A Tool for Human-in-the-Loop Analysis and Exploration of (not only) Prosodic Classifications for Post-modern Poetry

Timo Baumann[1] Hussein Hussein[2] Burkhard Meyer-Sickendiek[3] Jasper Elbeshausen[4]

Abstract: Data-based analyses are becoming more and more common in the Digital Humanities and tools are needed that focus human efforts on the most interesting and important aspects of exploration, analysis and annotation by using *active machine learning* techniques. We present our ongoing work on a tool that supports classification tasks for spoken documents (in our case: read-out post-modern poetry) using a neural networks-based classification backend and a web-based exploration and classification environment.

Keywords: data mining; human-in-the-loop; classification; free verse poetry; rhythmical patterns

1 Introduction

The widespread availability of spoken documents, and the ever more frequent availability of approximate textual alignments for such documents (e.g. for the Spoken Wikipedia [BKH18], via Youtube transcription, or, as in our case, for author-read post-modern poetry [Be19]) enables philologists to investigate this spoken material using data-mining and exploration methods, e.g. classifying material into different categories, finding outliers, visualizing differences, and so on. One such example is the classification and analysis of contemporary audio poems regarding their rhythmic features, especially since the end of classical metrics and the overcoming of rhyme. Research in this field of the so-called free verse prosody is existing since the 1980s, but till today this theory developed in the USA is still totally unknown in Europe. Unfortunately, cause this theory enables students, scholars and philologists to identify very precisely different rhythmic forms in modern and postmodern poetry. Poets such as Ezra Pound or Gottfried Benn, for example, have developed rhythmic forms which had a huge impact on the Poetry written from the 1950s up to the 1970s, the same also applies to the very influential Black Mountain Poets, Beat Poets or Concrete Poetry. Even the poetic features of Poetesses or Poetry from the German Democratic Republic (GDR) can be recognized and classified stylistically by such rhythmic types explained in free verse prosody discourse.

Active learning is a case of semi-supervised machine learning in which a user is allowed to choose the correct instances of unlabeled training data interactively for the learning

[1] Department of Informatics, Universität Hamburg, Germany, baumann@informatik.uni-hamburg.de
[2] Department of Literary Studies, Free University of Berlin, Berlin, Germany, hussein@zedat.fu-berlin.de
[3] Department of Literary Studies, Free University of Berlin, Berlin, Germany, bumesi@zedat.fu-berlin.de
[4] Department of Informatics, Universität Hamburg, Germany, 2elbesha@informatik.uni-hamburg.de

algorithm [Se09]. The *prodigy* is an existing annotation system based on active machine learning for the classification of text and image [Ma19]. *Prodigy* combines the state-of-the-art techniques of machine learning and user experience. The user asked to annotate examples which are not known to the model. The annotation is based on the web application which is extensible and designed to help the user to focus on one decision at a time by clicking on the correct answer.

In this paper, we present a *rapid prototyping* approach for such a corpus-based classification of spoken poems. We combine a web-based tool that enables the literary scholar (philologists and students alike) to explore existing (manual and automatic) classifications of poems and allows the annotation of poetic classes to poems, and the correction of automatic classifications with a backend that performs classifications using the manual annotations based on machine learning via neural networks. Our neural network has been trained with annotated postmodern poems for the classification task of modern and postmodern poems [BHMS18]. The main contribution of this paper is the integration of a *human-in-the-loop* via a web-based interface. This interface itself is agnostic as to the *types* of classifications that are performed, that is, it can also be used to build one's own class hierarchies and to explore possible classification schemes of literary data, for example in teaching environments.

2 Rapid Prototyping of Classifications

Explorative analyses as well as the data-based validation of a classification theory profit greatly from an iterative approach to annotation and validation: starting from a few annotations, a model can be built, this model be applied to yet un-annotated items and the outputs from the model can be validated (or rejected) by further annotations. When considering model confidence, it may be useful to primarily consider automatic classifications that are particularly inconfident (so as to help the model as much as possible) or those that are particularly confident (so as to make annotation as easy as possible). Our tool is to support both modes of operation. Finally, it may help to show the confidence of the model, as well as the contending classes. In our use-case, the classification starts with annotations that already exist for a number of poetic classes and the purpose of the tool is to gradually check and possibly extend this classification; this would hold even more for use-cases which do not start from a pre-existing classification. Thus, the tool needs to support a view of all the classes (and assigned poems) as well as a view that shows the individual poems as well as their (automatic) classification.

As more insight was gained about the corpus, it was useful to split (or merge) classes and to build a class hierarchy that subdivided the classes according to a dis-fluency spectrum (see Figure 1) to account for more fine-grained analyses over time. This spectrum of (dis-)fluency is based on the most important theory within the free verse prosody discourse: The idea of grammetrical ranking, developed by Donald Wesling.

The exploration process should be tracked, keeping the history of manual annotations and the corresponding class hierarchies. This history allows one to 'go back' and greatly simplifies testing out classification ideas. Given the potential to try out the feasibility and usefulness of different classifications (e.g. in our domain: attempting to classify poems or parts thereof regarding the degree of intertextuality, syntactic dissociation, paratactic or hypotactic sentences), it is of course necessary to store annotations with different purposes or from different users separately. Eventually, we would hope for our backend classifier to jointly learn from the combination of these classes, using multi-task learning.

3 Tool for Human-in-the-Loop Classification

The main focus of our tool, as developed in the previous section, is on the more efficient integration of the human-in-the-loop, as well as allowing explorative use-cases with little effort. For this, we develop a web interface which uses the results of an automatic classifier and makes as much information about the decision making process available as is possible, including the model's *confidence* (the trust of the model in its judgment) and the model's *attention* (places in the text and speech that the model deems particularly relevant for the judgment). Using this interface, philologists should be able to explore the classification results and potentially correct or enhance them. We also allow the philologist to change the existing manual class-assignments of a poem, for example splitting a class into multiple sub-classes, as the exploration of the corpus proceeds.

3.1 Web-based Frontend

An initial run of the neuronal network-based method after the training phase provides classification results of unknown poems. These are sorted according to the confidence values. A web interface is created, into which these results are fed. This web interface allows the user to display and correct either classifications of the network with high or low confidence. These corrections are analyzed using cost-benefit calculations to identify any significant differences.

The Interface is based on React (JavaScript library for creating user interfaces) and utilize the Javascript XML (Extensible Markup Language) or Javascript Syntax Extension (JSX) for fast modification of all elements of a HTML document with the HTML DOM (Document Object Model). The modularity of React components enables us to keep the single parts of the interface separate from each other. The making of further additions in the future is easily implementable. So far, the graphical user interface (GUI) of our tool is shown in Figure 1. The web-based frontend of the software consists of three parts: the poem-level classification module with the neural networks model-output on the left, the poem itself in the middle, and the interactive classification correction on the right whereas the retraining of the classifier can be triggered after every correction or only after an explicit command for retraining. The poem text itself is meant to be playable line-wise using timing information extracted from the *lyrikline* data.

In the classification part on the left, we presently implement a graphical visualization to show the distribution of the confidence values over the different classes. Our tool is also presently extended with a category view, where the user is able to sort through the available corpus or create new categories. The user can also look through the classification results of poems and jump into any single poem classification as she chooses.

The data used in the interface comes from different sources. We have the output of our neural network – giving us the confidence distribution – which is loaded to the website, i.e. our interface, as a JavaScript Object Notation (JSON). An additional JSON is gathered through Python scripts and provided us with the necessary meta information (like the poem itself, timing information, author, etc.) obtained from *lyrikline* website. All these information are internally saved as an array so that we are able to quickly search for different confidence values or different categories. However, the output is saved in different arrays and ultimately exported as a third JSON, so that the initial values of the model are not lost but rather can be used to test for the evaluation of different approaches to the philological correction of the data.

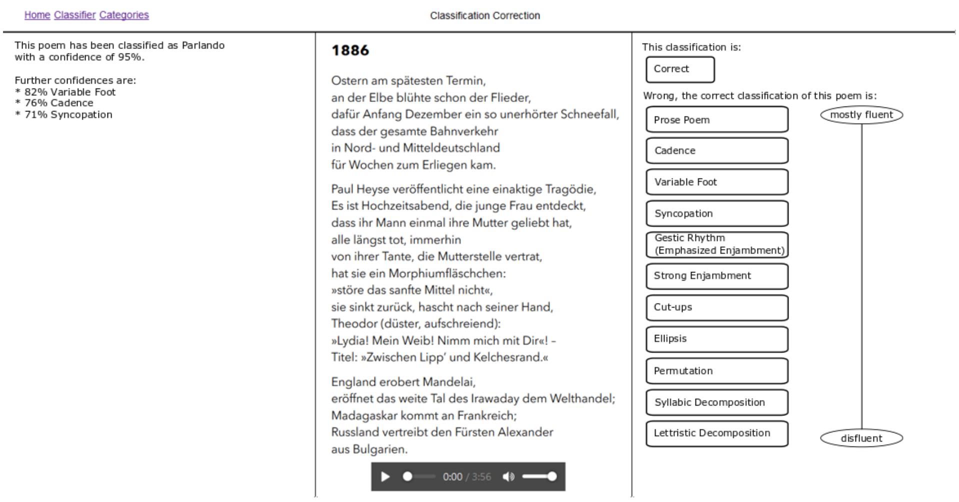

Fig. 1: The graphical user interface of the tool.

3.2 Automatic Classification

The aforementioned spectrum of (dis-)fluency is based on the most important theory within the free verse prosody discourse: The idea of grammetrical ranking, developed by Donald Wesling. Wesling's neologism “grammetrics” is a hybridization of grammar and metrics, based on the key hypothesis that in poetry as a kind of versified language, the grammatical units (sentence, clause, group, word, morpheme) and the metrical units (syllable, foot, part-line, line, rhymated pair, stanza, whole poem) interact in a way for which Wesling finds 'scissoring' an apt metaphor. The grammetrical raking assumes that meter and grammar can be scissored across each other [We96, p. 67].

Multimodal classifiers are used by [LGH08] to classify music mood with significantly better results than non-multimodal systems. The hierarchical document classification using an attention model is applied by [Ya16] with significantly better results than previous *n-gram* based classifiers. The existing model within the project is a hierarchical neural network with so-called gated recurrent units (GRU). It examines the source files, i.e. texts and spoken poems, on both line and poem level [BHMS18]. The existing model is able to differ and to classify poems existing on *lyrikline* into 12 categories based on a fluency-disfluency ranking [MSHB19a]: Prose Poem, Cadence, Parlando, Variable Foot, Syncopation, Gestic Rhythm (Emphasized Enjambment), Strong Enjambment, Cut-ups, Ellipsis, Permutation, Syllabic Decomposition, and Lettristic Decomposition.

3.3 Data

We used data from our partner *lyrikline* [Be19] in the project *Rhythmicalizer* [MSHB19b]. *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 233 German-speaking poets (from Germany, Switzerland, and Austria) reading 2, 581 German poems out of a total of 1, 365 poets and 12, 239 poems on *lyrikline*.

4 Conclusion and Future Works

We have presented our ongoing work on a tool for a human-in-the-loop classification, annotation, exploration and analysis of spoken corpora. We use this tool for the analysis of the *lyrikline* corpus of spoken post-modern poetry. Yet our tool is not limited to this domain and could easily be used for document collections with other types of spoken material in which classification is of relevance.

In the future, we would like to extend our tool so that the reasons for classification can be annotated, e.g. on a line-by-line level or even more fine-grained. Such information is highly relevant for the study of the poetic classification (why is a poem classified as a particular class rather than an other) and this could also be used by the backend, e.g. via the attention mechanism, leading to a sort of supervised attention mechanism.

Finally, we expect that many different users could create their own classifications and subclassifications, potentially leading to many different classification hierarchies. While our tool must support a personalization of classifications for this use-case, this information could still be integrated across users in a multi-task learning environment, in which the backend would attempt to re-create each user's classifications (that may well bear different meanings) based on a joint model across all user's classification.

5 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).

Bibliography

[Be19] Berlin, Literaturwerkstatt: , Lyrikline: listen to the poet. Available on www.lyrikline.org, 2019. Last accessed at 30. April 2019.

[BHMS18] Baumann, T.; Hussein, H.; Meyer-Sickendiek, B.: 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, 2018.

[BKH18] Baumann, Timo; Köhn, Arne; Hennig, Felix: The Spoken Wikipedia Corpus Collection: Harvesting, Alignment and an Application to Hyperlistening. Language Resources and Evaluation, 2018. Special Issue representing significant contributions of LREC 2016.

[LGH08] Laurier, C.; Grivolla, J.; Herrera, P.: Multimodal Music Mood Classification Using Audio and Lyrics. In: 2008 Seventh International Conference on Machine Learning and Applications (ICMLA’08). pp. 688–693, December 2008.

[Ma19] MakersOfspaCy: , prodigy: Radically efficient machine teaching. An annotation tool powered by active learning. Available on https://prodi.gy/, 2019. Last accessed at 05. June 2019.

[MSHB19a] Meyer-Sickendiek, Burkhard; Hussein, Hussein; Baumann, Timo: From Fluency to Disfluency: Ranking Prosodic Features of Poetry by Using Neural Networks. In: Digital Humanities 2019, DH 2019, Utrecht, Netherlands, Juli 09-12, 2019. 2019.

[MSHB19b] Meyer-Sickendiek, Burkhard; Hussein, Hussein; Baumann, Timo: , Rhythmicalizer. A Digital Tool to Identify Free Verse Prosody. Available on www.rhythmicalizer.net, 2019. Last accessed at 30. April 2019.

[Se09] Settles, Burr: Active Learning Literature Survey. Computer sciences technical report, University of Wisconsin–Madison, 2009.

[We96] Wesling, D.: The Scissors of Meter: Grammetrics and Reading. University of Michigan Press, 1996.

[Ya16] Yang, Zichao; Yang, Diyi; Dyer, Chris; He, Xiaodong; Smola, Alex; Hovy, Eduard: Hierarchical attention networks for document classification. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 1480–1489, 2016.