Evaluation of Approaches for Automatic E-Assessment Item Annotation with Levels of Bloom's Taxonomy*

Roy Meissner[1][0000−0003−4193−8209], Daniel Jenatschke[2], and Andreas Thor[3][0000−0003−2575−2893]

[1] Leipzig University, Germany, roy.meissner@uni-leipzig.de
[2] Leipzig University of Telecommunications, Germany, d.jenatschke@gmx.de
[3] Leipzig University for Applied Sciences, Germany, andreas.thor@htwk-leipzig.de

Abstract. The classification of e-assessment items with levels of Bloom's taxonomy is an important aspect of effective e-assessment. Such annotations enable the automatic generation of parallel tests with the same competence profile as well as a competence-oriented analysis of the students' exam results. Unfortunately, manual annotation by item creators is rarely done, either because the used e-learning systems do not provide the functionality or because teachers shy away from the manual workload. In this paper we present an approach for the automatic classification of items according to Bloom's taxonomy and the results of their evaluation. We use natural language processing techniques for pre-processing from four different NLP libraries, calculate 19 item features with and without stemming and stop word removal, employ six classification algorithms and evaluate the results of all these factors by using two real world data sets. Our results show that 1) the selection of the classification algorithm and item features are most impactful on the F1 scores, 2) automatic classification can achieve F1 scores of up to 90% and is thus well suited for a recommender system supporting item creators, and 3) some algorithms and features are worth using and should be considered in future studies.

Keywords: E-Assessment, Items, Annotation, Bloom's Taxonomy, Data Mining, Machine Learning Systems, Performance Levels, Knowledge Based Systems

1 Introduction

E-Assessment is an integral part of e-learning and many learning management systems support the creation of items and the execution of online tests or online exams. Despite this technical support, item creation is still a time-consuming process for teachers whose main goal is to prepare an online test in time. Additional work, like the annotation of items with metadata, often falls by the wayside, as its added value is only apparent after a certain timespan. An example for

* This work was supported by the German Federal Ministry of Education and Research for the tech4comp project under grant No 16DHB2102.

such annotation of items are levels of Bloom's taxonomy [5]. This annotation has no influence on the conduct of the online test and is not visible to the students at all. For large item pools, such annotations support teachers in the long run, since they can, for example, compare exams if they are competence-equivalent or create tests specifically for individual student groups.

Automatic item annotation is therefore a promising way to ensure high and comprehensive data quality (as many items as possible are correctly annotated) with low resource input (teachers check only difficult cases if necessary). Content and wording of the problem context as well as the actual question of an item usually contain all the information that a human being needs to classify an item into one of Bloom's levels. On the other hand, automatic annotation is difficult because domain experts have multi-layered background knowledge against which they assign an item to a certain level.

This paper presents a comprehensive evaluation of approaches for automatic item annotation with levels of Bloom's taxonomy. Using two real-world item pools from the fields of Computer Science and Educational Sciences, both machine-learning-based and rule-based methods are evaluated. The individual parts are systematically varied, e.g. pre-processing by means of NLP techniques or the selected machine learning method. As a result, the evaluation discusses the main factors influencing effective and efficient automatic item annotation.

The paper is structured as follows: In section 2 we introduce two generic approaches to automatic item annotation. Both are based on a preprocessing using typical NLP techniques and use a machine learning method or a rule-based knowledge base for classification. Section 3 presents the main contribution of this paper and discusses the results of the comprehensive evaluation. Different parameter configurations are systematically evaluated with two datasets and their results are interpreted. Section 4 discusses related work before we end with a short summary and an outlook on future work in Section 5.

2 Automatic Item Classification

Bloom defined six classes to structure cognitive learning outcomes [4]. He associated them with several inclusion conditions, based on encountered words, like verbs such as *arrange*, *define* or *describe* for the first taxonomy level *knowledge*. In this section we briefly describe two approaches for automatic item classification using Bloom's taxonomy levels. The first approach, rule-based, utilizes a set of manually curated rules that look for the aforementioned keywords and assign corresponding weights to the levels. The second approach employs machine learning and thus requires test and training data, i.e., items have already been manually classfied.

First of all, all items are preprocessed in a uniform way as shown in figure 1. Each item is first converted into a document that contains the item's context description and the question. Item type-specific information such as answer options for single- and multiple-choice questions, which are naturally missing for free-text questions, are not considered. Then, item features are extracted from

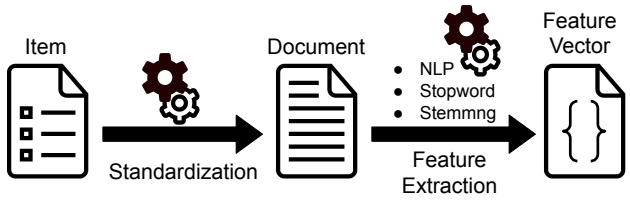

Fig. 1. Common item preprocessing pipeline

the document using Natural Language Processing (NLP) techniques. All 19 features from table 3 are caclculated for each item and form specific feature vectors. For the extraction of the items we use standard NLP libraries. In addition, the removal of stop words and stemming is done. Thus, items are transformed into feature vectors, which are used for both, the rule-based approach and the machine learning-based approach.

For the rule-based approach we follow the work of [12]. Bloom listed keywords (primarily verbs) and their assigned levels in [4], which we converted to rules. The assignment is done with a rule weight to characterize their relevance. Using the created rule set for item classification is straight forward: The obtained feature vector is searched sequentially for the keywords that occur in the rule set and the weights are added per level. One exemplary rule is: if the term *arrange* is found within the vector, increment the level *knowedge*. The item is lastly classified into the level with the maximum weight sum.

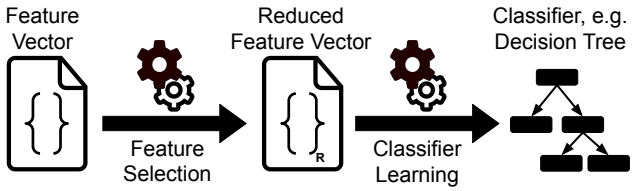

Fig. 2. Process for classifier learning

Figure 2 shows the procedure for the machine learning-based approach. Here the feature vectors of the training data are first optionally reduced, so that not all information has to be included in the creation of the classifier (we evaluate the influence of item features in section 3.2). The reduced item vectors are then passed to a machine learning algorithm (e.g. a decision tree algorithm) so that an executable classifier can be calculated. We will evaluate the quality in the following section.

3 Evaluation

3.1 Data Sets, Configuration & Metrics

For the evaluation we use two data sets from two different domains: Computer Science and Educational Sciences. Table 1 shows key statistics on the number

Data Set	Domain	#Items	Avg. Item Length in Characters
A	Computer Sc.	83	141.6
B	Educational Sc.	292	283.5

Table 1. Description of the data set used in the evaluation. The length of an item is defined as the number of characters of its context description and question.

of items and their size. Each item was independently classified by three domain experts and the majority opinion was adopted for the item, as for some items the domain experts ended up with different classifications. Since item classification is a highly context sensitive task it might be worth allowing more than one class per item. We leave this multi-class classification as subject for future work.

As so often, real world data sets show imbalances in their characteristics. For data set A, for example, the number of items in the level *apply* is about three times as large as for *knowledge*. In contrast, for data set B the class *knowledge* is three times as large as *comprehension*.

Items are used as both, training and test data sets for the machine learning approach and we used a 10-fold cross-validation method[4]. Partitioning, training and testing is not needed for the rule-based approach, as the rules were already verbalized [3].

In our experiments we varied the following parameters:

- NLP Libraries: Pattern, Open NLP, TreeTagger and Stanford CoreNLP
- Text Preprocessing: Stemming and Stop Word Removal
- Machine Learning Algorithms: Six different classifiers (see Table 2), which are all supported by the used library scikit-learn[5].
- Item Features: Up to four out of 19 item features (see table 3)

Classifier	Configuration
DTC: Decision Tree Classifier	max_depth = 5
GNB: Guassian Naive Bayes	none
KNN: k-nearest-Neighbor	#neigbors = 5
SVM: Support Vector Machine	gamma=2, C=1 cache_size=7000
RFC: Random Forest Classifier	max_depth=5 max_trees=10
QDA: Quadr. Discriminant Analysis	none

Table 2. Description of the six employed machine learning-based classifiers including their configuration.

[4] `https://scikit-learn.org/stable/modules/cross_validation.html`
[5] `https://scikit-learn.org/stable/index.html`

We ran the evaluation for all possible parameter combinations and report the minimum, maximum and average F1 scores, as well as the standard deviation of the F1 scores.

Code	**Attribute**
W	Bag of Words
V	Bag of Words but verbs only
N	Bag of Words but nouns only
CS	Number of sentences
CT	Number of tokens
CV	Number of verbs
CN	Number of nouns
TPS	Number of tokens per sentence (CT/CS)
NPS	Number of nouns per sentence (CN/CS)
NPT	Number of nouns per token (CN/CT)
VPS	Number of verbs per sentence (CV/CS)
VPT	Number of verbs per token (CV/CT)
KM	Number of keywords
KMR	Number of keywords per token (KM / CT)
PD	Number of Part of Speech (POS) Tags
PRD	Number of POS Tags per token (PD / CT)
PCD	Number of POS classes
PCRD	Number of POS classes per token (PCD / CT)
IT	Item type

Table 3. List of item features

3.2 Machine Learning-based Approaches

Figure 3 illustrates the F1 scores grouped by the NLP library. Obviously the employed library has only minor influence on the results and in general it seems irrelevant which NLP library is used. Maximum values differ only up to 0.38% and average values differ up to 3.9%, which corresponds with a differing in standard deviation of up to 2.4%. Only the library Pattern shows a small advantage due to its low standard deviation and high average score. Of course, the library used can have a large influence on the runtime, but this is not part of this evaluation.

Figure 4 illustrates the results by the used classifier. Obviously, 5 out of 6 algorithms deliver comparable results, from which the Decision Tree Classifier (DTC) algorithm performs best. It delivered the highest maximum F1 score of 0.907, the highest minimum F1 score, performs 0.074 points better then the second best algorithm on average and has the second best standard deviation of 0.078. Second best algorithms are GNB, and SVN, but GNB has a much larger dispersion than SVN but only slightly better average and maximum values. This

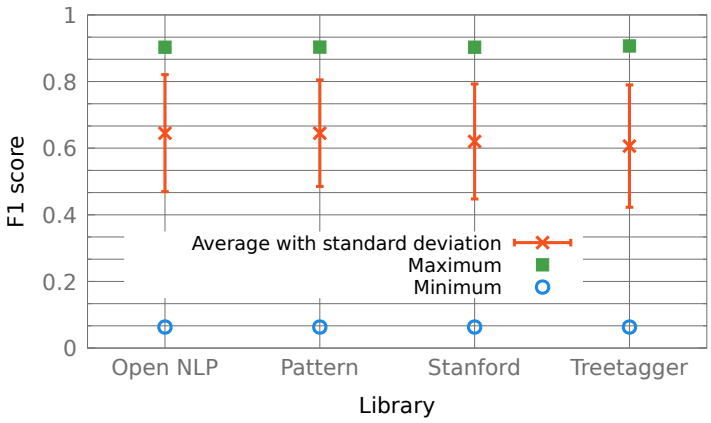

Fig. 3. F1 scores for different NLP libraries (Data set A)

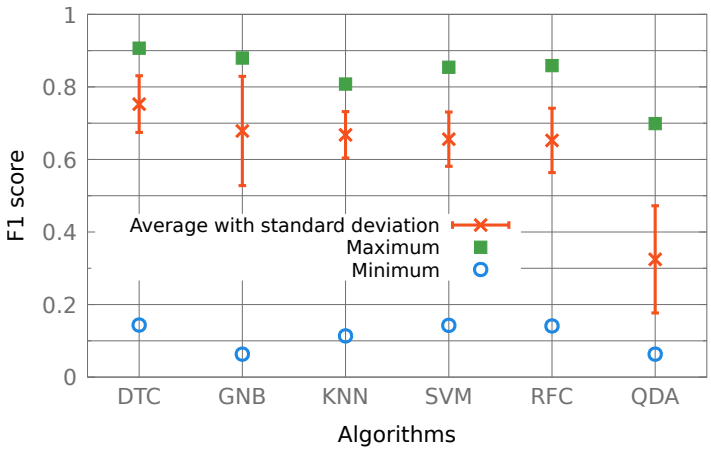

Fig. 4. F1 scores for the different classifiers (Data set A)

classifies the algorithms DCT and SVN as candidates for future work, e.g. for algorithm tuning. On the other hand, GNB is a parameter-free method and thus a suitable candidate for first exploratory studies. The same conclusions can be drawn from data set B whose results show a similar distribution with higher minimum and lower maximum values, but about the same average and standard deviation values. Interestingly the QDA algorithm performs much better for data set B (avg. F1 score is 0.62), but is also outperformed by the first four algorithms from figure 4.

In our third experiment we investigate in the influence of stop word removal and stemming. The results for these pre-processing steps are shown in table 4. They reveal that stemming has nearly no influence on the average and minimum results, even though there is a mixed difference in the maximum values. In contrast, stop word removal decreases the average F1 scores by 3.75% on average and increases the standard deviation by 0.9% on average. Data set B shows

Stem	Stop	Avg	Min	Max	σ
✗	✓	0.608	0.0632	0.872	0.178
✓	✓	0.607	0.0632	0.888	0.177
✗	✗	0.638	0.0632	0.907	0.170
✓	✗	0.635	0.0632	0.883	0.168

Table 4. F1 scores for stemming (Stem) and stop word removal (Stop) for data set A. A check mark indicates usage of the algorithm, a cross mark that the algorithm was not used.

the same characteristics, with the only difference that the usage of stemming increases the maximum results by 1.4% on average. Regarding these mixed results from both data sets and the higher tendency of decreasing the F1 scores, we can not recommend the usage of either of these algorithms for classification performance improvements.

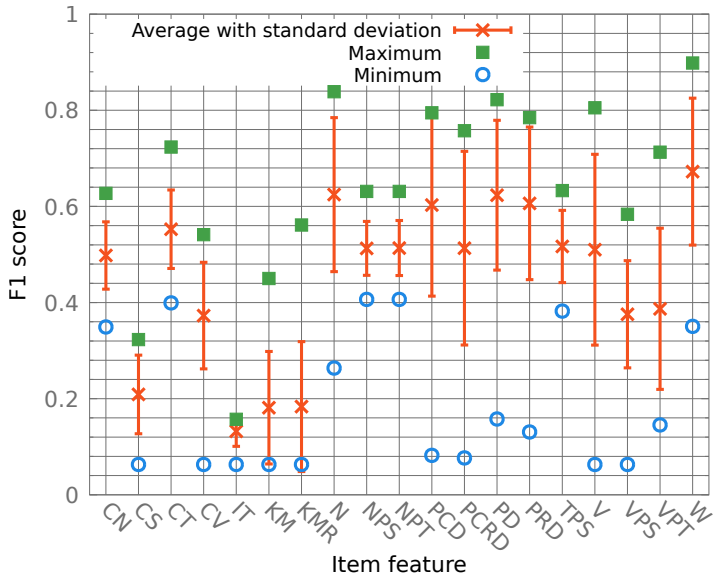

Fig. 5. F1 scores for the different data attributes for data set A. See table 3 for their abbreviations.

Figure 5 evaluates the individual influence of the 19 extracted item features for data set A. Only four features achieve a maximum F1 score of more than 0.8 (N, PD, V, W). These features (with the exception of V) also achieve the best average F1 scores, but their standard deviation is among the eight highest ones. In particular, item feature W (Bag of Words) achieves the best overall scores but has a poor standard deviation. The results for Bag of Words might explain why the maximum results of the different NLP libraries are so close. A simple item processing with Bag of Words, without stemming and stop word removal already achieves an F1 score of 0.88 using the Decision Tree Classifier. Such a

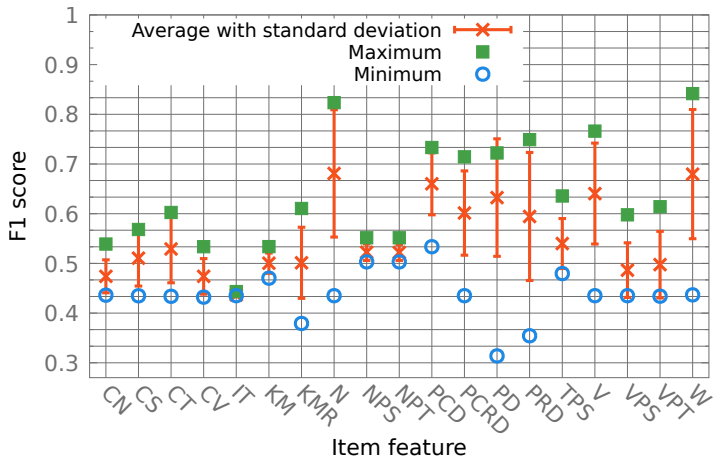

Fig. 6. F1 scores for the different data attributes for data set and B. See table 3 for their abbreviations.

simple workflow does not require any of the advanced preprocessing techniques and is supported by all NLP named libraries.

A second observation is that all features related to Part of Speech tagging (PD, PRD, PCD, and PCRD) show comparitivly good results, which indicates that analysing the word classes of items gives information about the performance level classification. This corresponds with the high results for verbs (V) and nouns (N). It seems that nouns are more valuable for determining the performance level of items, even though we would have anticipated that verbs are more valuable, as they prompt to do something. These results are contrary to the list of words Bloom defined in [4], which are mostly verbs and which are used for the rule-based approach. A possible explanation is that nouns appear more often within items (about 32% of words are nouns, about 10% are verbs in our data sets). At the same time, the significance of nouns and verbs provides a possible explanation why stemming and stop word removal do not have a significant influence on the results (see table 4).

Figure 6 shows the individual influence of the 19 item features for data set B. Again all the minimum values are higher and all the maximum values are lower than for data set A. Apart of this fact the results show a comparable impact of the above named features from data set A. Of particular intrest are the bad performing features from figure 5 (top) (CS, IT, KM, KMR), which perform much better for data set B. A detailed analysis of the data set regarding these features revealed that they perform better because of data set characteristics, like the usage of specific item types for specific performance levels. There is also much less difference between features V and N which stems from different phrasing of items.

To further increase the quality of the classification, up to four random item features were used in a final experiment. Figure 7 shows the results grouped by

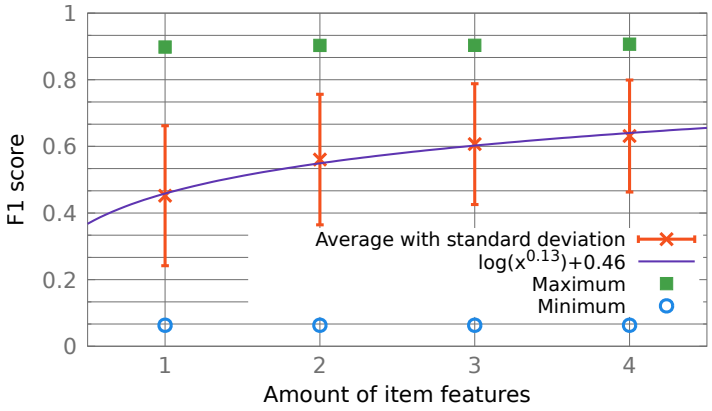

Fig. 7. F1 scores for combinations of item features for data set A. The data represents averages for all runs with x item features.

the number of features used for data set A. A first observation shows that both the maximum and minimum F1 scores are almost not improved by adding more features. However, the increase of the average F1 score as well as the reduction of the standard deviation is striking, i.e., the classification results become more robust on average. Data set B is also showing the same behaviour with slightly different scores and again lower maximum and higher minimum values.

To summarize the results, the biggest influence factors on the performance of the machine learning-based classification are the chosen machine learning algorithm, as well as the employed item features. Using different NLP libraries has nearly no impact on the maximum values, even though a slight difference in average values is observable. Stop word removal and stemming can be omitted. Combining different item features is not delivering significantly higher maximum F1 scores but increases the average F1 score. In general, the achieved F1 scores (maximums are 0.907 and 0.84 for data set A and B, respectively) may allow for an automatic item annotation but are in any case suitable for a recommender system.

3.3 Rule based system

In contrast to the machine learning-based approaches, the rule-based approach does not require any training data or parameters to be specified. We thus calculated precision, recall and F1 score for each of Blooms classes (i.e., levels) and computed the average, minimum, and maximum value. So in table 5 the average F1 score is the average of the F1 scores of all classes; minimum and maximum values correspond to the minimum and maximum F1 scores in the distribution of the target classes. Thus these last two values are *not* comparable to subsection 3.2 where minimum and maximum were calculated over all parameter configurations.

As visible in table 5 the average F1 scores are low, differ a lot between the two data sets and have a high standard deviation. Especially the minimum F1 scores

Data Set	Avg	Min	Max	σ
A	0.485	0.18	0.79	0.305
B	0.378	0.08	0.63	0.289

Table 5. F1 scores for the rule-based systems for both data sets.

are low and show that this approach performs bad for selected target classes. These results are not surprising as we already introduced in section 3.1 that the mapping of items to target classes is context sensitive. The implemented rule base is not context sensitive, even though the rating vector has been initialized with values that corresponds to the target class distribution within the data set.

As seen in section 3.2 and figure 5 the item features that correspond to the ones used for the rule-based system, namely Number of Keywords (KM) and Item Type (IT), perform bad in comparison to other item features. Especially KM coincides with the results presented in this subsection and reaches with an F1 score of 0.45 about the same dimension as the rule-based system for data set A. Furthermore the results show that the item type alone is no satisfying indicator for performance levels, as attribute IT only reaches a maximum F1 score of 0.16 and 0.44 for data sets A and B. This might stem from the fact that IT provides more information about which performance levels are not applicable than which are.

An analysis of the available items show that the keywords used for the rule based approach only appear in about 12.1% of all items for data set A and 29.6% of all items of data set B. There are two possible interpretations: 1) the item authors did not use the phrasing Bloom proposed for items. They should stick to standard phrasing in order to achieve better classification and have clear items for students. 2) The inclusion rule set, gathered by Bloom, is too small and misses typical keywords and phrases. As discussed for items above, the rule set and thus the amount of keywords and phrases might be too small and should be extended by experts. The focus of the keywords on verbs seems limited, as our results revealed that nouns are used much more often and seem to have a higher impact on the classification results.

4 Related Work

Itemy may be automatically classified during their creation process from existing knowledge sources, like ontologies [2], Linked Open Data [8] or text [9]. This process involves item templates, like sentence patterns or parameterized SPARQL queries. Used templates are annotated with Blooms taxonomy and as a result every created item is also annotated with the respective data [7]. A downside of these approaches is that they only considor the currently created item and are context-free, thus ignoring information the classification might benefit from or depend on, like similar items and available learning material.

There are rule-based, machine learning-based and neural network-based approaches to classify existing items, as well as combinations of the former ones.

Rule-based approaches either focus on rules that have been created by experts, or they on creating rules from existing data sets. Chang et al. used the expert approach and obtained comparable results to our study [6]. Haris et al. showed a much better average F1 score of 0.77 for their automatic rule creation system, which in addition analysis the semantic structure of items [10]. Jayakodi et al. use a rule creation system, but did not measure the classification performance in detail. They concentrated on assessing the benefits of available verbs of items, which they summarize to be a limited approach [11].

Machine learning-based approaches for item classification typically employ only a small number of features (e.g. up to four features in [1,15]). Stemming, stop word removal, and various NLP techniques are used by many authors without testing their performance impact, which we have done in our evaluation. A recent survey shows that most approaches focus on support vector machines, k-nearest Neighbor and Naive Bayes as algorithms [14]. The results of our study indicate that decision tree and random forest classifiers show comparable results. Neural Networks are used by Yusof et al., whom focused on convergence time and classification precision, but not on recall and F1 scores [15]. Interestingly they rated the attribute document frequency (DF) as a valuable attribute for precise classification results.

Lastly it is possible to combine different classifiers in voting and ensemble systems to improve the quality of the classification [1,13]. For example Osadi et al. reports an average F1 score of 0.79 for all of Blooms classes, which is comparable to our maximum F1 scores [13].

In general it is noticeable that there is no generic data set available for providing comparable results and that F1 scores, if available at all, differ a lot between different research groups and used data sets, as also shown in the literature review by Sangodiah et al. [14]. Additionally most articles are not analysing their results with respect to external factors and typically do not focus on more than four item features, which, in contrast, we did within this paper.

5 Summary and Future Work

In this paper we presented a comprehensive evaluation of machine learning and rule-based systems for the automatic classification of items with performance levels defined by Bloom. The results show that the machine learning approaches outperform the rule based approach that primarily addresses key verbs. The actual classifier and the employed item features have the biggest influence of the F1 score of the classification.

In future work we will extend the rule-based approach so that not only verbs and the item type are considered. Our evaluation results show that nouns and information from POS tagging should also be considered. Another field of research is the parameter optimization of the algorithms, especially in view of the fact that the results sometimes produced large standard deviations, as well as the consideration of neural networks. There are also a lot of experimental input features available, which have not been tested for the assessment topic so far. Finally, the

provision of standardized training and test data sets would also be a valuable contribution to the community. Results, scripts, and programs used for this study are provided at `https://gitlab.com/Tech4Comp/automatic-item-annotation`

References

1. D. A. Abduljabbar and N. Omar. Exam questions classification based on bloom's taxonomy cognitive level using classifiers combination. *Journal of Theoretical and Applied Information Technology*, 78(3):447, 2015.
2. T. Alsubait, B. Parsia, and U. Sattler. Generating multiple choice questions from ontologies: Lessons learnt. In *Proceedings of the 11th International Workshop on OWL: Experiences and Directions (OWLED 2014)*, pages 73–84, 2014.
3. C. Beierle. *Methoden wissensbasierter Systeme : Grundlagen, Algorithmen, Anwendungen*. Vieweg + Teubner, Wiesbaden, 4., verb. aufl. edition, 2008.
4. B. S. Bloom. *Taxonomie von Lernzielen im kognitiven Bereich*. Beltz-Studienbuch. Beltz, Weinheim u.a., 3. aufl. edition, 1973.
5. B. S. Bloom et al. Taxonomy of educational objectives. vol. 1: Cognitive domain. *New York: McKay*, pages 20–24, 1956.
6. W.-C. Chang and M.-S. Chung. Automatic applying bloom's taxonomy to classify and analysis the cognition level of english question items. In *2009 Joint Conferences on Pervasive Computing (JCPC)*, pages 727–734. IEEE, 2009.
7. M. Cubric and M. Tosic. Towards automatic generation of e-assessment using semantic web technologies. *International Journal of e-Assessment*, 2011.
8. M. Foulonneau. Generating educational assessment items from linked open data: The case of dbpedia. In R. Garcia-Castro, D. Fensel, and G. Antoniou, editors, *Prof. of the Semantic Web ESWC 2011 Workshops*, 2011.
9. C. Gutl, K. Lankmayr, J. Weinhofer, and M. Hofler. Enhanced automatic question creator–eaqc: Concept, development and evaluation of an automatic test item creation tool to foster modern e-education. *Electronic Journal of e-Learning*, 9(1):23–38, 2011.
10. S. S. Haris and N. Omar. A rule-based approach in bloom's taxonomy question classification through natural language processing. In *2012 7th International Conference on Computing and Convergence Technology (ICCCT)*, pages 410–414. IEEE, 2012.
11. K. Jayakodi, M. Bandara, and I. Perera. An automatic classifier for exam questions in engineering: A process for bloom's taxonomy. In *2015 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*, pages 195–202. IEEE, 2015.
12. N. Omar, S. S. Haris, R. Hassan, H. Arshad, M. Rahmat, N. F. A. Zainal, and R. Zulkifli. Automated analysis of exam questions according to bloom's taxonomy. *Procedia-Social and Behavioral Sciences*, 59:297–303, 2012.
13. K. Osadi, M. Fernando, W. Welgama, et al. Ensemble classifier based approach for classification of examination questions into bloom's taxonomy cognitive levels. *International Journal of Computer Applications*, 162(4):76–92, 2017.
14. A. Sangodiah, M. Muniandy, and L. E. Heng. Question classification using statistical approach: A complete review. *Journal of Theoretical & Applied Information Technology*, 71(3), 2015.
15. N. Yusof and C. J. Hui. Determination of bloom's cognitive level of question items using artificial neural network. In *2010 10th International Conference on Intelligent Systems Design and Applications*, pages 866–870. IEEE, 2010.