et al. 2011) implemented via Sphinx-4 (Walker et al. 2004). The line boundaries (the start of the first word and the end of the last word in each line) are detected. The forced alignment of text and audio in spoken poetry, especially in concrete poetry, is non-trivial and often individual words or lines cannot be aligned. Therefore, the automatically extracted alignment information is manually corrected by the first author more than once (rectifying alignment information as well as in some cases correcting the written text of poems and the audio file).
- **Parser**: We processed the text data of poems by using a statistical parser in order to extract syntactic features. The Stanford parser (Rafferty–Manning

2008) is used to parse the written text of poems. The parser used the Stuttgart-Tübingen-TagSet (STTS) table developed at the Institute for Natural Language Processing of the University of Stuttgart (Schiller et al. 1999) for the parsing of German poems. The main problems in poem parsing (Hussein et al. 2019) involve the absence of punctuation marks. The data in this experiment contain 44 poems in concrete poetry (syllabic decomposition: 14, lettristic decomposition: 14, and permutation: 16) as well as 2 poems in the contrasting class without sentence endings. In addition, many poems are written with special characters: sometimes the text is written in lowercase with some words in uppercase, which makes the recognition of sentence boundaries by using the parser quite difficult. Furthermore, some sentences within the poems comprising the contrasting class go beyond the line boundary and run on to the next line. Such unconnected syntactic elements result from the dissolution of poetic lines, caused by so-called enjambment.

3.2 Feature engineering-based approach

We processed every poem individually, line by line, even if there are run-on lines (enjambments) within a poem, in order to extract features for the recognition of poems in the concrete poetry class. The most important indicators for concrete poetry are the absence of a verb within a complete sentence or half-sentence and the existence of asemantic material. We used parser information that comprises abbreviations of words' Part-of-Speech (PoS). Different features are extracted. We focused on the following inflected verbs: finite verbs (VVFIN), imperative verbs (VVIMP), auxiliary verbs (VAFIN), auxiliary imperative verbs (VAIMP), and finite modal verbs (VMFIN). We identified the punctuation marks in order to differentiate between concrete poetry and the contrasting class, because complete sentences in lines can be discerned by sentence-ending punctuation (. ? ! ; :), and clauses by commas. Therefore, we found all the punctuation marks in every poetic line. We also identified the following types of nouns: normal noun (NN) and proper name (NE). Two types of conjunctions are distinguished: subordinate conjunction in a sentence (KOUS) and coordinating conjunction (KON). Foreign language material (FM) as well as non-words (XY) are categorized as asemantic material. However, parsers cannot yet distinguish between nominative and accusative, so the most important indicator for a complete sentence is the verb. The features are recorded as follows: If the poetic line contains one or more verbs, a value of one is added to the feature vector; otherwise a value of zero is added. The same process is implemented in every poetic line for noun, comma, sentence-ending punctuation, conjunction, and foreign language material as well as non-words. Four sets of features sets are used in the analysis:

- **A** (2 features): verb and sentence-ending punctuation.
- **B** (3 features): verb, comma, and sentence-ending punctuation.
- **C** (5 features): verb, noun, comma, sentence-ending punctuation, and conjunction.

- **D** (6 features): verb, noun, comma, sentence-ending punctuation, conjunction, and asemantic material.

Several machine-learning algorithms are selected from the WEKA data mining toolkit (Hall et al. 2009) in the classification process:

- **IBk**: the Instance-Based (IB) classifier with a number of (k) neighbors is the K-nearest neighbours (KNN) classifier, using the euclidean distance and 1-nearest neighbour (Aha et al. 1991);
- **LogitBoost**: This classifier performs additive logistic regression (Friedman 1998 et al.);
- **RandomForest**: The classifier of random forest consists of several uncorrelated decision trees (Breiman 2001);
- **J48**: The J48 algorithm used to generate a pruned or unpruned decision tree (Quinlan 1993).

3.3 Neural networks-based approach

The approach based on neural networks for classification of prosodic styles is described in (Baumann et al. 2018a; Baumann et al. 2018c). The model must deal effectively with *data sparsity*, since there are a broad variety and a relatively small number of poems in the experiment. Therefore, we use as few free parameters as possible that need to be optimized during training. For this reason, in textual processing we focused on character-by-character encoding of poetic lines (and using character embedding). The textual information, the spoken recitation on the line level and the information regarding pauses between lines are utilized. We use a bidirectional recurrent neural network (RNN, using gated recurrent unit (GRU) cells (Cho et al. 2014)) which encodes the sequence of characters into a multi-dimensional representation that is trained to be optimal towards differentiating the prosodic classes. Pre-training with additional data from the German Text Archive (Geyken et al. 2011) is implemented. The model is not trained using an explicit notion of words. Instead, it may implicitly encode word-level information (such as PoS) via the constituting sequences of characters. This is in line with recent work on end-to-end learning, for example, in speech recognition (Hannun et al. 2014; Graves–Jaitly 2014), which no longer explicitly models phonemes or words, but directly transfers audio features to character streams. While processing on the word level might allow our model to build a better higher-level understanding of the poem's meaning, this semantic information would likely not help in style differentiation. In addition, word representations would not capture the usage of whitespace—for example, indentation to create justified paragraphs— nor special characters. We combine the line-by-line representations using a poem-level encoder which is fed to a decision layer and a final softmax to determine the poem's class, yielding the hierarchical attention network as shown in FIG. 1.

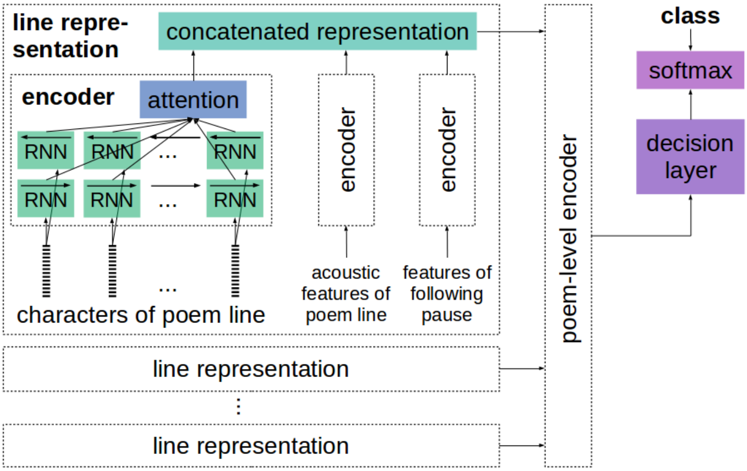

FIG. 1: Full model for poetry style detection using neural networks

4 Experimental results

Classification performance is measured with the weighted F-measure, which is the harmonic mean between precision and recall. The classification results in a 10-fold cross-validation are presented in TAB. 1. The following categories are classified together: syllabic decomposition versus lettristic decomposition versus permutation, decomposition versus permutation, permutation versus contrasting class, and concrete poetry versus contrasting class. The four classifiers used in the feature-based approach yielded mostly the same classification results for each feature vector. Therefore, there is no need to write the results of four classifiers for each feature vector. It can be seen that the increase in the number of features in the feature engineering-based approach yielded better results for the four classification pairs (the best results are provided by the feature vector (D)). The classification results of the neural networks-based approach show that the most valuable information seems to be in speech (except for the classification of permutation versus contrast class), whereas the information regarding pauses does not play an important role. The results in the table indicates that the neural network approach based on information contained in the text and audio of poems is more successful than the traditional feature-engineering approach. The weighted F-measure from the NN-based approach for the classification of decomposition versus permutation as well as of concrete poetry versus contrast class is 0.97 and 0.96, respectively. This indicates that the difference between concrete poetry and “normal” poetry can be detected by using a computational approach, and furthermore, that within concrete poetry the difference between permutation techniques and decomposition techniques are the best ones to detect automatically. We can explain this difference by the very fact that decomposition seems to be more radical in terms of its deviation from regular language than permutation.

	feature engineering & classifier				representation learning & NNs		
	A	B	C	D	text-only	text+speech	text+speech+pause
Syllabic vs. lettristic dec. vs. permutation	0.50	0.52	0.56	0.70	0.76	0.87	0.83
Decomposition vs. permutation	0.71	0.73	0.75	0.79	0.85	0.97	0.95
Permutation vs. contrasting class	0.56	0.65	0.68	0.68	0.74	0.68	0.70
Concrete poetry vs. contrasting class	0.62	0.70	0.78	0.79	0.85	0.96	0.96

TAB. 1: Classification results (weighted F-measure) using feature- and NNs-based approaches

5 Conclusion and future works

We present two approaches for the identification of concrete poetry in modern and postmodern free verse poetry by analyzing the *lyrikline* corpus, which is the largest corpus of spoken poetry. The first approach is based on the extraction of various features as defined in literary theory. The features are derived from a parser (based on text data only) focusing on syntactical units such as verbs, nouns, commas, sentence endings, conjunctions, and asemantic material. These features are extracted in order to measure the influence of various modeling parameters on the classification process. The second approach is based on hierarchical neural networks, using textual information and the spoken recitation of poetic lines as well as the information regarding pauses between lines. Both approaches are used to distinguish between concrete poetry and rather regular poems that use complete and correct sentences. The neural networks-based approach yielded the best results for classification of concrete poetry with the contrasting class (weighted F-measure of 0.96).

The difference in results between the first approach with all features considered (D) and the second approach with text-only features is small. This indicates that an attempt to improve the classification results can be made by integrating parser features into the neural networks approach. A further step would now be to identify in a similar manner syntactical features within modern poetry—for example, the difference between paratactical and hypotatctical line structures. Paratactical lines can be found in the famous expressionistic “Reihungsstil”; hypotactical lines can be found for example in the sonnets of Rainer Maria Rilke. Would it be possible to detect the difference between parataxis and hypotaxis in poems by using a computational approach as well?

Acknowledgements

This work has been funded by the Volkswagen Foundation through the program ‘Mixed Methods in the humanities? Funding possibilities for the combination and the interaction of qualitative hermeneutical and digital methods’ (funding code 91926 and 93255).

References

Aha, D. – Kibler, D. – Albert, M. (1991). Instance-Based Learning Algorithms. *Machine Learning* 6, 37–66.

Baumann, T. – Hussein, H. – Meyer-Sickendiek, B. (2018a). Style detection for free verse poetry from text and speech. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING 2018)*, Santa Fe, New-Mexico, USA.

Baumann, T. – Köhn, A. – Hennig, F. (2018b). The Spoken Wikipedia Corpus Collection: Harvesting, Alignment and an Application to Hyperlistening. *Language Resources and Evaluation*, January, 1–27.

Baumann, T. – Hussein, H. – Meyer-Sickendiek, B. (2018c). Analysing the Focus of a Hierarchical Attention Network: the Importance of Enjambments When Classifying Post-modern Poetry. In *Proc. Interspeech 2018*, 2162–2166.

Breiman, L. (2001). Random Forests. *Machine Learning* 45(1), 5–32.

Cho, K. – van Merrienboer, B. – Gulcehre, C. – Bahdanau, D. – Bougares, F. – Schwenk, H. – Bengio, Y. (2014). Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (1724–1734), Doha: Association for Computational Linguistics.

Emanuely, A. (2013). Das Gedicht à la lettre: Isidore Isou und der Lettrismus als Literatur des Widerstandes. *Zwischenwelt* 30(1), 37–41.

Ernst, U. (1992). Permutation als Prinzip in der Lyrik. *Poetica* 24(3/4), 225–269.

Friedman, J. – Hastie, T. – Tibshirani, R. (1998). Additive Logistic Regression: a Statistical View of Boosting. Technical report, Stanford University.

Geyken, A. – Haaf, S. – Jurish, B. – Schulz, M. – Steinmann, J. – Thomas, C. – Wiegand, F. (2011). Das deutsche textarchiv: Vom historischen korpus zum aktiven archiv. *Digitale Wissenschaft*, 157.

Gomringer, E. (1969). *Worte sind Schatten: die Konstellationen 1951-1968*. Rowohlt.

Graves, A. and Jaitly, N. (2014). Towards end-to-end speech recognition with recurrent neural networks. In *International Conference on Machine Learning*, 1764–1772.

Hall, M. – Frank, E. – Holmes, G. – Pfahringer, B. – Reutemann, P. – Witten, I. H. (2009). The WEKA Data Mining Software: An Update. *SIGKDD Explorations* 11(1), 10–18.

Hannun, A. – Case, C. – Casper, J. – Catanzaro, B. – Diamos, G. – Elsen, E. – Prenger, R. – Satheesh, S. – Sengupta, S. – Coates A. et al. (2014). Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*.

Hartung, H. (1975). *Experimentelle Literatur und Konkrete Poesie*. Göttingen: Vandenhoeck und Ruprecht.

Hussein, H. – Meyer-Sickendiek, B. – Baumann, T. (2019). How to Identify Elliptical Poems Within a Digital Corpus of Auditory Poetry. In *Proceedings of the 30th Conference on Electronic Speech Signal Processing* (ESSV 2019) (247–254), Dresden, 6–8 March 2019.

Jandl, E. (1966). Laut und Luise. Nachwort von Helmut Heißenbüttel. Olten and Freiburg: Walter-Verlag

Katsamanis, A. – Black, M. – Georgiou, P. G. – Goldstein, L. – Narayanan, S. (2011). SailAlign: Robust Long Speech-Text Alignment. In *Proc. of Workshop on New Tools and Methods for Very-Large Scale Research in Phonetic Science.* University of Pennsylvania, Philadelphia, USA, January.

Mittelmeier, M. (2016). *Dada. Eine Jahrhundertgeschichte*. München: Siedler.

Mon, F. (2012). Lettern Laute Silben Wörter: Anfänge der Konkreten und Visuellen Literatur im deutschsprachigen Raum. In *Poesie – konkret* (187–195), Köln.

Quinlan, R. (1993). *C4.5: Programs for Machine Learning*. San Mateo (CA): Morgan Kaufmann Publishers.

Rafferty, A. N. – Manning, C. D. (2008). Parsing Three German Treebanks: Lexicalized and Unlexicalized Baselines. In *Proceedings of the Workshop on Parsing German, PaGe '08* (40–46), Stroudsburg (PA): Association for Computational Linguistics.

Schiller, A. – Teufel, T. – Stöcker, C. – Thielen, C. (1999). Guidelines für das Tagging deutscher Textcorpora mit STTS (Kleines und größes Tagset). Available on http://www.ims.uni-stuttgart.de/forschung/ressourcen/lexika/TagSets/stts-1999.pdf. Last accessed at 25. January 2019.

Walker, W. – Lamere, P. – Kwok, P. – Raj, B. – Singh, R. – Gouvea, E. – Wolf, P. – Woelfel, J. (2004). Sphinx-4: A Flexible Open Source Framework for Speech Recognition. Technical report, Mountain View (CA), November.