Semantic-Based Feedback Recommendation for Automatic Essay Evaluation

Tsegaye Misikir Tashu[1,2] and Tomáš Horváth[1,3]

[1] ELTE Eötvös Loránd University, Faculty of Informatics,
Department of Data Science and Engineering, Telekom Innovation Laboratories
Pázmány Péter sétány 1/C, 1117, Budapest, Hungary
[2] ELTE Eötvös Loránd University, Faculty of Informatics,
3in Research Group, Martonvásár, Hungary
[3] Pavol Jozef Šafárik University in Košice, Faculty of Science, Institute of Computer Science
Jesenná 5, 040 01 Košice, Slovakia
{misikir,tomas.horvath}@inf.elte.hu http://t-labs.elte.hu

Abstract. With the development of sophisticated e-learning platforms, educational recommender systems and automatic essay evaluation are becoming an important features in e-learning systems. Most of the works in educational recommendation techniques are focused in recommending learning materials or learning activities to the learners. In this paper we proposed and implemented a semantic-based feedback recommendation approach for automatic essay evaluation, which will allow assessors to interact with automatic essay evaluation systems, give feedback on learner's essay solution in the form of textual comments and provide recommendation to other similar essay solution based on the solution which the assessor has given textual feedback. To compute the semantic similarity and to provide feedback recommendation, we used neural word embedding and relaxed word mover's similarity. The proposed approach achieves high performance accuracy, compared to the state-of-the-art methods, according to our experimental results.

Keywords: Educational Recommender system, Semantic similarity, Text similarity, Feedback recommendation

1 Introduction

Existing commercial and open source e-Learning platforms offer functionalities to help teachers and support learners. Although e-Learning have come a long way, some of its aspects are still in their early stages. One of such aspect is e-Testing, offered by most of the platforms, however, these e-Testing authoring tools have limited functionalities.

Using e-Testing modules, e-Learning platforms can provide both subjective and objective exams. We denote an exam "objective" if it has objective evaluation criteria (e.g. computing a mathematical equation or choosing from several options which might be correct or not, etc.). On the contrary, we call an exam subjective if the correctness of it's solution depends on subjective preferences of the evaluator (e.g. an essay about a specific topic where not only the information contained are evaluated but also it's writing style or some other characteristics). However, providing appropriate feedback (score and textual comment) timely, especially for subjective type of exams, is a challenging task for the teacher and is not yet addressed well in most of the e-Testing modules. Moreover, for the solution students submitted through e-Testing systems, providing automated feedback suggesting a possible solution are also not well addressed in the literature. On the other hand, e-Testing systems should also provide additional features that help teachers to highlight parts of a text (student solution) and provide a comment related to this part.

Adding these features into current e-Testing systems might not be challenging but making them, in some sort of sense, intelligent is an open research issue. Rather than letting the assessor to give feedback to all student solution in a pen and pencil way, which is a high burden to a human and also time consuming, we need to develop algorithms that will choose some representative student solutions for which the assessor would give feedback (in form of textual comments) and the system will automatically recommend feedback to other student solutions based on their similarity to the chosen representative solutions.

The issue related to essay scoring can be addressed by using Automatic Essay Evaluation (AEE) systems which are used to automatically evaluate and score essay exam solutions considering both their syntax and semantics [19,23]. Many evaluative studies have reported relatively high levels of correspondence between the scores produced by AEE systems and those produced by human markers [19,8,2,23]. However, despite these positive results and the potential benefits of AEE technology, AEE systems are yet to be widely accepted by professional educators [26,13]. A possible reason for this is the lack of transparency of these systems [6] in the results produced leading to assessors' low confidence in the validity of AEE systems.

Issues on AEE systems (acceptance) and the issue of providing textual comments on student solution can be addressed using the following manners: One way is to allow the assessors to interact with AEE systems by crosschecking the machine score and allowing to modify this score. The second solution is to incorporate a feedback recommendation system that allows the assessor to select part of a student solution, give comments on this part and the feedback recommendation system will recommend the most likely to be comment for other student solution provided.

Therefore, in this paper we propose a semantic-based feedback recommendation approach for AEE systems that will allow the assessors to interact with systems, allow them to give feedback and give recommendation to other similar essays based on their similarity to the solution which has been evaluated by the assessor.

The rest of this paper is organized as follows: Section 2 provides an overview of the existing works and approaches. In Section 3, the proposed semantic based feedback recommendation approach is introduced. Experiments and results are described in Section 4. Section 5 concludes the paper and discusses prospective plans for future work.

2 Related Work

According to AL-Smadi and Gütl [1], the reasons for using e-Testing instead of pen-and-pencil tests are both practical and pedagogical. The practical ones are given by the increase in students numbers and, implicitly, of assessors quantity of work. The e-Testing is meant to resolve the problem of evaluation of a large number of students in a short period of time. The pedagogical reasons come from the need for systems which evaluate students knowledge correctly and efficiently. In the past, the purpose of the e-Testing systems was to shorten the time spent by the teachers for the evaluation process, but now, the e-Testing systems have new challenges to overcome: the efficient management of questions, the building of intelligent tests, and providing timely feedback to learners in the form of textual comments.

The research on automatic essay exam evaluation and scoring essay exams on the e-learning platform is ongoing for more than a decade where Machine Learning (ML) and Natural Language Processing (NLP) techniques were used for evaluating essay exams. The history of developing AEE systems started in 1966 by[19] and followed by other research works like E-Rater [3,2], Intelligent Essay Assessor (IEA) [8], IntelliMetric [22], Pairwise [23] and others. These systems automatically score essays but assessors are not allowed to interact and give feedbacks in the form of textual comments on each students solutions.

e-Learning systems use different recommendation techniques in order to suggest online learning activities to learners, based on their preferences, knowledge and the browsing history of other learners with similar characteristics. Recommender systems assist the natural process of relying on friends, classmates, lecturers, and other sources of making the choices for learning [16].

Most of the works on recommendation techniques in education are focused on recommending learning materials or learning activities to the learners [11,14]. On the other hand, educational data mining has considered to support universities, teachers and learners by helping the learners improve their performance, to know how the learners learn and how they adapt to new problems [24].

There were also attempts to develop recommender systems that provide feedback to the students' essay submissions in the form of textual comments. Gibbs and Simpson [9] describe several conditions under which feedback has a positive influence on learning. Feedback should be (i) sufficient in frequency and detail (ii) focused on students performance, on their learning and on the actions under students control rather than on the students themselves and/or on personal characteristics, (iii) timely, in that it is received by students while it still matters and in time for application or for asking further assistance, (iv) appropriate to the aim of the assignment and its criteria, (v) appropriate in relation to students conception of learning, their knowledge and of the discourse of the discipline (vi) attended to and (vii) acted upon.

The Altered Vista (AV) system, proposed by Recker and Walker [25], uses a database in which learner evaluations of learning resources are stored. It allows learners to browse the reviews of others and get personalized learning resource recommendations from the system. AV does not support learners directly by giving them feedback on their work. Instead, it provides an indirect learning support by recommending suitable learning tools.

Another similar web based application called PeerGrader (PG) was also introduced [10]. PeerGrader helps learners' to improve their skills by reviewing and evaluating solutions of their fellow learners blindly. In PG, each learner will get a task list and can choose a task by the PG. Learners submit their solutions and another learners can read these solutions and provide textual comments. Learners can modify solutions based on the comments received and re-submit an updated version of their solutions again to the system where other learners can review it again. After learners submitted their final updated version, the PG calculates grades for these solutions. The evaluation of a single learner answer is very time consuming because of the complexity of the reviewing process and the textual comments. This may cause learner dropouts and deadline problems [20].

The Scaffolded Writing and Rewriting in the Discipline (SWoRD) system was introduced by [5] to address the problem of writing homeworks in the form of a long text which cannot be reviewed in detail by a teacher for time reasons. SWoRD relies on peer reviews. Students who conduct peer reviews, read possible task solutions that were provided by other students and evaluate them. Based on peer reviews, the system provides feedback to learners in the form of recommendations.

All the previous works recommend feedbacks based on the peer reviews. This is very time consuming for the learner to wait and to get the feedback and also the feedback might not even be useful because their fellow learners' might make the review blindly. Therefore, in this paper we propose a semantic-based feedback recommendation approach for AEE systems that will allow assessors to give feedback in the form of textual comments on selected student essay solutions and make recommendations to other similar essay solutions based on evaluated solutions by the assessor.

3 Feedback Tag Recommendation

As we discussed in sections 1 and 2, in this paper we will address an open research problem called semantic-based feedback recommendation. Our e-Testing system [4] which was launched eight (8) months ago has different features that help both the assessors and learners. It allows assessors to create their own course, add students they want to assess, create both subjective and objective exams for the course and it also allows the assessors to give feedbacks in the form of textual comments on selected (and highlighted) parts of students submissions. The system allows learners to register and get their own username and password, send a request to register for course, Submit their solution for the registered exams and view their score and the assessor's feedbacks.

The feedback system is the same as a manual way of giving textual feedbacks to the learners.

To make the feedback feature of the system more useful and supportive towards the assessors, we introduce a semantic based feedback recommendation approach which works in the following manner: First, the assessor gives textual comments on some essays by selecting a phrase, a sentence or a paragraph. On each students essay, the assessor can give as many comments as he/she wants. Then, feedback recommendations system will find similar essays of other students, check whether the highlighted text (or a very similar text) is present in those essays and give recommendation for comments. To address the issue, we have proposed and implemented a semantic-based feedback tag

[4] http://etestsupport.com/

recommendation approach. We also implemented two baseline solutions, introduced in algorithms 1 and 2, for experimental purposes.

The first baseline is a simple pattern matching approach, introduced in the Algorithm 1, that has the following data on its input:

- a student essay e, a text, for which the feedbacks are going to be recommended,
- the set of all essays E submitted by all the students not evaluated so far by the teacher, such that each essay correspond to one student (i.e. $|E|$ is the number of students),
- the set $F = \{(h,c) \mid h \text{ is a text}, c \text{ is the teacher's comment to } h\}$ of teacher's feedbacks present in already evaluated essays, i.e. comments c on some (parts of) texts h from students' essays evaluated by the teacher.

The algorithm simply looks up in a so far not evaluated essay e if it's parts are identical to already commented parts of other essays on the same topic.

Algorithm 1 General Rule based algorithm

```
procedure TAG RECOMMEND($e, E, F$)
    for all $e \in E$ do
        $e.sentences$ = tokenize($e$)          ▷ Create sentence tokens for each essay $e$
        for all $(h,c) \in F$ do
            for all $s \in e.sentences$ do
                if $h \subseteq s$ then
                    $F \leftarrow F \cup (s,c)$
```

The second baseline, introduced in the Algorithm 2, is based on semantic and lexical similarity with the same data on input as in the case of Algorithm 1 and a similarity threshold θ what is a hyper-parameter of the algorithm and has to be set up by the user (might require a certain domain knowledge). Equation 1 and 2 are used in algorithm 2 to compute the similarity.

Algorithm 2 General Similarity based algorithm

```
procedure TAG RECOMMEND($e, E, F, \theta$)
    for all $e \in E$ do
        $e.sentences$ = tokenize($e$)          ▷ Create sentence tokens for each essay $e$
        for all $(h,c) \in F$ do
            for all $s \in e.sentences$ do
                if $similarity(h,s) \geq \theta$ then
                    $F \leftarrow F \cup (s,c)$
```

3.1 Cosine similarity

Cosine similarity is one of the most widely used lexical similarity measure in text document similarity. Given the set of all essays E submitted by all the students not evaluated so far by the teacher, sentences $s_1, s_2, \ldots, s_n$ of a student essay $e \in E$ and the set $F = \{(h,c) \mid h \text{ is a highlited text}, c \text{ is the teacher's comment to } h\}$ of teacher's feedbacks present in already evaluated essays, i.e. comments c on some (parts of) texts h from students' essays evaluated by the teacher, the cosine similarity between each s_j (where $1 \leq j \leq n$) and a given h is defined as follows:

$$cosine\_sim(s_j, h) = \frac{x_{s_j} x_h}{\|x_{s_j}\| \|x_h\|} \tag{1}$$

where x_{s_j} is a vector representation of sentence s_j of the essay e and x_h is a vector representation of a highlighted text h.

3.2 Relaxed Word Movers based similarity

To compute the semantic similarity between sentences in essays and highlighted text, we use the word mover distance [15] and we will redefine it as a relaxed word mover similarity using cosine similarity.

The relaxed word mover similarity utilizes the property of *word2vec* embeddings [17]. Therefore, the similarity between two text documents D_1 and D_2 is the maximum cumulative similarity that word vectors from document D_1 travels to match exactly to word vectors of document D_2.

In this regard, in order to compute the semantic similarity using the relaxed word movers similarity between sentences $s_1, s_2, \ldots, s_n$ contained in an essay $e \in E$ not evaluated so far by the teacher and h a highlighted text from the set F of teacher's feedbacks present in already evaluated essays, defined above, $s_1, s_2, \ldots, s_n$ will be mapped to h using a word embedding model. Let $\mathbf{s}_j$ and $\mathbf{h}$ be nBOW representations of s_j and h, respectively, where $1 \leq j \leq n$. Let $T \in \mathbb{R}^{m \times m}$ be a flow matrix, where $T_{kl} \geq 0$ denotes how much the word w_k in s_j has to "travel" to the word w_l in h, and m is the number of unique words appearing in s_j and h. To transform $\mathbf{s}_j$ to $\mathbf{h}$ entirely, we ensure that the complete flow from the word w_k to the word w_l equals to some value d_k. The relaxed word movers similarity is defined as follows using cosine similarity measure:

$$\max_{T \geq 0} \sum_{k,l=1}^{m} T_{kl} \, cos\_sim(w_l, w_k) \tag{2}$$

subject to

$$\sum_{k=1}^{m} T_{kl} = d_k, \forall i \in \{1, \ldots, n\}$$

4 Experimental study

4.1 Dataset selection and preparation

In order to perform the experiment, a data set which has highlighted texts and comments on the students' essay is mandatory. However, we could not find such a data set, probably because there was no attempt to address such an issue so far. Therefore, for this experiment, we have simulated the process of teacher evaluating (i.e. highlighting and commenting) essays. We used the benchmark data provided by the Hewlett Foundation at Kaggle[5] for an AEE competition. There are 10 data set,s corresponding to 10 different topics, in the benchmark containing student essays. All the data sets were rated by two human raters.

From these data, we randomly selected 2000 essays belonging to four different datasets, i.e. essay topics, for this experiment. Then, we randomly selected 6 different essays in which we manually highlighted a randomly selected sentence and generated a comment for this highlighted sentence. This simulates a teacher in evaluating 6 essays on different topics such that in each essay he/she comments one selected (highlighted) sentence. Then, we have searched the similar sentences in all the 2000 essays to those 6 sentences which were highlighted manually. We did it by a simple syntactic similarity search such that sentences which are 100% similar to those 6 sentences were highlighted and manually assessed by a human. The resulting set of 2000 essays each containing at least one highlighted text and a related comment serves as the experimental dataset for the proposed approaches. It is important to note that in the resulting dataset there are 6 different feedbacks corresponding to 6 different texts. We will denote these 6 sentences and related comments as “feedback one” (shortened as F1), “feedback two” (shortened as F2), . . . , “feedback six” (shortened as F6), respectively.

4.2 Data Preprocessing

In preprocessing an essay, the following tasks were performed: tokenization; removing punctuation marks, determiners, and prepositions; transformation to lower-case; stop-word removal and word stemming. In the stop word removal step, the words that are in the stop word list [12] were removed. After removing the stopwords the words have been stemmed to their roots [18] .

For essay evaluation, the freely available word2vec word embedding, which has an embedding for 3 million words/phrases from Google News trained using the approach in[17], was used as a word embedding model in the implementation of the proposed approach.

Python was used to implement the proposed semantic-based feedback recommendation algorithms and other base line algorithms discussed above. As the Relaxed Word Mover’s Similarity (RWMS) algorithm is dependent on a word embedding, we used the freely-available Google News word2vec[6] model. Also, Scikit-learn[7] and Numpy[8] Python libraries were used.

The performance of the proposed semantic based feedback recommendation system is compared to the rule-based and cosine similarity [7,27] approaches described in the Algorithms 1 and 2.

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

[6] `https://code.google.com/archive/p/word2vec/`

[7] `http://scikit-learn.org/`

[8] `http://www.numpy.org/`

4.3 Evaluation Metrics

To assess the successful decision-making capacity of the feedback recommendations algorithm and to evaluate the recommendation quality, we used classification accuracy metrics. Precision and recall are the most popular metrics used for evaluating information retrieval systems and recommender systems [21].

They are used to measure the amount of correct and incorrect classifications as relevant or irrelevant feedback that are made by the recommender system and are therefore useful for learning tasks such as finding good and relevant feedbacks in e-Testing systems.

Precision: (also called confidence in data mining) is a measure of exactness or fidelity and is calculated as the ratio of recommended feedbacks that are relevant to the total number of recommended feedbacks. This is the probability that a recommended feedback corresponds to the learners solution. A precision score of 100% would indicate that every recommendation retrieved was relevant [21,4].

If $AF = \{(h_a, c_a)\}$ denotes the set of actual feedback tags (i.e. highlighted texts and related comments) by the assessor and $RF = \{(h_r, c_r)\}$ denotes the set of recommended or predicted feedback tags by the recommender system, then Precision can be defined as in the Equation 3:

$$Precision = \frac{|AF \cap RF|}{|RF|} \tag{3}$$

Recall: (also called sensitivity in psychology) is a measure of completeness. Recall score of 100% would indicate that all relevant recommendations were retrieved. Recall is calculated as the ratio of recommended feedbacks that are relevant to the total number of relevant feedbacks [21,4].This is the probability that a relevant feedback is recommended and is defined as follows in equation 4:

$$Recall = \frac{|AF \cap RF|}{|AF|} \tag{4}$$

F1-measure: Since both Recall and Precision are important in evaluating the performance of a system which generates relevant recommendations, they can be combined to get a single metric, the F1-measure, which is a weighted combination of Precision and Recall [21,4], and is defined as follows in equation 5.

$$F1 - measure = \frac{2 * Precision * Recall}{Precision + Recall} \tag{5}$$

4.4 Experimental results and Discussions

Results in figure 1 show the comparison between the actual feedbacks and predicted (recommended) feedbacks. RWMS has an accuracy of 0.92, 0.97, 0.95, 0.99, 0.96 and 0.61 in correctly recommending F1, F2, F3, F4, F5 and F6, respectively.

The results of lexical based feedback recommendation algorithms have an accuracy of 0.12, 0.18, 0.44, 0.99, 0.98 and 0.58 in correctly recommending F1, F2, F3, F4, F5, F6, respectively, according to figure 2. According to figure 3 the rule-based feedback

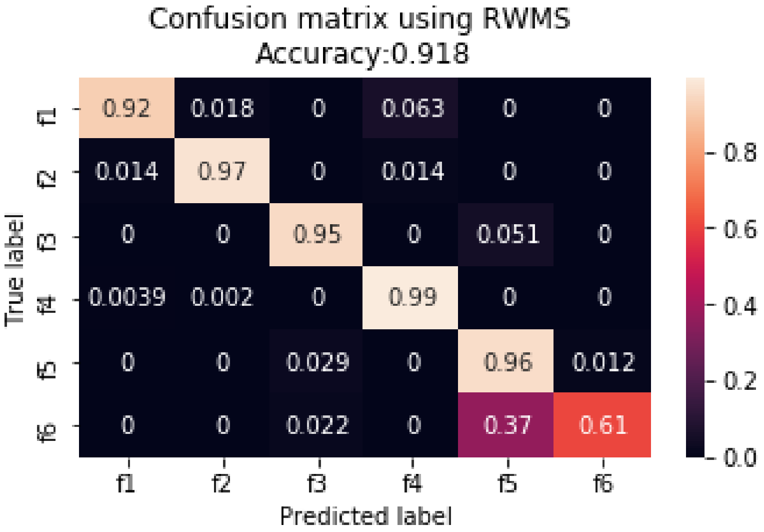

Fig. 1. Normalized Confusion matrix which shows the number of the cases from actual recommendation are correctly predicted(recommended) and how may of the cases are incorrectly predicted (recommended) by the RWMS algorithm

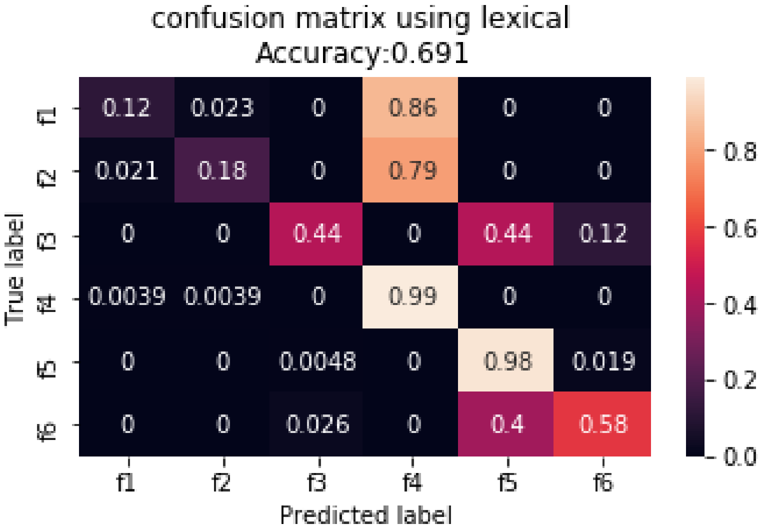

Fig. 2. Normalized Confusion matrix which the number of the cases from actual recommendation are correctly predicted(recommended) and how may of the cases are incorrectly predicted (recommended) by the Lexical based algorithm

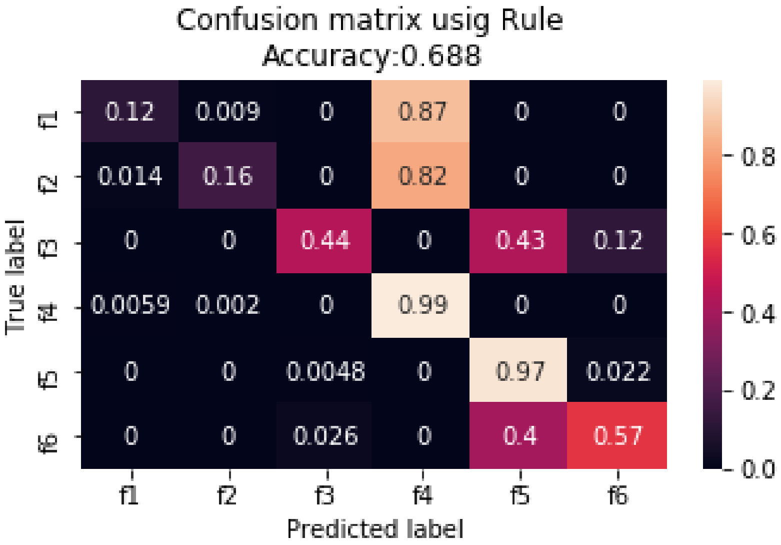

Fig. 3. Normalized Confusion matrix which show the number of the cases from actual recommendation are correctly predicted(recommended) and how may of the cases are incorrectly predicted (recommended) by the Lexical based algorithm

recommendation algorithms have an accuracy of 0.12, 0.16, 0.44, 0.99, 0.97 and 0.57 in correctly recommending the related feedbacks F1, F2, . . . , F6, respectively. Therefore, according to the the results of 1, 2 and 3 the proposed semantic based feedback tag recommendation algorithm has outperformed the baseline algorithms.

Table 1 shows the performance of the algorithms using Precision, Recall and F1-measure. Using the semantic based feedback recommendation, 92% of retrieved recommendation are relevant and 91% of relevant recommendation are retrieved while 74% of retrieved recommendation are relevant; and 69% and 68% of relevant recommendation are retrieved using rule based and lexical based algorithms respectively. The results in table 1 also show that the semantic based algorithm has outperformed the baseline algorithms.

Table 1. Performance of feedback tag based recommendation using Precision, Recall and F1-Score for dataset

Methods	Precision	Recall	F-Score
Semantic Based	0.925207	0.918042	0.914620
Lexical Based	0.742056	0.691038	0.635401
Rule Based	0.746979	0.688090	0.631034

In general, the results show that rule-based recommender algorithm and lexical-based recommender algorithm can work well in cases where there is a word by word mapping between a sentence s, from the set of all essays E submitted by all the students but not evaluated so far by the teacher, and a highlighted text h from the set $F = \{(h,c)\}$ of teacher's feedbacks present in already evaluated essays, i.e. comments c on some (parts of) texts h from students' essays evaluated by the teacher. Semantic similarity-based recommender algorithms can work by understanding the meaning behind them. That's the main reason for semantic similarity-based recommender algorithms to have a total accuracy of 91.8% while lexical-based and rule-based recommender algorithms have an accuracy of 69.1% and 68.8%, respectively.

5 Conclusions

Today, many e-Learning platforms offer authoring tools for e-Testing. These authoring tools allow assessors to create essay exams and the scoring will be done either automatically or manually. In most of the cases, the authoring tools do not have an option for the teacher to give feedback in the form of textual comments. To address such issues, we have successfully hosted the e-Testing system where the assessor can give feedbacks in the form of textual comments.

In order to make this feature more useful and helpful, a semantic-based feedback recommendation approach was proposed and implemented in this work. The proposed algorithm uses ground truth feedbacks from the assessor to give feedback recommendation to other similar student essay solution. To compute the semantic similarity between sentences, we used a relaxed word movers similarity distaance that computes semantic similarity based on neural word embedding.

The experiment was carried out on 2000 randomly selected essay from 10 different datasets which were provided by Kaggle for automatic essay evaluation and we simulate the role assessors to obtain textual feedbacks. The performance of the proposed approach was evaluated and compared to other state-of-the-art algorithms. According to the experimental results, the proposed approach has outperformed the baseline algorithms. In order to test the algorithms in real time scenario and to provide datasets for related research works, we will integrate the algorithm into our own e-Testing. Our aim is to publish real-world datasets which researchers can use in development of feedback recommendation algorithms and contribute to the given area of research.

Acknowledgements

The research has been supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.2-16-2017-00013).
Supported by Telekom Innovation Laboratories (T-Labs), the Research and Development unit of Deutsche Telekom.

References

1. M. Al-Smadi and C. Gtl. Soa-based architecture for a generic and flexible e-assessment system. In *IEEE EDUCON 2010 Conference*, pages 493–500, April 2010.

2. Yigal Attali. A Differential Word Use Measure for Content Analysis in Automated Essay Scoring. *ETS Research Report Series*, 36(August), 2011.
3. Yigal Attali and Jill Burstein. Automated Essay Scoring With e-rater® V.2. *The Journal of Technology, Learning, and Assessment*, 4(3), 2006.
4. Punam Bedi and Ravish Sharma. Ant-based friends recommendation in social tagging systems. *Int.J. Swarm Intelligence*, 1(4):321–343, 2015.
5. Kwangsu Cho and Christian D. Schunn. Scaffolded writing and rewriting in the discipline: A web-based reciprocal peer review system. *Computers & Education*, 48(3):409 – 426, 2007.
6. Mary K. Enright and Thomas Quinlan. Complementing human judgment of essays written by english language learners with e-rater scoring. *Language Testing*, 27(3):317–334, 2010.
7. A Ewees, A, Mohammed Eisa, and M. M. Refaat. Comparison of cosine similarity and k-NN automated essays scoring. *International Journal of Advanced Research in Computer and Communication Engineering*, 2014.
8. Peter W Foltz, Darrell Laham, and Thomas K Landauer. Automated Essay Scoring : Applications to Educational Technology. In *World Conference on Educational Multimedia, Hypermedia and Telecommunications (ED-MEDIA)*, 1999.
9. Gibbs G., Simpson C., James D., and Fleming S. Conditions under which assessment supports students learning. In *Learning and Teaching in Higher Education*, 2004.
10. Edward F. Gehringer. Electronic peer review and peer grading in computer-science courses. In *Proceedings of the Thirty-second SIGCSE Technical Symposium on Computer Science Education*, SIGCSE '01, pages 139–143, New York, NY, USA, 2001. ACM.
11. Khairil Imran Bin Ghauth and Nor Aniza Abdullah. An empirical evaluation of learner performance in e-learning recommender systems and an adaptive hypermedia system. 2010.
12. Pedro Hípola. G. Salton, Automatic text processing: The Transformation Analysis and Retrieval of Information by Computer. *Procesamiento de Lenguaje Natural*, 1991.
13. Michael Kane. Validating score interpretations and uses. *Language Testing*, 29(1):3–17, 2012.
14. Mohamed Koutheaïr Khribi, Mohamed Jemni, and Olfa Nasraoui. *Recommendation Systems for Personalized Technology-Enhanced Learning*, pages 159–180. Springer Berlin Heidelberg, Berlin, Heidelberg, 2015.
15. Matt J Kusner, Yu Sun, Nicholas I Kolkin, and Kilian Q Weinberger. From Word Embeddings To Document Distances. *International Conference on Machine Learning*, 37:957–966, 2015.
16. Jie Lu. A personalized e-learning material recommender system. In *International Conference on Information Technology for Application*, page 374379, 2004.
17. Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositional. In *NIPS*, 2013.
18. Tsegaye Misikir. Stemming algorithm for awngi text: A longest match approach, 2013.
19. Ellis Batten Page. Grading Essays by Computer: Progress Report. In *Invitational Conference on Testing Problems*, 1966.
20. Niels Pinkwart, Vincent Aleven, Kevin Ashley, and Collin Lynch. Evaluating legal argument instruction with graphical representations using largo. In *Proceedings of the 2007 Conference on Artificial Intelligence in Education: Building Technology Rich Learning Contexts That Work*, pages 101–108, Amsterdam, The Netherlands, The Netherlands, 2007. IOS Press.
21. Guy Shani and Asela Gunawardana. *Evaluating Recommendation Systems*, pages 257–297. Springer US, Boston, MA, 2011.
22. Mark D Shermis and Jill. Burstein. Automated essay scoring a cross-disciplinary perspective. *British Journal of Mathematical & Statistical Psychology*, 2003.
23. Tsegaye Misikir Tashu and Tomas Horvath. Pair-wise: Automatic essay evaluation using word movers distance. In *Proceedings of the 10th International Conference on Computer Supported Education - Volume 2: CSEDU,*, pages 59–66. INSTICC, SciTePress, 2018.

24. Nguyen Thai-Nghe, Tomás Horváth, and Lars Schmidt-Thieme. Factorization models for forecasting student performance. In *EDM*, 2011.
25. Andrew Walker, Mimi M. Recker, Kimberly Lawless, and David Wiley. Collaborative information filtering: A review and an educational application. *Int. J. Artif. Intell. Ed.*, 14(1):3–28, January 2004.
26. PAIGE WARE. Computer-generated feedback on student writing. *TESOL Quarterly*, 45(4):769–774, 2011.
27. Peipei Xia, Li Zhang, and Fanzhang Li. Learning similarity with cosine similarity ensemble. *Information Sciences*, 2015.