ESCRITO – An NLP-Enhanced Educational Scoring Toolkit

Torsten Zesch and Andrea Horbach
Language Technology Lab, Department of Computer Science and Applied Cognitive Science
University of Duisburg-Essen, Germany
{torsten.zesch, andrea.horbach}@uni-due.de

Abstract

We propose ESCRITO, a toolkit for scoring student writings using NLP techniques that addresses two main user groups: teachers and NLP researchers. Teachers can use a high-level API in the teacher mode to assemble scoring pipelines easily. NLP researchers can use the developer mode to access a low-level API, which not only makes available a number of pre-implemented components, but also allows the user to integrate their own readers, preprocessing components, or feature extractors. In this way, the toolkit provides a ready-made testbed for applying the latest developments from NLP areas like text similarity, paraphrase detection, textual entailment, and argument mining within the highly challenging task of educational scoring and feedback. At the same time, it allows teachers to apply cutting-edge technology in the classroom.

Keywords: automatic assessment, text classification und software toolkit

1. Introduction

Scoring student writings is a core task for teachers, which requires a lot of manual effort. Using assisted or automated scoring might have a tremendous impact on the quality of teaching, as it potentially shifts the focus from tedious assessment tasks to communicating knowledge. In the case of free-form student writings, like essays or answers to factual questions, correctly assessing a response is still a challenging task. There are typically many ways how a correct response can be expressed, and the variety of responses is increased even more by orthographic and grammatical deviations, which occur frequently in student writings. Therefore, automatically scoring student writings usually involves various other tasks central to NLP such as spell checking, grammatical error correction, POS tagging, paraphrase recognition, textual entailment, or argument mining.

In recent years, attempts have been made to link free-form scoring to some of the above-mentioned fields. For example, the SemEval 2013 Student Response Analysis Task (Dzikovska et al., 2013) combined short-answer scoring with recognizing textual entailment. More fundamental research along such lines would be desirable. In addition to basic research, more practical experimentation in the classroom is needed as well. These goals require the involvement of two disjunct groups of people: *NLP researchers* (who develop scoring methods) and *teachers* (who bring the methods to the classroom). However, existing implementations are often very specific to certain use cases and datasets. That makes it difficult for teachers without a technical background to use them out-of-the-box, let alone apply them to new data. At the same time, the application field is highly complex, which discourages NLP researchers from testing the latest developments in this field.

We thus present ESCRITO, the Educational SCoRIng TOolkit, a toolbox for free-text scoring based on natural language processing and machine learning, which caters to both user groups. ESCRITO has two main goals: (i) to enable teachers to quickly build free-text scoring systems and apply them in real-life scenarios, and (ii) to provide an application testbed for the integration and evaluation of NLP algorithms. Teachers can access ESCRITO using a high-level API that allows them to specify and execute scoring pipelines on their own data following best-practices in the field. Research scientists will find a low-level API, which allows them to access, customize, and extend all relevant aspects of automatic scoring including preprocessing, feature extraction, and machine learning setup.

We ensure reproducibility of results through detailed automated documentation of experimental setups. We also have designed ESCRITO to be as language-independent as possible. It has been successfully applied to data in various languages. All parts of ESCRITO have already been used in research projects concerning essay scoring (Zesch et al., 2015b; Horbach et al., 2017c), spellchecking on learner data (Horbach et al., 2017a), clustering (Zesch et al., 2015a), and neural short-answer scoring (Riordan et al., 2017). This shows the wide applicability of the framework and that state-of-the-art approaches can be easily modeled within the framework.

Related Work To the best of our knowledge, there are no other publicly available general-purpose scoring frameworks addressing either programmers or practitioners. Proprietary systems such as e-rater (Attali and Burstein, 2004) can only be used commercially and as a sort of black-box. While a number of scoring implementations are publicly available, such as Neural Essay Assessor (Taghipour and Ng, 2016), an essay scoring system for Swedish (Östling et al., 2013), or clustering-based scoring Zesch et al. (2015a), these implementations are typically centered around a specific dataset and method, and not straight-forward to extend or apply to new data. Equally, there are approaches for supporting teachers with free-text answers in MOOCs. An example is a plugin for the learning management system Moodle which sorts answers by their similarity to a reference answer (Pado and Kiefer, 2015).

2. Educational Scoring Toolkit

Educational free-text scoring is often tackled as a classical supervised learning task with the goal to assign a label to some piece of text written by a learner in response

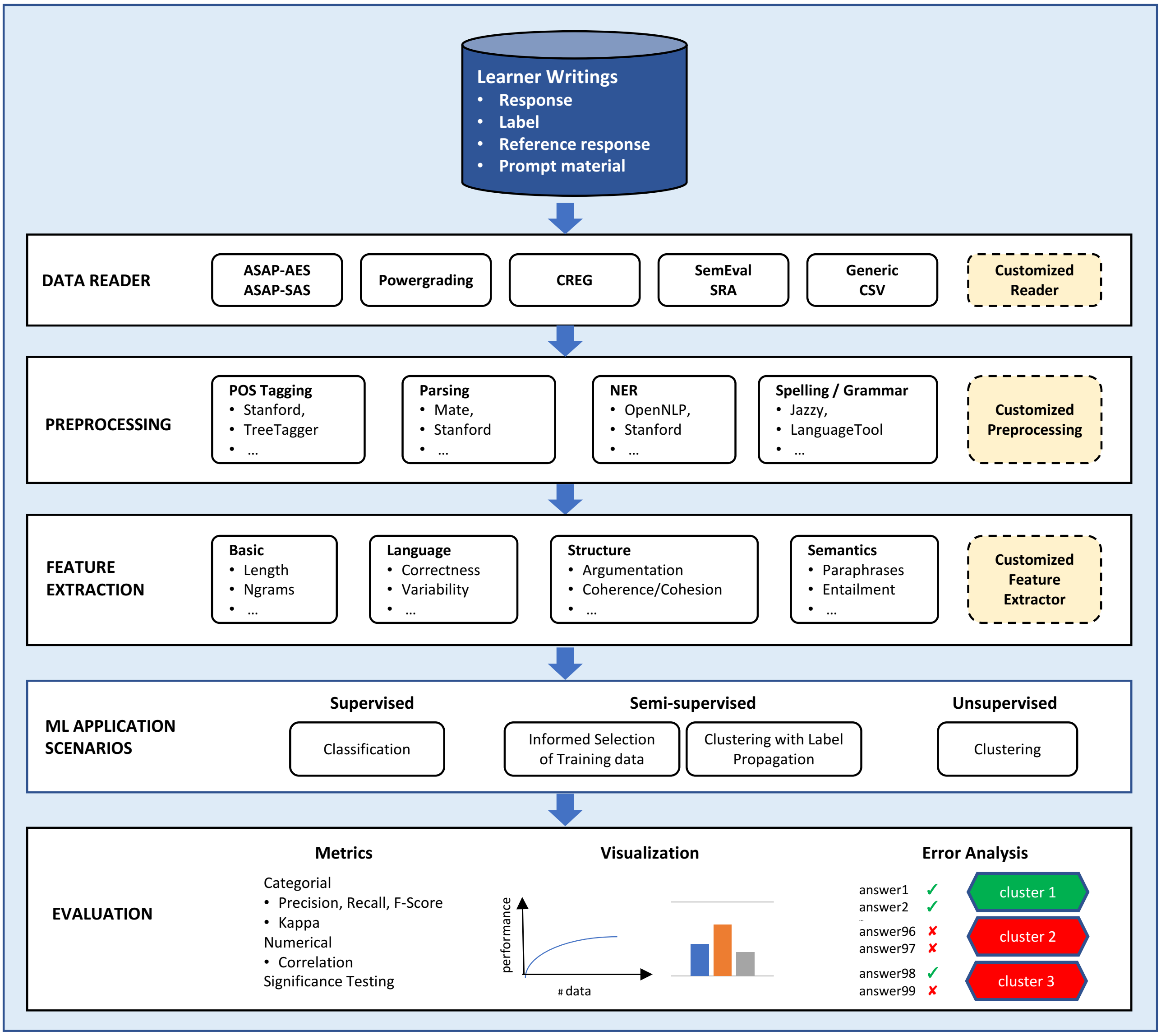

Figure 1: Overview of the ESCRITO architecture.

to a prompt. This text can either be an answer to a so-called short-answer question, asking for answers consisting of only a few words or a few sentences, or it can be a longer text, such as an essay consisting of several hundred words. Figure 1 shows examples for the range of free-text exercises and the variety of possible learner answers for individual prompts.

The labels assigned may be numeric as well as categorical (correct/incorrect or more fine-grained diagnostic labels). Their value can either be based on content alone (content-scoring) or on content and form (typically in holistic scores for essay scoring). Typically, the prompt in response to which answers are given is available. Predefined reference answers, i.e. sample solutions, are available in some cases. These additional materials can also be leveraged in automatic scoring; for example, to compare whether an answer to be scored is similar to the reference answer or to identify domain-specific vocabulary useful for spell-checking.

In order to model this complex setup, we build ESCRITO on top of DKPro TC (Daxenberger et al., 2014), an UIMA-based open-source framework that provides easy access to various algorithms for supervised text classification and extensive parameter documentation enabling reproducible research. ESCRITO extends DKPro TC with respect to the specific needs of educational scoring applications: it offers easy access to existing educational datasets, various preprocessing options, state-of-the-art scoring features, evaluation and visualizations for common scoring scenarios, as well as options to integrate new data and to customize existing or to add new preprocessing components and feature extractors. Figure 1 gives an overview of the system's architecture.

2.1. Easy Access to Existing Datasets

Datasets often come each in their own format, such that data preparation can be tedious and time-consuming. We provide pre-implemented readers for state-of-the-art datasets, such as ASAP-AES[1] for essay scoring and ASAP-

[1] https://www.kaggle.com/c/asap-aes

PHRASE-LENGTH LEARNER ANSWERS FOR CONTENT SCORING (POWERGRADING - PROMPT 1)
QUESTION: What is one right or freedom from the First Amendment of the U.S. Constitution?

LEARNER ANSWERS:

- **correct:** freedom of speech
- **correct:** free speech
- **correct:** freedom to talk freely
- **correct:** freedome of religeon
- **incorrect:** the right to bear arms
- **incorrect:** life

PARAGRAPH-LENGTH LEARNER ANSWERS FOR CONTENT SCORING (ASAP-2 - PROMPT 1)
QUESTION: After reading the groups procedure, describe what additional information you would need in order to replicate the experiment. Make sure to include at least three pieces of information.

LEARNER ANSWERS:

- **3 points:** Some additional information you will need are the material. You also need to know the size of the contaneir to measure how the acid rain effected it. You need to know how much vineager is used for each sample. Another thing that would help is to know how big the sample stones are by measureing the best possible way.
- **1 point:** After reading the expirement, I realized that the additional information you need to replicate the expireiment is one, the amant of vinegar you poured in each container, two, label the containers before you start yar expirement and three, write a conclusion to make sure yar results are accurate.
- **0 points:** The student should list what rock is better and what rock is the worse in the procedure.

TEXT-LENGTH LEARNER ANSWER FOR SCORING OF PERSUASIVE ESSAYS (ASAP-1 - PROMPT 1)
INSTRUCTION: Write a letter to your local newspaper in which you state your opinion on the effects computers have on people. Persuade the readers to agree with you.

LEARNER ANSWER:

- **6 points:** Dear, @ORGANIZATION1 I think the effects that computers do on people are really positive. Computers can be used for all sorts of things. Examples like finding things out about history. People that changed the world and other information. Computers give the power for children to learn. For example, their are lots of websites that offer online tutoring or good ways to help you pass school. Other positive way is online dateing sites. You can meet new people and is a good way to make life even better. Popular sites like @CAPS1, @CAPS2, @CAPS3, and so on make a good way to keep in touch with friends from your past, or even make new ones. But the most that I think thats the best in my opinion is going to school online. Once your done with colloge and you are a nuse, for an example you can get a higher degree like a registered nurse then being a @ORGANIZATION2. I think computers has a positive effect on people.

Table 1: Examples for free-text tasks asking for answers of very different complexity.

SAS[2] for short-answer scoring, the Powergrading short-answer dataset (Basu et al., 2013), the dataset by Mohler and Mihalcea (2009), the Student Response Analysis (SRA) dataset (Dzikovska et al., 2013), and the German CREG corpus (Meurers et al., 2011) to facilitate this task. Furthermore, we also support generic educational datasets, consisting of at least a set of learner answers, each with an ID and a score. We provide generic readers for such datasets in line-based formats (such as CSV) which novice users may use to integrate their own data without having to write their own reader. These readers support different properties of the data, such as categorical and numerical scoring labels. They also support datasets where a prompt text is available and both datasets without reference answers as well as datasets where one or several reference answers are provided and used for comparison with the learner answer. We additionally provide interfaces for expert users to easily integrate new dataset readers.

2.2. Preprocessing

For preprocessing, ESCRITO integrates all tools available through DKPro Core (Eckart de Castilho and Gurevych, 2014), which provides access to a large number of interchangeable and interoperable components in several languages. In the high-level mode, the system automatically checks which preprocessing components are needed for a selected language and feature setting. It assembles a standard preprocessing pipeline according to best practices, so that a teacher does not have to worry about the implementation details of linguistic analysis. In the expert mode, the

[2] `https://www.kaggle.com/c/asap-sas`

user can select from the complete range of DKPro Core preprocessing components and assemble a pipeline according to their needs while they get a warning if a feature extractor requires a preprocessing step that is not included by the user.

In ESCRITO, we add preprocessing components according to the special requirements of the task. Learner answers typically show a much higher orthographic variability than standard language. Spelling errors are typically ignored in content-scoring while their absence or presence is a useful feature for form-based scoring. Therefore, learner answers can optionally be normalized before feature extraction or language errors are simply annotated to be consumed later by a feature extractor that extracts the number and type of errors in an answer as features. Therefore, we add spell-checking components based on spellcheckers like Jazzy[3]. Our method (Horbach et al., 2017b) automatically extends their lexicon using prompt-specific material, e.g. the words in the prompt the answer refers to. Thus, lexical items that a learner likely referred to when creating their answers appear in the dictionary of a spell-checker even if they come from a very specific domain and would not appear in standard spelling dictionaries. Furthermore, we implement spell-checking mechanisms that prefer domain-vocabulary over non-domain-vocabulary when automatically correcting spelling errors.

In addition to spell-checking, we provide optional marking of elements in a learner answer as being copied from the question in the prompt. Consider the example question *Where was Peter born?* where the rheme-only answer *in Berlin* and the full answer *Peter was born in Berlin* convey the same content. Marking this material provides the option to ignore it later in some feature extractors.

Even more task-specific preprocessing components can easily be plugged in by wrapping them as UIMA annotators.

2.3. State-of-the-art Scoring Features

ESCRITO provides a wide variety of state-of-the-art features from both essay and content scoring as well as means for easily integrating newly developed ones.

N-gram features are known to be a powerful feature group determining the content of an answer. We extend the n-gram feature extractors provided through DKPro TC by integrating different normalization techniques and determining n-grams not only on the basis of words and characters, but also based on dependency triples. **Length features**, such as number of tokens or sentences are also known to have a highly predictive power, especially when answers are written under a time limit.

Several feature groups target the language of an answer in terms of complexity and correctness: **Linguistic Complexity** is especially important for essay scoring. We measure linguistic complexity through variance on the lexical level (type-token-ratio), on the syntactic level (distribution of POS tags, average and maximal depth of parse trees, number and type of subordinate clauses), and via a number of readability measures from DKPro TC readability. We extract **language correctness** features about the nature and frequency of different language errors identified by our own spell-checking methods as well as grammar and stylistic errors found by LanguageTool.[4]

We target the structure of an essay through **Coherence and Cohesion features** which measure the usage of connectives in an answer, as well as the content overlap between adjacent sentences. We also provide features on the **argumentative structure** of an essay, e.g. through the number and distribution of claims and citations in a text. Especially in short-answer scoring, prompts often include a target answer and answers are typically correct if they are similar to or entail the target answer. Based on DKPro Similarity (Bär et al., 2013), we provide **textual similarity** feature extractors that measure similarity between the learner and the target answer on the surface level (token overlap and string similarity measures), on the syntactic level (overlap of dependency triples), or the semantic level (e.g. using concept alignment).

In case that there are several possible reference answers for a learner answer, we offer different ways of combining the evidence from these reference answers by either taking the maximum, minimum, average, or all feature values produced when comparing a learner answer to the individual target answers. These differences can, for example, be important when handling both similarity scores as well as distances. New features can easily be implemented and integrated by using interfaces for features either based on the learner answer text alone or its comparison with a prompt text or reference answer.

Integration of Deep Learning Deep learning methods became widely used in various NLP areas including educational free-text scoring (Taghipour and Ng, 2016; Riordan et al., 2017) and are often a very powerful alternative to traditional shallow learning methods. DKPro TC has been extended (Horsmann and Zesch, 2018) to also provide interfaces to widely used deep learning frameworks including Keras (Chollet and others, 2015), DeepLearning4J [5], and Dynet (Neubig et al., 2017), while ensuring reproducibility and easy preprocessing through DKPro TC. We integrate this extension to make sure deep learning methods can be used in ESCRITO.

2.4. Machine Learning Scenarios

We specify commonly used experimental setups that allow for both supervised and unsupervised machine learning scenarios according to the needs of the two user groups.

From an NLP researcher's perspective, the supervised case with labeled train and test data is certainly the most common one. We provide setups for both cross-validation and train-test setups with the option to choose from different machine learning tools as provided by Weka (Hall et al., 2009) and wrapped through DKPro TC.

Additionally, learning curve evaluations come in handy when one wants to assess how many training instances are needed until no further improvement can be reached with more data. ESCRITO implements learning curves which

[3] https://github.com/reckart/jazzy

[4] https://www.languagetool.org/

[5] http://deeplearning4j.org

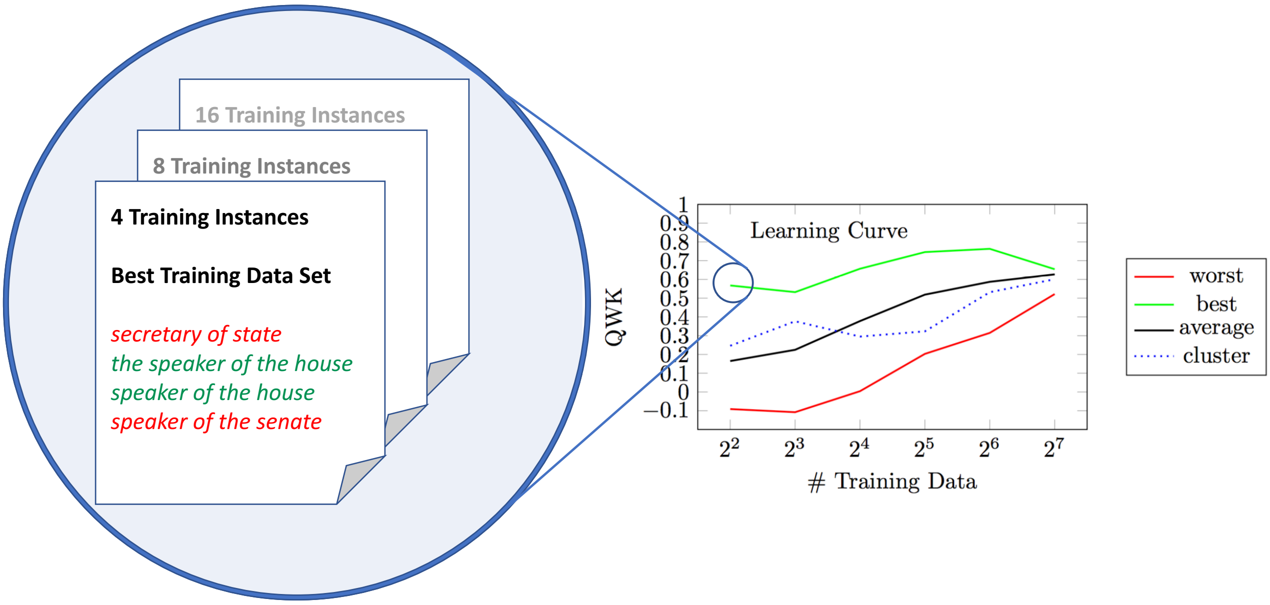

Figure 2: An example for the evaluation of a learning curve experiment.

simulate that only a limited number of training data is available and provides best, worst, and average learning curves across large numbers of randomly selected items.

Real-life scoring scenarios often fall somewhere in between the supervised and the unsupervised case: some part of the data is typically unlabeled and should be labeled by the tool (and afterwards potentially re-checked by a teacher). Therefore, we assume that scenarios where a model is to be trained on existing labeled training data and applied on new unlabeled test-data will occur frequently; for example, when the same or similar prompts have already been used and scored in an earlier exam and are now administered to a new cohort of students. In such a case, we can provide evaluations on the training data using cross-validation to allow a performance estimation and present classification results on the unlabeled test data with additional confidence scores. This allows a teacher to re-visit the automatically scored data manually, while concentrating on the uncertain cases and checking with higher preference.

When a teacher wants to score data from a new domain with no existing labeled training examples, two options are possible. First, in a completely unsupervised scenario, items are clustered according to the similarity between answers, such that clusters are formed that contain similar answers. A teacher can then inspect clusters and assign scoring labels either to whole clusters or to individual members of a cluster. In this way, they can save annotation time and effort and are at the same time informed about common misconceptions in the student answers (Basu et al., 2013). Second, a teacher might want to label some, but not all their data, train a classifier and then re-label the complete dataset (or only the so-far unlabeled data). In this scenario, we provide methods to select the items to be labeled in an informed way, for example by selecting items so that they cover as much of the feature space as possible. As an alternative, we provide methods from *active learning*, where items are dynamically selected in a way that a machine learner profits most from them, such that human annotation effort is reduced (Horbach and Palmer, 2016).

2.5. Evaluation and Visualization

We report frequently used evaluation metrics depending on the type of labels used in a dataset: accuracy, quadratically and linearly weighted kappa, precision, recall, F-measure as well as correlation scores such as Pearson(Pearson, 1895) and Spearman(Spearman, 1904). To facilitate error analysis, incorrectly classified items (i.e. false positives and false negatives for a class) are written to separate files and can easily be inspected. Besides plain text result files, ESCRITO also writes tables and figures in LaTeX formatting or images. Additionally, we provide mechanisms to compare different experiments and perform significance testing on them.

In the unsupervised case, scoring results per answer are presented to the user for manual inspection where the user can customize the data to be ordered, e.g. by scoring confidence of the system, by the assigned class, or clustered based on item similarity so that similar items can be reviewed together.

Figure 2 shows as an example part of the evaluation of a learning curve experiment. The user gets output in the form of a chart showing the performance of the best, worst and average training data configurations resulting from a large number of random samples of training data as well as the performance of selecting training data based on cluster centroids. The user can inspect for every point on the curve which particular item selection lead to that result.

3. Example Use Cases

We will now have a closer look how the toolkit can be used, which we exemplify based on two example users.

A **textual entailment researcher** who has developed a new, promising algorithm might want to evaluate it as part of a educational scoring pipeline. With ESCRITO, she can easily assemble a baseline system that is evaluated on multiple datasets and then compare the results against an augmented system, that uses entailment as an additional feature. She can even easily get access to the instances on which the two system configurations differed in order to examine the cases in which the entailment-based system outperformed (or underperformed) the baseline.

An *educational science researcher* wants wants to know whether automatic scoring in a particular test discriminates against non-native speaking students, for example because they might use a wording not so frequently used in the training data. To do so, he can easily train and cross-validate scoring models for his test with a set of annotated answers. By inspecting the automatic scores in comparison to the human annotations (for example in the form of files with lists of false postive and false negative answers), he can easily determine whether answers from non-native speakers are more likely to be misclassified by a certain algorithmic setup.

A **teacher**, who wants to score student writings in Italian as a consistency check in addition to his own manual scoring, can use on of the pre-configured setups to train a model and apply it to a new cohort. The data only needs to be formatted in the default format of one response per line with the label separated by a tab. ESCRITO automatically selects the default preprocessing pipeline for Italian and the teacher can easily inspect the resulting classifications. If he decides that it would be better to ignore spelling errors in the scoring, he can configure that using a high-level configuration API.

4. Summary

We presented ESCRITO, a toolkit for scoring of free-text answers in the educational domain. We support two user groups for educational NLP applications, teachers and NLP researchers, through both a high-level plug-and-play version and an easily extendable low-level API. We do so by providing baseline methods and setups for a number of common datasets, which are easily extendable through new datasets, preprocessing methods, or features.

ESCRITO enables *reproducible* research by carefully logging the experimental configuration and allowing to publish complete experimental setups including all preprocessing steps. ESCRITO directly addresses *multi-linguality* by providing preprocessing and feature extraction for a wide range of languages as well as a language-independent core of scoring functionality. We hope that ESCRITO will foster research on applying cutting-edge NLP technologies in educational applications as well as practical experimentation using free-text scoring in real-life scenarios.

5. Acknowledgments

This work has been funded by the German Federal Ministry of Education and Research under grant no. FKZ 01PL16075.

6. Bibliographical References

Attali, Y. and Burstein, J. (2004). Automated essay scoring with e-rater® v. 2.0. *ETS Research Report Series*, 2004(2).

Bär, D., Zesch, T., and Gurevych, I. (2013). DKPro Similarity: An Open Source Framework for Text Similarity. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 121–126, Sofia, Bulgaria, August. Association for Computational Linguistics.

Basu, S., Jacobs, C., and Vanderwende, L. (2013). Powergrading: a Clustering Approach to Amplify Human Effort for Short Answer Grading. *Transactions of the Association for Computational Linguistics (TACL)*, 1:391–402.

Chollet, F. et al. (2015). Keras. `https://github.com/fchollet/keras`.

Daxenberger, J., Ferschke, O., Gurevych, I., and Zesch, T. (2014). DKPro TC: A Java-based Framework for Supervised Learning Experiments on Textual Data. In *Proceedings of ACL: System Demonstrations*, pages 61–66, Baltimore, Maryland. Association for Computational Linguistics.

Dzikovska, M. O., Nielsen, R., Brew, C., Leacock, C., Giampiccolo, D., Bentivogli, L., Clark, P., Dagan, I., and Dang., H. T. (2013). SemEval-2013 Task 7: The Joint Student Response Analysis and 8th Recognizing Textual Entailment Challenge. **SEM 2013: The First Joint Conference on Lexical and Computational Semantics*.

Eckart de Castilho, R. and Gurevych, I. (2014). A broad-coverage collection of portable NLP components for building shareable analysis pipelines. In *Proceedings of the Workshop on Open Infrastructures and Analysis Frameworks for HLT*, pages 1–11, Dublin, Ireland. Association for Computational Linguistics and Dublin City University.

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

Horbach, A. and Palmer, A. (2016). Investigating active learning for short-answer scoring. In *Proceedings of BEA 2016*, pages 629–637.

Horbach, A., Ding, Y., and Zesch, T. (2017a). The influence of spelling errors on content scoring performance. In *Proceedings of the forth Workshop on NLP Techniques for Educational Applications (NLPTEA 2017)*.

Horbach, A., Ding, Y., and Zesch, T. (2017b). The Influence of Spelling Errors on Content Scoring Performance. In *Proceedings of 4th Workshop on NLP Techniques for Educational Applications (NLPTEA 2017) at IJCNLP*.

Horbach, A., Scholten-Akoun, D., Ding, Y., and Zesch, T. (2017c). Fine-grained essay scoring of a complex writing task for native speakers. In *Proceedings of the Building Educational Applications Workshop at EMNLP*, page to appear, Copenhagen, Denmark.

Horsmann, T. and Zesch, T. (2018). DeepTC – An Extension of DKPro Text Classification for Fostering Reproducibility of Deep Learning Experiments. In *Under review at LREC 2018*.

Meurers, D., Ziai, R., Ott, N., and Kopp, J. (2011). Evaluating Answers to Reading Comprehension Questions in Context: Results for German and the Role of Information Structure. In *Proceedings of the TextInfer 2011 Workshop on Textual Entailment*, pages 1–9, Edinburgh, Scottland, UK. Association for Computational Linguistics.

Mohler, M. and Mihalcea, R. (2009). Text-to-text semantic similarity for automatic short answer grading. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '09, pages 567–575, Stroudsburg, PA, USA. Association for Computational Linguistics.

Neubig, G., Dyer, C., Goldberg, Y., Matthews, A., Ammar, W., Anastasopoulos, A., Ballesteros, M., Chiang, D., Clothiaux, D., Cohn, T., Duh, K., Faruqui, M., Gan, C., Garrette, D., Ji, Y., Kong, L., Kuncoro, A., Kumar, G., Malaviya, C., Michel, P., Oda, Y., Richardson, M., Saphra, N., Swayamdipta, S., and Yin, P. (2017). DyNet: The Dynamic Neural Network Toolkit - Technical Report.

Östling, R., Smolentzov, A., Tyrefors Hinnerich, B., and Höglin, E. (2013). Automated essay scoring for swedish. In *The 8th Workshop on Innovative Use of NLP for Building Educational Applications, Atlanta, GA, USA, June 13, 2013*, pages 42–47. Association for Computational Linguistics.

Pado, U. and Kiefer, C. (2015). Short Answer Grading: When Sorting Helps and When it Doesnt. In Linköpings universitet Linköping University Electronic Press, editor, *Proceedings of the 4th workshop on NLP for Computer Assisted Language Learning, NODALIDA 2015*, Linköping Electronic Conference Proceedings, pages 42–50, Wilna, Mai. LiU Electronic Press and ACL Anthology.

Pearson, K. (1895). Note on regression and inheritance in the case of two parents. *Proceedings of the Royal Society of London*, 58:240–242.

Riordan, B., Horbach, A., Cahill, A., Zesch, T., and Lee, C. M. (2017). Investigating neural architectures for short answer scoring. In *Proceedings of the Building Educational Applications Workshop at EMNLP*, page to appear, Copenhagen, Denmark.

Spearman, C. (1904). The proof and measurement of association between two things. *The American journal of psychology*, 15(1):72–101.

Taghipour, K. and Ng, H. T. (2016). A neural approach to automated essay scoring. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 1882–1891.

Zesch, T., Heilman, M., and Cahill, A. (2015a). Reducing annotation efforts in supervised short answer scoring. In *Proceedings of the Building Educational Applications Workshop at NAACL*.

Zesch, T., Wojatzki, M., and Scholten-Akoun, D. (2015b). Task-independent features for automated essay grading. In *Proceedings of the Building Educational Applications Workshop at NAACL*.