Web Semantics: Science, Services and Agents on the World Wide Web 72 (2022) 100700

Contents lists available at ScienceDirect

Web Semantics: Science, Services and Agents on the World Wide Web

journal homepage: www.elsevier.com/locate/websem

A hybrid E-learning recommendation integrating adaptive profiling and sentiment analysis

Hadi Ezaldeen [a,*], Rachita Misra [b], Sukant Kishoro Bisoy [a], Rawaa Alatrash [a], Rojalina Priyadarshini [b]

[a] Department of Computer Science and Engineering, C.V. Raman Global University, Bhubaneswar, Odisha, 752054, India
[b] Department of Computer Science and Information Technology, C.V. Raman Global University, Bhubaneswar, Odisha, 752054, India

ARTICLE INFO

Article history:
Received 18 December 2020
Received in revised form 8 October 2021
Accepted 17 December 2021
Available online 24 December 2021

Keywords:
Hybrid E-learning recommendation
Adaptive profiling
Semantic learner profile
Fine-grained sentiment analysis
Convolutional Neural Network
Word embeddings

ABSTRACT

This research proposes a novel framework named Enhanced e-Learning Hybrid Recommender System (ELHRS) that provides an appropriate e-content with the highest predicted ratings corresponding to the learner's particular needs. To accomplish this, a new model is developed to deduce the Semantic Learner Profile automatically. It adaptively associates the learning patterns and rules depending on the learner's behavior and the semantic relations computed in the semantic matrix that mutually links e-learning materials and terms. Here, a semantic-based approach for term expansion is introduced using DBpedia and WordNet ontologies. Further, various sentiment analysis models are proposed and incorporated as a part of the recommender system to predict ratings of e-learning resources from posted text reviews utilizing fine-grained sentiment classification on five discrete classes. Qualitative Natural Language Processing (NLP) methods with tailored-made Convolutional Neural Network (CNN) are developed and evaluated on our customized dataset collected for a specific domain and a public dataset. Two improved language models are introduced depending on Skip-Gram (S-G) and Continuous Bag of Words (CBOW) techniques. In addition, a robust language model based on hybridization of these couple of methods is developed to derive better vocabulary representation, yielding better accuracy 89.1% for the CNN-Three-Channel-Concatenation model. The suggested recommendation methodology depends on the learner's preferences, other similar learners' experience and background, deriving their opinions from the reviews towards the best learning resources. This assists the learners in finding the desired e-content at the proper time.

1. Introduction

Online learning proves its effectiveness in various educational levels and subjects nowadays, especially in critical circumstances where traditional education suffers from many difficulties. With the development of knowledge technology, academic institutions are adopting new educational strategies and means to further virtual learning effectively [1]. Personalized recommendation is one of the most prominent strategies for accelerating the pace of e-learning. The best way to apply the optimal learning path is through adaptation methodologies that can answer many questions such as: What courses should be used to achieve this goal? Who is a learner needing the targeted content? What content properties accomplish the desired results and develop the learning procedure? [2].

Most of the existing web-based recommendation systems mainly fall into the following primary classifications; (A) Collaborative Filtering Recommendation, (B) Content-Based Filtering Recommendation, and (C) Hybrid Recommendation that tries to overcome problems of "Cold-start" and "Sparsity" from which the other two recommendation approaches suffer. Precisely, when the extensive historical data required by the system has not yet been gathered to draw any inferences for users or items based on sufficient information, the problem of Cold-start happens with this. Also, when user feedback data is sparse and insufficient for identifying sufficient reliable similar users, the problem of Sparsity occurs. The hybrid recommender system is a synergy of multiple techniques using different inputs to recommendation service where every single technique is complemented, specifically when applied to vast web-based platforms [3].

* Correspondence to: Department of Computer Science and Engineering, C.V. Raman Global University, Bidya Nagar, Mahura, Janla, Bhubaneswar, Odisha, 752054, India.

E-mail addresses: hadi.talal@gmail.com (H. Ezaldeen), rachita.dhunu.misra@gmail.com (R. Misra), sukantabisoyi@yahoo.com (S.K. Bisoy), rawaa.alatrash@gmail.com (R. Alatrash), priyadarshini.rojalina@gmail.com (R. Priyadarshini).

https://doi.org/10.1016/j.websem.2021.100700
1570-8268/

An effective e-learning recommender system explores the semantics among textual e-content. In this context, learner profiling is the scope of adding personalization and customization to instructional platforms. The learner model is an essential component of an adaptive e-learning system due to its ability to represent learner characteristics using semantic knowledge under which the learning system can make better recommendations [4]. More details have been presented in the literature study in Section 2.

The content recommender system can provide the ability to classify learning elements using attribute values included in the learner profile [5]. Moreover, analyzing learners' feedback by associating specific sentiments latent in a part of text with different classes of a learning resource produces meaningful semantics in the educational context towards adaptation and personalization. Sentiment analysis is considered a text mining technique [6] to systematically identify the sentiments attributed to each learning resource and categorize them by rating to fuzzified sets. It thus contributes to achieving the desired outcomes in different learning scenarios.

Humans by nature are influenced by sentiments impacting motivation which drives a person's behavior and supports decision making. Since people express their opinions, sentiments, and attitudes in a text form on the web, the sentiment analysis application aims to build a model that analyzes what the individual wants of service, subject, or product through extracting affective information from a huge amount of data obtainable online [7]. Affective computing is an emerging interdisciplinary research area of sentiment analysis capable of capturing public sentiment automatically from user posts or reviews [8]. However, automated text-based opinion mining is a difficult process, but in practice, it is very useful. Affective computing research has increasingly evolved from traditional unimodal analysis to more complex forms of multimodal analysis [9]. However, the volume of data in social media repositories and instructional platforms is growing exponentially; hence, conventional algorithms struggle to extract sentiment from such big data uploaded [10]. Deep Learning (DL) has emerged as a powerful machine learning technique that resulted in state-of-the-art findings since a decade ago ranging from computer vision and speech recognition to Natural Language Processing (NLP) [11].

With the assistance of recent studies, sentiment analysis methods are viewed from two perspectives [12]. The first is based on the granularity of text analysis, again divided into three levels (Document level, Sentence-level, and Aspect-level). The second one relies on the method of operation can be divided into knowledge-based and machine learning-based approaches. Apart from purely one approach, some models combine techniques from both fields. While the learning methods based on sentence-level are to classify the sentiment per sentence, the key issue is defining the expression in the sentence whether it includes the entity under scrutiny (object) or contains explicit words that represent the feeling towards that entity (personal). However, just knowing the polarity of the sentence is not enough. Therefore, multi-class affective classification of expression is the main challenge for deep neural models.

As such, the resources and primary content available are restricted to text-based sentiment analysis. Hence, most of the research work has thus far fallen into the scope of NLP [13]. Presently, one of the major bottlenecks in the sentimental classification task is how to accurately represent the semantics of the natural language that refers to the user's intention. That is, some words appear irrelevant to the context, but they directly influence the classification model's performance. Word embedding methods are a particularly prevalent area in NLP which may be considered one of the critical breakthroughs of DL and its challenges in language modeling and feature learning capabilities [14]. Thus, using Deep Neural Networks (DNNs) with NLP can lead to instrumental breakthroughs in several domains as detailed in Section 2.

The current framework uses knowledge-based techniques to provide the ontological and lexical knowledge required for term expansion, and thus to generate the semantic matrix used for automatic learner profiling [15,16]. Integration is done on powerful fine-grained sentiment analysis models based on customized Convolutional Neural Network (CNN) and NLP techniques that have a superior capability to automatically learn multiple layers of features representations of a huge amount of big data. Hybridization methods have been proposed to improve the accuracy of recommendations. To some extent, using the combination of such methods can tackle the Cold-start problem of a recommendation system [17].

Ultimately, it is desired to link the conceptual learning context to the learners' requirements semantically and depending on deep analysis. In the current work, the following contributions have been accomplished:

- Proposed a novel e-learning hybrid recommender system framework to select and recommend an appropriate e-content according to the learner's particular needs.
- Suggested a new approach to automatically build the semantic learner profile based on significant rules and semantic text representation using WordNet and DBpedia ontologies to expand user terms. The learner profile is updated according to learner behavior and different browsing actions.
- Tailored constructions of CNN and the hybridization of a couple of NLP methods are developed by thirteen CNN-based models of fine-grained sentiment analysis to predict the ratings from the textual reviews about a particular learning resource as part of the proposed recommender system.
- A personalized recommendation methodology is introduced depending on the dynamic learner profile to provide e-learning resources with the highest predicted ratings relevant to the learner's domain and preferences.
- Top findings of our experiments come with multi-classifying the reviews of e-learning resources in a range of five classes using our customized dataset created for a specific domain with fine-grained sentiment labels [1–5].

The study framework centered on the fine-grained sentiment analysis from e-content reviews and the construction of the semantic learner profile. The objective is to improve the efficacy of the hybrid recommendation approach to make recommender systems for educational purposes smarter.

This positively affects the learners and is reflected in their attitudes and selections. What matters in this work is that the stimulated paradigms and practical advice have been derived from the extensive experiments in getting the best empirical results.

A detailed literature study with an analytical review of the existing adaptive e-learning system and the various techniques used has been presented in Section 2. The proposed system is introduced in Section 3. Experiments, results and discussion are given in Sections 4 and 5, respectively. Section 6 has some future work directions along with the conclusion of this current work.

2. Related works

This section presents state-of-the-art studies conducted in the adaptive e-learning domain regarding learner modeling methods, reviews the literature with new trends on sentiment analysis and comparative analysis based on the nature of work carried out.

2.1. Integrating the systematic inferring of learner model into adaptive E-learning system

Technological progression has created a growing interest in exploring the learner's behavioral data during learning procedures to provide process-oriented feedback in several analytical forms, such as; underlying descriptive traits, learning patterns, pedagogical context [18]. Numerous researches in adaptive learning strategies have been aligned with various studies on user profiling models, development, implementation, and evaluation [19, 20]. For such strategies to adapt the e-learning, researchers deliberated both approaches; adaptive objectives based on learning paradigm and content; and adaptive sources based on learner model.

Learner's characteristics have been observed primarily on learning patterns. Mostly, the learner model is inferred by investigating navigation and concluded information to align the recommendation targets. The learner model incorporates various components like learner's preferences, competence and knowledge level, sentimental and motivational factors, and other individual features or specific differences to build a learner profile tailored to the learning domain [21].

Research on discovering a personalized learning full-path to enhance advanced e-learning platforms is precious in education technology, where the learner data is relied on to discover advantageous hidden patterns. This study [22] has proposed a full-path based learning recommendation model using clustering techniques and learning resource datasets. Long Short-Term Memory (LSTM) model has been trained based on feature similarity metric on learner collection to predict the learning paths. The recommendations are provided on appropriate learning full-path. More learning paths can be automatically generated from the learners' data streams to enhance their work.

An adaptive learner profiling approach has been presented in [23] to classify the learners depending on their current state. Their proposed learner classification system uses characteristics like the learners' performance, knowledge, and preferences to provide suitable learning content. Decision Tree and Bayesian Belief Network algorithms have been used to update learner profiles. Increasing learner performance can be by discovering the appropriateness of the level of learning content provided, which can be derived from learners' opinions and behaviors. Further, in the context of enhancing a learner-centered e-learning environment, a classifier based on captured data from learners' learning behaviors has been proposed to identify the learning style [24]. The learning behavioral data regarding several courses have been resourced by the web mining and then clustered using Fuzzy C-Means algorithm into the categories of Felder–Silverman Learning Style Model (FSLSM) [25]. Despite this, their approach does not automatically adapt to learners' choices that rely on their learning patterns.

In such an aspect, a new methodology that characterizes learner behavior has been framed within an e-learning recommendation system [26] to produce services based on the learner's preferences through their courses. The learner profile has been modeled dynamically based on learner behavior. Other similar student learning styles and ratings should be considered when updating the learner profile to serve a more appropriate level of study content. Likewise, as part of the personalization process, the learner's behavior has been merged with educational skill tests within an adaptive learning management system [27]. Personalized Page Rank algorithm has been employed depending on the learners' performances and classifications using Navies Bayes classifier. However, the collection of the learning patterns should be more extended related to additional types of learning activities and objects.

To effectively use the ontology representation and adaptive architecture, the authors in [28] attempted to integrate both to improve the strategy assisted by cloud storage and variant collaborated agents to introduce personalized e-learning. Their proposed framework relies on concepts of learning styles and learner specifics included in user ontology and the concepts of e-learning objects in course ontology. This ontological approach is limited to user and course. Further, taking into account the exact value of a specific dimension in 'Felder–Solomon learning style index' used can objectify the learner's personalization in a better way.

A multitude of learner modeling approaches based on ontologies was proposed that describe a better way a learner prefers to learn. A semantic web ontology-based framework has been proposed in [29], focusing on the personalization of e-learning by learner modeling and recommending convenient learning objects. The e-learning ecosystem consists of four layers to construct the ontologies that represent all the components more explicitly. However, the learning environment can be much more intelligent by creating ontology reasoning and instances. The adaptation rules require being more flexible to add new resources that support particular learning styles.

A semantic retrieval of information on student's characteristics is achieved by integrating ontology and fuzzy concepts to construct the learner profile for the recommendation of learning content [30]. The learner profiling model relies on retrieving learner preferences utilizing ontology-based semantic similarity with WordNet. Learning style is inferred using Decision Tree algorithm with significant rules, depending on FSLSM model.

The core objectives of the existing studies have been aligned with learner modeling, whether considered technology or a process, to better define criteria according to a comprehensive view [31]. The shortcoming of the existing e-learning systems become evident, specifically when performed in instructional environments wherein the learner characteristics are diverse. Ignoring individual traits and feedback eventually creates an unchanging (static) learner profile. This, in turn, may prevent the system from achieving individualization and providing a competency-based approach in education. Whatever the ontology-based methods have been presented in this domain, it needs to be evolved day-to-day and get adapted concerning the context, usage, and semantic representations through improved modality to personalize the services.

2.2. DNNs space with NLP techniques for sentiment analysis

Other people's sentiments influence our decisions by virtue of the information they convey. The analysis of personality sentiment is one of the most popular NLP applications. It has achieved noticeable results on automated mining of sentiments and opinions from the texts [32]. Particularly, with a combination of DL techniques such as CNN [33] which is originated in computer vision and then used in the field of search query retrieval, sentence modeling, semantic parsing, and other NLP missions including emerging sentiment analysis trends.

In a broad view, sentiment analysis can be done on a document-level [34] where the sentiment is detected for the entire text, a topic-level to identify the topic sentiment [35], or on a sentence-level [33,36] that produces the sentiment per sentence. Also, based on the aspect-based level [37] so that the sentiment is computed for each aspect of the reviewed aspects in a text.

Numerous studies in the literature have focused mainly on completely machine learning-based or knowledge-based techniques, whereas other researchers investigated the use of hybrid approaches for sentiment analysis via affective computing [38]. Consequently, multiple supervised feature-driven approaches using learning models such as Naive Bayes, Decision

Tree, Support Vector Machine, Random Forest, Gradient Boosting, Logistic Regression, Multi Layer Perceptron [39], and Bayesian Networks [40] have become well-known for the classification task of texts. Moreover, different approaches have been developed to acquire even more detailed insight into people's sentiments such as the unsupervised semantic orientation approach [41], a weakly-supervised approach based on DNNs [42], a multitask learning-based approach using attention-based DNNs [43,44], Self-supervised attention learning-based approach using DNNs [45]. Moreover, recent studies have elaborated on various approaches and techniques to evaluating online courses via reviews mining as introduced in [46].

In contrast, knowledge-based approaches exploit the presence of reasonably clear affect words to classify text into affect categories using lexical sources of affect concepts such as linguistic annotation scheme, SentiWordNet, SenticNet [47], and other probabilistic knowledge bases trained from a linguistic corpus. When linguistic rules are involved, the main flaw of knowledge-based methods is their inability to identify the sentiment. Another drawback of knowledge-based methods is the typicality of their knowledge representation which is often rigidly specified and does not allow for handling nuances among many concepts. There is a lack of sentiment ontology despite the rise of affective computing and related disciplines. More recently, the provided representation of OntoSenticNet 2 supports the fusion of reasoning engines able to deduce implicit sentiment information, unlike purely syntactic techniques [48]. In contrast to methods based on attentive neural networks and word embeddings, a two-stage hybrid model using the semi-automated ontology builder is implemented to improve aspect-based knowledge-driven sentiment analysis [49].

Deep neural models usually learn in a task-specific neural space rather than to depend on any human-defined symbolic representations wherein low dimensional continuous vectors are used to define semantic concepts represented by the task-specific knowledge implicitly [50]. Two-dimensional CNN has been implemented in different fields for improved results, such as NLP with the new trends of fine-grained sentiment classification of e-content reviews [51]. Many design decisions of DNNs are to identify what is essential or relatively inconsequential with respect to relative simplicity and powerful performance considerations.

With the rapid development of DL, recent advances in sentiment analysis try to predict the affective state of text that pertains to different aspects or interpretations using attention-based hybrid neural networks [52]. In this aspect, Meskele and Frasincar proposed hybrid approaches leveraging neural models based on attention mechanism leveraging lexicalized domain ontology and neural attention models for sentence-level aspect-based sentiment analysis [53]. Moreover, a knowledge-based approach is provided to retrieve the classes the keywords belong to. The content is examined by a hybrid, bidirectional LSTM coupled with convolutional layers and an attention mechanism that outputs the final textual features [54]. Alleviating the accumulation of errors in the pipeline method is a proposed hybrid framework for multiple perspective attention based on double BiLSTM. This uses a new joint strategy for an aspect and sentiment pair extract [55]. Another hybrid neural network for aspect-based sentiment analysis is designed based on an aspect-attention mechanism and a self-attention mechanism addressing issues like information loss that occurs with recurrent neural networks [56]. The proposed models try to improve the semantic representation of sentences at two auxiliary features of part-of-speech and word location.

The simultaneous use of symbolic and subsymbolic AI for knowledge representation and reasoning demonstrates the transition from mono-to-interdisciplinary knowledge representation and reasoning [47]. Recently, ensemble application of Symbolic and Subsymbolic AI tools are integrated within top-down and bottom-up DL architecture based on logical reasoning to build a commonsense knowledge base (SenticNet 6) for sentiment analysis which is applied to polarity detection tasks from text [57]. A couple of distributed and symbolic executors have been trained in step-by-step fashion for better interpretability and accuracy [58]. Symbolic methods are sensitive to paraphrasing and hard to train. Conversely, attention neural models can permit sound paraphrase alternations and are trained in end-to-end fusion, except for limitations about explicit interoperability and implementation effectiveness attributed to state-space and performance.

The most open problem in processing natural language texts is the ambiguity (polysemy) for single- and multiple-word units. This affects all levels and is expressed in the phenomena of polysemy, homonymy, and synonymy. As noted, such types are lexical, syntactic, structural, semantic, pragmatic ambiguities as described by Zhang and Teng [59] in detail. In many cases, a type of heterogeneity occurs, and the set of morphological characteristics is not sufficient to explain it. Ambiguity reduction can be tackled by semantic analysis and parsing based on statistical methods to discard extremely improbable options. Although the natural language is intrinsically symbolic, it is rather difficult to process it using symbolic and objective models and logic-based rules. Since it is changeable, complex, and inherently ambiguous, statistical algorithms have been used in the dominant approaches for its processing based on statistical deep machine learning [60].

Furthermore, word embeddings are used to create a distributed representation in a multi-dimensional feature space of either words or characters that grab the semantic relations between them. Recently, researchers have reported the efficiency of using word embedding models on online resource reviews, such as Word2Vec [61,62], and FastText. Many neural network architectures are initialized using different word embeddings models that differ the way vectors are extracted out of vocabulary. One way is the random initialization of the word representations by setting them using random integers. The second way is to generate from general-purpose with utilizing static pre-trained word vectors and then fine-tuning them for each epoch. More features (from document, sentence, and word) can be extracted to enhance the efficiency of pre-trained embedded vectors. In this direction, suffixes such as emotion, subjective words, the total number of syllables, number of characters with or without punctuation, and part-of-speech (POS) tag can be appended to the word vector [63].

Designing sentiment analysis models by combining DL constructions and word embedding representations has led to more accurate results on measuring the text polarity in several contexts. Although such a model using these constructions and representations made on educational data still appears limited. The embedding model may lack some representations of words that are out-of-vocabulary (OOV) words unseen in the training phase. This may lead to some extent of mishandling in the complex context which results in overfitting. For such cases, approximated solutions are used to handle the OOV problem of pre-training word vectors. In particular, language modeling can perform the sequence of words in the given sentence corresponding to the OOV word and then compares it with similar sentences to predict the meaning of the word. Also, word embeddings can be trained on the corpus from scratch. Alternatively, word embedding models can be performed at two levels (character and word embeddings) [64]. However, the model's performance can be evolved if word embeddings are created for a specific domain.

In addition, when addressing NLP tasks, it is considered to pad the sentences with meaningful semantic vectors. Recently, Gimenez et al. studied the effectiveness of the semantic padding methodology applied in CNN [65] where the unused space is padded to fix the input size matrix utilizing words that existed in the sentence through three padding techniques such as "Roll, Random, and Loop". However, the internal representation can be better learned by the network, so that improves the network's interpretability.

Various CNN models have been developed by Kim [33] for sentiment analysis on sentence-level using pre-computed word embeddings "word2vec" to vectorize different public datasets. However, the series of experiments that have been carried out in the field of sentiment analysis have reported better results for binary classification rather than fine-grained sentiment classification which is the grand challenge where this current work occupies a fitting place. DL-based approach for sentiment analysis was presented in [66] to define feelings polarity of learner's reviews after completing various online courses. Variant DL architectures combined with word embedding representations have been trained on smaller and e-learning private resources. These models evidence how specific word embedding representations are more effective than training on a general-purpose corpus.

It becomes necessary to unlock the full potential of sentiment analysis models made in the context of adaptive e-learning. Consideration should be given to the particular area in which training data is collected and the over-sensitiveness of prominent models in which personalized e-learning approaches have been implemented. Moreover, these models need to specify model parameters such as network pattern, hyper parameters, feature size, and regularization. Here, it is imperative to consider the sensitivity analysis that is impacted by the components of the structure. The performance of the model also depends on the changes in the configurations to classify the sentence.

A powerful learning system needs to have an adaptive process using hybridization methods initiated by the learner model and dependent on instructional requirements. Coveted outcomes are achieved when day-to-day learning patterns are integrated with the model, providing a clear competitive edge for e-learning platforms.

3. Proposed enhanced e-learning hybrid recommender system

Selecting suitable online educational materials from the e-learning service provider according to the learners' needs and preferences is complex and challenging. The motivation has been derived from [30] the prediction of the learner profile which relies on retrieving learner preferences utilizing ontology-based semantic similarity. Further, investigating the particular solutions to acquire deep seeing into learners' reviews express their perspectives and opinions on the learning resources [66]. In addition to striving to realize, the most advantages of effective SA-based CNN architectures for fine-grained sentiment classification in real-world settings [33].

In this work, the Enhanced e-Learning Hybrid Recommender System (ELHRS) relies on the learner behavior and semantic analysis of e-learning objects in order to deduce the semantic learner profile automatically as described in Section 3.1. Sentiment analysis models based on CNN are employed for opinion mining from the reviews of learners engaged in the learning environment and exploring their feedback regarding the best resources is introduced in Section 3.2. Learner preferences are fed into ELHRS to recommend relevant e-learning resources that are obtained best reviewed as the recommendation methodology is depicted in Section 3.3.

Fig. 1 illustrates the proposed framework architecture of ELHRS in which four levels are involved. Level 1 denotes the learner profiling model and the data obtaining process. Level 2 shows how CNN input and output have been handled based on different training parameters, as both Levels (2 and 3) form the training process of the CNN-based model. The personalized e-content recommendation process is performed in level 4.

3.1. Automatic building model of semantic learner profile

Learner profiling is the process of identifying the learner to reach them by achieving more levels of diversity in the e-learning environment to know their targets and why they browse the web portal. Learner profile is related to his/her actions on the web that reflect the interests which comprise a set of information about a specific learner. Learner profile is built in a way that captures the semantics of interaction between users and learning resources. Thus, learners are categorized according to the domain of learning based on behavior. Each learner is linked to a set of e-learning categories, and this process can be done automatically using a user's historical data.

Accordingly, the learner model is identified as a vector of relative relations to e-learning category features. The proposed approach is to build the learner profile which depends on an implicit way to help learners find the relevant materials as illustrated within the architecture of Level 1, Fig. 1.

Learner behavior in online mode adequately expresses his/her characteristics and reflects his interests. Intuitively, a learner who frequently surfs specific material from a particular website section mostly belongs to a similar domain. The model defines the learner's profile as a set of values corresponding to the learner's characteristics. Each value expresses the extent to which it is confirmed that the learner possesses a particular characteristic or belongs to a particular domain. These values are given by:

$$F = \{f_1, f_2, \ldots, f_\mathrm{n}\}$$

Here, f_i denotes that the learner belongs to a specific e-learning category. In this case, the e-learning categories are indicated scientific terms such as "ML" can be referred to f_1, and f_2 would denote "IoT" and so on.

That means a learner profile Pu for a learner u is a set of values $\{v_1, v_2, \ldots, v_n\}$ where v_i denotes the certainty of learner u is linked to f_i with this probable value. It would be such as:

$$Pu = \{0.849, 0.55, \ldots\}$$

That indicates the learner has the characteristic "ML" with the value $v_1 = (0.849)$, and the characteristic "IoT" with the value $v_2 = (0.55)$. In the learner profile model, each characteristic takes a certainty value within the range [0, 1]. The value zero denotes complete uncertainty whether or not the learner has the characteristic f_i, whereas the value '1' denotes outright certainty of learner having the characteristic f_i and vice versa.

As indicated above, the learner profile links to the learner's actions on the web. Hence, it can redact the previous fact as follows:

If the learner u does an action b, he will likely have a confirmation characteristic (feature) f_i with a certain value, then he belongs to a specific domain (term). The possible learner's actions (behaviors) are represented as a set B:

$$B = \{b_1, b_2, \ldots, b_n\}$$

For example, the action b_1 would be surfing a specific material from ML domain. b_2 would be surfing a specific material from IoT domain, and so on.

The dynamic learner profile depends on semantic relation between the learner term (learning category) and the action b_i that is surfing scientific material from this category.

Based on that, a function P is defined as:

$$P : BxF \rightarrow [0, 1]$$

Formally, $P(f|b)$ is the certitude of a learner who does action b has the feature f. The proportional value of $P(f|b)$ is within the range [0, 1] as it is mentioned above. $P(f|b)$ values are computed by extracting the mutually semantic relations between the

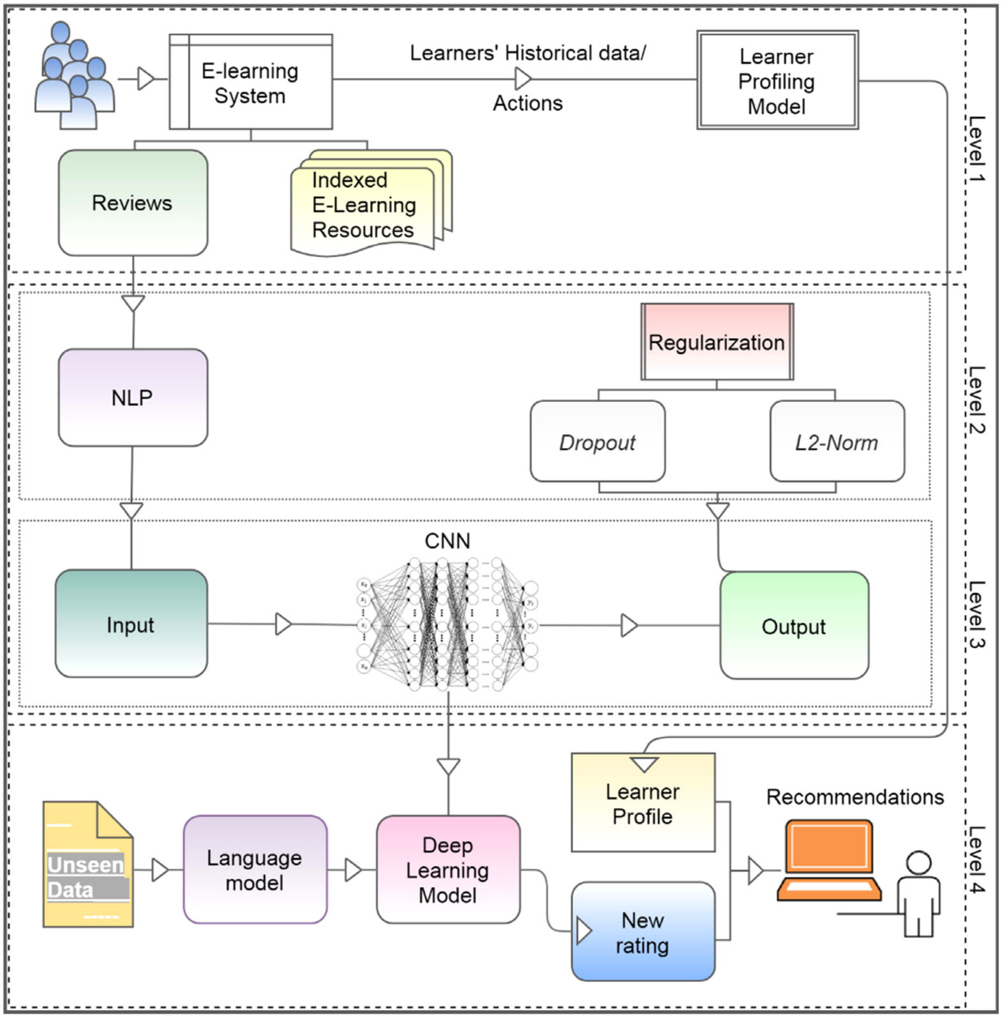

Fig. 1. Architecture of ELHRS.

terms and the learning materials in the e-learning system using ontology-based semantic analysis. The computation of $P(f|b)$ values form the 2-dimensional semantic matrix (term X material) that utilizes DBpedia and WordNet ontologies for expanding the learner's terms linked to the e-learning categories as described in Section 3.1.1.

Suppose the current probability of the learner interest in ML is v_i and the learner has surfed "Python Programming", this relative value should thus change. Let this change value be G. Essentially, G is related to the original v_i because the higher the probability, the greater the certitude ratio is unless the original probability ratio is zero. Also, it is related to $P(f|b)$ because the change ratio must be related to the coefficient of change. But, v_i cannot exceed a specific maximum value which is the sum of all the posts submitted by all actions associated with f. To formulate that, $'Max'$ is defined as follows:

$$\text{Max} \begin{cases} if \sum_f P(f|b) \geq 1, & then\ Max = 1 \\ Else, & Max = \sum_f P(f|b) \end{cases}$$

The increase on v_i must be proportional to the distance from Max, and thus G will also become proportional to $(\text{Max} - v_i)$. It means the farther we are from the maximum value the greater is the increase (G value).

To compute the values f_i of the features in the learner profile mentioned above, it depends on what the learner has done online (actions) maybe surfing specific materials that belong to a specific term, considering the semantic relationship between the term and e-learning resource. In the learner profile, if this learner u has a value v_i for a feature f_i, v_i will be affected and changed by doing a series of actions b_i associated to f_i, as follow:

If $b_i = \phi$ (when b_i is empty)
then $v_i = 0$
else,
initialize $v_{i=1} = P\,(f|b)\,;$
for $(i > 1$ to $n)$

$$v_i = v_i + G; \tag{1}$$

end of for
end of if

Here, G indicates the change of value v_i after the learner has done the behavior (b_i), which is computed as follow:

$$G = |v_i * \ (Max - v_i)| * P\,(f|b) \tag{2}$$

It must be considered that the value of change of some feature is to be achieved by two conditions:

(a) The proportionality should not be reversed between the two values v_i and G via the component v_i of Eq. (2), this condition is to be attained.

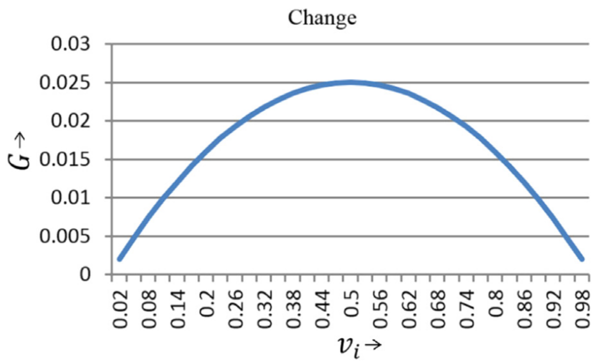

Fig. 2. Amount of change (G) with respect to feature value (v_i).

(b) The value v_i should not overtake the range [0, 1]. This condition is to be attained by the component ($Max - v_i$) in Eq. (2).

It is noted that the change achieves a Gaussian distribution to make the change when the ratio is very large or very small. The closer the ratio is to the medium the lesser the change is, and this helps to maintain non-extreme mean values of the learner's characteristics. As Fig. 2 visualizes G that is applied to v_i, whenever the value v_i is going to be larger than 0.5, the value of G could be obtained smaller, and v_i could be reached the certitude after a significant number of behaviors which the learner has done.

In fact, one material may be involved in more than one term. In this way, the value of features indicated these terms to which this material belongs should change as explained above.

The greater the learner selects books belonging to a particular category, the greater his interest in this group. However, increasing the learner's interest in a specific category should be significantly associated with decreasing interest in some categories. So, to keep the updated learner profile updated and corresponding to the recent behaviors, the oldest domain with certitude value v_j for which the user has not been surfing its materials anymore and has the same certitude value v_i is searched. Thus, for breaking tie-up values for the same, the domain in which v_i has been modified by adding G_i value, and decreasing v_j by the same amount of increased value G_i, as follows:

$$v_j = v_j - G_i \tag{3}$$

3.1.1. The semantic-based approach for term expansion to compute the semantic matrix

The automatic learner profiling model uses the probability values $P(f|b)$ as input semantic data for building the dynamic learner profile. This needs to effectively compute the semantic matrix linking terms to materials mutually which comprises $P(f|b)$ values as mentioned above.

Since the model depends on the choices of the learner and the semantic analysis of the content, accordingly e-content is collected from various sources available on the e-learning platform. The collected content of e-learning materials is processed. Following that, the content of each material is searched for entities, and the content is then indexed in the system. Based on this, the extracted entities (terms) and the linked e-content documents are stored. Thus, the initial terms are matched with the e-learning categories [67].

The goal is to leverage the semantic relations between the learning material content and terms to discern how likely this term would be chosen if this lesson was chosen. This will enhance the suggested semantics-aware learner profiling model to recommend closely related e-content and increase the precision of search results.

For accomplishing this, it is needed to compute the probability of a term given the specific material surfed by the learner to find $P(f|b)$, as mentioned above that is f indicates the original term and b indicates the learning material. This can be achieved by finding the best semantic similarity of the learning material content with the original term and specific series of its expanded concepts.

The traditional recommendation engines may apply some methods according to a guiding principle, such as replacing a word with its synonyms. The goal of the system is to determine how similar the content is to the original term regardless of the presence of the term and its synonyms within the documents. It is needed to look for the set of related words to the original term inside the indexed documents. Therefore, the ontology-based semantic similarity approach is proposed to expand the initial terms using DBpedia knowledge base [68] and WordNet lexical database [69].

The knowledge-based approach for term expanding with concepts extracted from a global ontology and getting semantically weighted using WordNet considerably improves information retrieval in the recommender system. Especially when e-learning material needs to be semantically matched with terms, it is more efficient to add terms from the domain rather than adding terms that statistically occur with the original terms or adding terms that are lexically similar to the original term. The ontological and semantic knowledge from DBpedia and WordNet are explored to enrich the term expansion by adding concepts more related to the original concept. This enhances the computation of the semantic relation matrix linking initial terms to materials, by which the sparsity problem can be quite tackled.

Here, DBpedia is used to find a set of related concepts for a word as it is a big source of shared knowledge. It is a very intensive RDF graph based on the linked-data style containing concepts from different scientific domains. It uses SPARQL query language for its queries to obtain the adjacent concepts of the specific term where this term is represented in the query by a single word.

The starting point is to determine the concept of each initial term by using DBpedia and subsequently determining all adjacent concepts relevant to the concept domain.

For illustration purposes, consider having a single word regarded as a term so that the commence is to find the concept in DBpedia that represents this word. It can be done simply by using the lookup service provided by DBpedia. By using this service, the literal words can be converted into URI of concepts in DBpedia. If the word is not found in DBpedia, the list of related words will be empty, and the original word is kept to look for it.

DBpedia is represented with tuples of data by the pattern (subject, property, and object). All tuples are searched in which the subject is the target concept, then all objects from these tuples are extracted. Thus, a list of concepts neighboring to the initial term is obtained. The indexed document consists of just a series of textual strings, and these concepts must be represented as text. Here, the label property value that is represented by URI (rdfs:label) is obtained for every concept in the list of extracted adjacent concepts. The produced textual strings are recognized as the words that will be looking for in the index. The SPARQL query used to extract the result is given by:

select distinct? name where {term[]?Concept.?Concept rdfs : label?name}

In order to extract the semantic relation between term (f) and learning object (b), the set (C_f) that forms the concepts linked to (f) is determined as:

$$C_f = \{c_1, c_2, \ldots, c_n\} \quad where\ c_i\ is\ linked\ to\ 'f' \tag{4}$$

The set C_f may involve many concepts linked to a single term. Some of these concepts may be very generic and have no related semantics to the domain of the original term. It would be significant to choose more specific concepts.

The common concepts shared among several learning objects would not be semantically related to a specific term. Every concept in C_f must be significant, and thus the concepts with more specifics should be preferred over the common ones. Furthermore, adding more terms may lead the system to overfit, resulting in a low recall and overall ineffectiveness. To handle this matter, the benefit is taken from the WordNet lexicon to represent the specificity of a word. Based on the principle of specificity, the word becomes more valuable and expressive.

Thus, the specificity of a concept (c) is measured utilizing WordNet as a lexical database of semantic relations between words which adopts the principle of combining words that have similar meanings named Synsets. Term hierarchies in WordNet as a directed tree are used to specify the related concepts from Synsets. This helps out excluding the generic terms by approaching appropriate parts of the hierarchies, whereas using Synsets of WordNet obtained by selecting randomly ablating parts of the hierarchies does not meet the goal to add more specific concepts related to the original concepts.

As WordNet provides the syntactic categories; noun, verb, adjective, and adverb. It defines semantic relations between words and word senses; Synonymy, Hyponymy, Hypernymy, Antonymy, Meronymy, Troponymy, and Entailment. The two staple relations, Hypernymy (super-name) and its inverse Hyponymy (sub-name) indicate that the concept has a more general or specific semantic meaning. WordNet just respects the syntactic category "noun" with respect to the two relations (Hypernymy and Hyponymy), which the proposed approach focused on to expand the original terms, while the other aforementioned semantic relations do not serve the goal of investigation.

Therefore, the extension or generalization is a semantic shift from subordinate to superordinate levels; Hyponymy replaces Hypernymy. Since there is typically just a single hypernym, this semantic relation arranges the meanings of nouns into a hierarchical structure.

The Synsets have been organized in a hierarchical structure that forms a directed tree with a root where higher specificity is achieved by going deeper in the tree. The number of relations that connect this concept to the root is used to calculate the concept's height. Term hierarchies in WordNet directed tree make this measure is adequate to specify the related concepts.

All Synsets (SY_c) that a concept (c) is included in can be defined as follows

$$SY_c = \{s_1, s_2, \ldots, s_n) \ \ where\ c \in s_i \tag{5}$$

There may be several routes up to the root through this tree hierarchy. Hence, the maximum distance to the root is considered the height of a Synset.

Let $d(z)$ denotes to the height of s:

$$d(z) = d(hyper(z)) + 1, \quad where\ d(root) = 0 \tag{6}$$

where $hyper(c)$ is the direct Hypernymy of the concept.

The height of the concept reflects the specificity of the concept. Thus, this measure is suitable to be used to qualify the concepts that are extracted from DBpedia.

Then the following formula finds the specificity of c as given:

$$M_c = Average_{z \in SY_c}(d(z)) \tag{7}$$

Then, the term (f) probability with a material (b) is found considering the intersection of the weighted concepts in (C_f) and the vocabulary set (VS_b) in the material b, as follows:

$$S(f, b) = \sum_{c \in C_f \cap VS_b} M_c \tag{8}$$

Now, to find this probability where the sum of all probabilities should be: 1, the values of (S) are needed to normalize. So, to find the final probabilities values of $P(f|b)$, the following formula is applied:

$$P(f|b) = \frac{S(f, b)}{\sum_{z \in B} S(f, z)} * P(f) \tag{9}$$

where B denotes the entire scientific materials available, $P(f)$ refers to the term probability that could be determined through its frequency with respect to the rest terms

Thus, by using Eq. (9) in Eqs. (1) and (2) the 2-dimensional (term X material) semantic matrix values can be computed for the semantic learner profile.

3.2. CNN-based model for sentiment analysis of e-content reviews

Sentiment analysis can gauge how scientific material is relevant and can be recommended to learners using hybridization methods. This section introduces the structure of the proposed goodness construction of the CNN-based model for sentiment analysis embedded in ELHRS to predict the rating score from reviews posted by learners who express their opinions about an e-learning resource via text entries. In order to build a CNN-based model, firstly, the input and output must be decided. The input pre-computed vectors of CNN are generated by the enhanced language models using Skip-Gram (S-G) and Continuous Bag of Words (CBOW) that emerged from NLP [61,62]. The CNN-based model's output is the ratings of the e-learning resources, which are subsequently validated to serve as the basis for the recommendation phase.

3.2.1. Outline of CNN-based model for sentiment analysis

This model can be summarized according to two processes: the first one is the training process and the second one is the recommendation process. The recommendation process relies on the harmonized CNN parameters that are produced by the training process.

3.2.1.1. Training process. To train the CNN-based model, the language model based on S-G and CBOW techniques has been used to solve the input problem as addressed in Section 3.2.2, and the output problem which is regularized by L2-norm and Dropout technique [70], elaborated by the architecture of Level 2 and 3, Fig. 1. The review sentences that are vectorized by the language model and mapped to rating scores of the corresponding learning materials serve as the training data for the CNN-based model. The rating score could be either explicit or implicit wherein it relies on the rate given by the learners. The language model input is the textual reviews of the existing learning resources.

3.2.1.2. Recommendation process. The recommendation approach involves incorporating the learner's preferences and employing the CNN-based model to turn the text information into features extracted from the e-content reviews with the corresponding rating records posted by learners. This service for predicting the rating score of the e-learning resources reviews is provided by the suggested method for the recommendation. When new reviews are admitted, the algorithm can classify the unseen data connected to upcoming e-learning resources. Eventually, the suggested framework can perform well. The predicted rating score of e-learning resource from reviews can help rank it and indicates whether it achieves a good rating and the learner needs it.

To obtain the final predicted rating of the learning recourse linked to the scope of learner's interest:

a. The mean of all predicted ratings extracted from the reviews regarding this learning resource is applied.

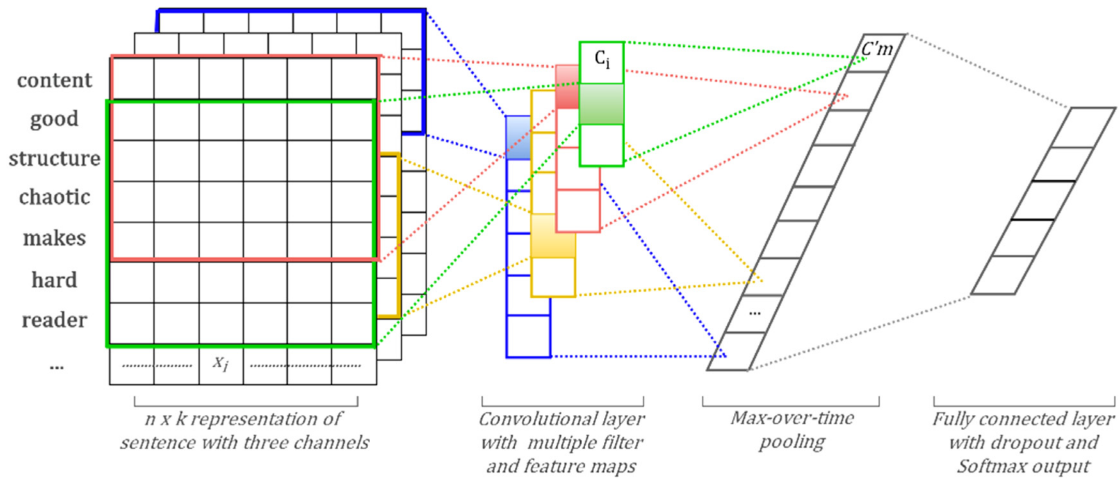

Fig. 3. CNN-based model's construction with three-channel for an instance sentence.

b. Then, a set {N} of learning resources with the top ratings computed among the learning recourses linked to the scope of the learner's interest is selected.

c. Personalized top-N recommendations are provided to the target learner. Level 4 architecture in Fig. 1 shows this process that the recommendation methodology relies on as described in Section 3.3.

3.2.2. The architecture of the CNN-based model

The CNN-based model employed by ELHRS is applied to text reviews data performing multilayer feature representations learning for the multi-classification task at the output layer for delivering meaningful suggestions, as described in Section 3.2.1, and shown in Level 3 architecture in Fig. 1. For this purpose, variant CNN-based models are used that differ with respect to deep and multichannel architecture.

Natural language can be viewed as a one-dimensional sequence of words as data of a grid-like topology to be handled by CNN, which can effectively model text problems. For the CNN construction, firstly, the convolution is used to manipulate the input word vectors sequence to create a feature map, and then max-pooling in the time dimension is applied on the feature map to get the feature (maximum value of all elements) of the entire sentence corresponding to this convolution kernel. Finally, the features extracted via the multi-hidden layers by different convolutional kernels are pooling to form fixed-length vector representations of the text. Then for text classification, it is linked to the 'Softmax' layer that completes the model. Practically, several convolutional kernels with different window sizes to process the sentences are used, and the convolutional kernel is pooling with the same window size to form an array which can complete the process more efficiently.

The CNN-based model has been structured by composing nine layers as a baseline model formed from; four convolutional layers against different feature maps; four max-over-time region pooling layers; and one fully connected layer as shown in Fig. 3.

In this model, the input data is the output pre-computed vectors of the S-G and CBOW models, which form the words' representation that is obtained from the embedding space. Every sentence in the dataset is represented, where $x_i \in \Re^k$ is the k-dimensional word vector, corresponding to the i^{th} word in a sentence of length n. Thus, the representation of the sentence is formed by the concatenation of all its x_i, as follows:

$$x_{1:n} = [x_1, x_2, \ldots, x_n], \quad \text{where } x \in \Re^{nk} \tag{10}$$

In each convolutional layer, the procedure of convolving a filter $w \in \Re^{hk}$ with a kernel size of v words is used to produce a new feature via a convolution operation. Formally, let c_i be a feature retrieved from a window of words $x_{i:i+v-1}$, as follows:

$$C_i = f\left(w.x_{i:i+v-1} + b\right) \tag{11}$$

where f indicates the non-linear activation function that 'ReLu' function is applied, and $b \in \Re$ denotes a bias expression. In this case, the filter is convolved over every possible window that contains a series of words $x_{1:v}, x_{2:v+1}, \ldots, x_{n-v+1:n}$ to produce a feature map:

$$C = [C_1, C_2, \ldots, C_{n-v+1}], \quad \text{where C} \in \Re^{\text{n}-\text{v}+1} \tag{12}$$

Then, it is followed by the pooling process where the max-pooling over time layer accepts the summit value from each feature map $C' = Max\{C\}$. The pooling operation generally deals with the variable of the sentence length that is utilized to produce one feature out of one filter.

Thus, a sequence of the features is extracted via applying several filters and kernel sizes. That is, the convolutional process generates a vector of extracted features; hence,

$$z = \left[C'_1, C'_2 \ldots, C'_m\right] \tag{13}$$

Here, m denotes the number of filters.

These features make up the penultimate layer which subsequently serves as the entrance to the fully connected "Softmax layer". This complementary layer eventually produces the probability of distributing the classes across the output. The output units compute the likelihood of each class with units from five classes being utilized, as addressed in Section 4.1.

The back-propagation algorithm is applied to perfect a characteristic tuning of the CNN parameters through the gradient of the loss function "categorical_crossentropy".

In order to eliminate or reduce the co-adaption problem (overfitting), L2-norm and Dropout [71] techniques are used on the penultimate layer, by which randomly dropping out $'p'$ a ratio of hidden units during forward-propagation in the training process. The output set in forward-propagation generally is set to be:

$$y = w * z + b \tag{14}$$

where the vector z that is resulted in the penultimate layer.

By using drop out technique, the output y turns into:

$$y = w * (z \bigoplus r) + b \tag{15}$$

where, $r \in R^m$ is a 'masking' vector of binary values that is Bernoulli random variables whose probability of p is being 1, $\bigoplus$ is

"Element-wise multiplication operator", and the applied "dropout rate" is set to 0.5.

At test time, the learned weight vectors are reformed by p, where $\hat{w} = pw$, after that ŵ is hired without dropout to appraise unseen data.

3.3. Recommendation methodology

The suggested learner profiling model is targeted to recommend the best learning resources more relevant to the learner as shown in Level 4 architecture in Fig. 1. This target is to be achieved by linking the semantic learner profile built automatically to the recommendation process. The top-k of features (terms) depending on the v_i values from the learner profile are to be picked; hence k is set to be equal to 5. These values are then normalized to set them as proportions, wherein their sum should be 1. Each value is divided by the summation of the top v_i values from the learner profile to determine the proportion $'P'$ of learner's interest in each term (f_i) out of that top-5 terms (e-learning categories) as follows:

$$P(f_i) = \frac{v_i}{\sum_{i=1}^{k} v_i}, \tag{16}$$

where k is the number of top terms

According to this proportion, some e-learning resources (books) with the highest predicted ratings are taken from each term out of the Top-five, depending on the CNN-based model and its recommendation process. These selected books with the highest predicted ratings from each term representing the learner's preferences would be recommended to the learner.

To update a value of the feature in learner profile using Eq. (2), an example is given:

If b_i is "Surfing a learning material from 'IoT' domain and f_2 is linked to 'IoT' term, as stated in Section 3.1. The value of the semantic relation $P(b|f)$ between this term and the material would be 0.671, which is to be obtained from the semantic matrix generated as mentioned in Section 3.1.1.

And $v_2 = 0.55$, by considering Max = 1, then by applying Equation (2), the result would be as follow:

$$G = 0.55x(1 - 0.55) * 0.671 = 0.166$$

The new value of v_2, using Eq. (1), is then to be:

$$v_2 = 0.55 + 0.166 = 0.716, \text{ and so on.}$$

For example, a learner profile has the following values of the features given below:

f_1	f_2	f_3	f_4	f_5	f_6	$\ldots f_n$
0.849	0.716	0.473	0.561	0.125	0.231	<0.125

Then the set F_u should be selected is $\{f_1, f_2, f_4, f_3, f_6\}$ which is the learner's interests (preferences) with the values given, it would be as the following order that is the recommendation process relies on:

f_1	f_2	f_4	f_3	f_6
0.849	0.716	0.561	0.473	0.231

Then, the proportion of interest of each term is obtained using Eq. (16), which would result as follows:

$$P(f_1) = \frac{0.849}{0.849 + 0.716 + 0.561 + 0.473 + 0.231} = 0.3$$

Suppose f_1 is "ML" and $P(f_1)$ in the example equals 0.3 so that ELHRS will recommend three books with the highest predicted rating from the same leaner's term by employing the CNN-based model for fine-grained sentiment analysis.

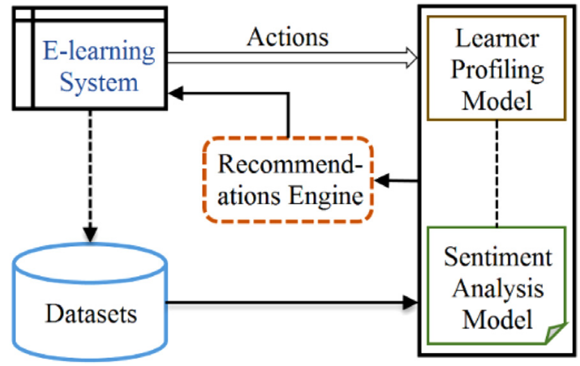

Fig. 4. Process of the proposed recommendation system.

The suggested recommendation framework can be used as a new recommender system within a personalized e-learning platform. It is a significant improvement over the existing recommendation approaches by suitably providing services tailored to the user. Fig. 4 illustrates the proposed system process. The system process can be scalable and applicable in other business domains via different definitions.

4. Experimental methods

There is a close connection between the ELHRS performance and the achieved accuracy of CNN-based models for sentiment analysis so that the focus is on the results of CNN-based models through implementing powerful models and then evaluating their performance. This section describes the several empirical studies on our customized dataset and public dataset that have been conducted to investigate system efficiency, as well as the pre-processing tasks applied on the text reviews with word embedding methods used for the enhanced language models, and how the CNN-based models have been implemented.

4.1. Data sets

Firstly, to build the semantic learner profile, it is needed to build the semantic matrix that relies on the semantic similarity between the e-learning categories (terms) and e-learning resources (scientific materials) which is generated using ontological DBpedia knowledge base and WordNet for lexical similarity as presented in Section 3.1.1. For that, forty terms were selected which models the features in the learner profile. E-content of 100-materials from these terms' domains were collected by considering the realistic requirements of actual learners based on experts' recommendations in our university.

Secondly, the effectiveness of CNN-based models has been verified through conducting several experiments for sentiment analysis on a customized dataset from online text reviews in the e-learning domain. For this purpose, work was carried out to create our dataset named ABHR (Amazon Book Hybrid Recommendation) [72]. Scraping is done on Amazon.com pages to collect reviews of various books by utilizing "Python Script". The records were collected which include user reviews of books with their corresponding ratings. This data collection method results in data sets comprising reviews for many different books from Amazon website with their ratings. The book reviews were scraped from domains that belong to different terms such as {semantic_web, machine_learning, knowledge_representation} according to the selections of the e-learning categories and matching to the proposed learner profile model.

Table 1
Analytical view of the use of S-G and CBOW.

Model	S-G	CBOW
The algorithm used to train	Neuronal Network	Neuronal Network
Prediction	Predicting the surrounding words based on a word given	Predicting the target word with respect to surrounding words
The goal	The goal of S-G is maximizing the probability of predicting context words (w_c) to be from target words' context (w_t) for the entire training pairs of the corpus; the formula of maximizing the probability is here as given below $\sum_{(w\_c \mid w\_t) \in \text{ corpus}} \log p(w\_c \mid w\_t)$	The sack of CBOW is maximizing the probability of predicting w_t to be the target of the w_c context for the entire training pairs of the corpus; the formula of maximizing the probability is here as given below: $\sum_{(w\_t \mid w\_c) \in \text{ corpus}} \log p(w\_t \mid w\_c)$
Dataset size	Can work with small amount of data	Need big size of training pairs to work properly
Representation of rare and frequent words	Well Representation for rare words and phrases but less for frequent ones.	Well Representation for frequent words but less for a rare one.
The pace of training	Slower than CBOW.	Faster hundred times than S-G.
Manipulating samples	In the case of two words beside each other which forms the input pairs to be fed to this model, one is infrequent and another is frequent. When it comes to minimizing the Loss function, both words will be treated in the same way because both words will play two roles as target and context.	While in this method, the pairs mentioned above are treated differently because the infrequent or rare words will appear only on a group of context words that form the input of the model. These context words have been utilized to predict the target word. Thus, resulted in the model will always assign a lower probability to rare words.

ABHR consists of 40,000 sentences extracted from book reviews with fine-grained sentiment labels [1-5] corresponding to specific classes [very negative, negative, natural, positive, and very positive]. For instance, a sentence[1] from a book review like ['content', 'good', 'structure', 'chaotic', 'makes', 'hard', 'reader'] with its corresponding rate '3'. The detail for this is demonstrated in Section 4.2, along with pre-processing.

In addition, the text of these reviews maps to a corpus that has been represented using S-G and CBOW to extract the word embeddings out of the vocabulary.

4.2. The pre-processing of the text reviews

Several tasks were applied to pre-process ABHR as follows:

- The different HTML tags in the corpus have been filtered, omitting the symbols except for the numbers and letters.
- Unique vocabulary (Tokens) out of the corpus have been created, removing any punctuations, stop words or special characters from the sentence.
- The filtered vocabulary have been encoded regardless of their length to keep the text information more semantic depth.
- The vocabulary that engage in encoding all sentences included in the corpus has been effectively vectorized into a form of vectors where the word embedding models can use them.

As a result, the whole corpus has been encoded to a dictionary, with each value mapping to an integer or vector representing a single word, with the key included in the dictionary being the word itself. The padding technique is then used to form fixed-length vector representations of the sentences whenever necessary. Wherein the resulted vectors are fed into the CNN-based models.

For example: the vector of the former sentence [1] as follows:

[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 21 3 836 2546 133 154 113]

Moreover, the corresponding score of the previous sentence[1] is mapped as follows:

The sentence's number in ABHR	The corresponded rating 1	2	3	4	5
349	0	0	1	0	0

After applying these processing procedures on the textual reviews, word embedding vectors are extracted from the corpus using 'S-G' and 'CBOW' language models on ABHR.

4.3. Word embedding methods

Word Embeddings is used to produce dense word vectors by which more information is collected and translated from sparse input by mapping the high dimensional data into lower-dimensional vector space. It maps the statistical structure of the language used in the corpus to infer the semantic meaning relations of the terms into a geometric space. The relationship between word vectors can be extracted in their dimension space whereas the context can be deduced through word neighbors and vice versa. Cosine or Euclidean distance can be used to cluster similar words near together in their embedding space while dissimilar words are clustered far apart. This approach is to learn the optimal vector representation of words [61,62].

Word2Vector is conducted in this context to represent the words in a massive corpus. This method represents the words as continuous vectors in N-dimension space to be used for similarity. It adopts the principle of inferring the meaning of the word based on the space in which it appears. It comprises two techniques, S-G and CBOW which employ probabilistic prediction to get the syntactic and semantic information and acquire the relationships and the similarity between words. Table 1 shows an analytical view comparing the use of S-G and CBOW techniques.

4.4. Implementation of the CNN-based models

Several ways and techniques are utilized to extract word embeddings out of the corpus to be fed to train the CNN-based models wherein the output of these NLP models forms the input of CNN. In order to extract the pre-trained vectors out of vocabulary, two methods are implemented S-G and CBOW. First, the cleaning procedures were utilized to clean ABHR. The inputs of both models are our vocabulary words while their outputs are the vectors corresponding to the vocabulary. These pre-trained vectors are then used as input to the different CNN-based models, categorized in two ways (random initialization and pre-trained vectors).

[1] Example sentence from ABHR, corresponding to text information of a review.

- The first model is the random initialization model (CNN-RIM) and is considered the baseline model, where the vocabulary words are assigned to an integer number. Each integer is pointed to a specific word and forms its index.
- The other models used the pre-computed vectors that have been trained on ABHR from scratch.

As a result, a new dataset of enhanced pre-trained vectors that were produced with each word mapping into a vector where words with adjacent meaning are grouped closely and dissimilarly far apart in their embedding space by using "Euclidean distance". These vectors were obtained by applying CBOW and S-G techniques to the corpus.

These models with the pre-computed vectors have been trained based on two manners: either keep the word vectors frozen without training during the training process presented in all CNN-Static models or update the word vectors during the training process as presented in CNN-Fine-Tune models.

In addition, other CNN architectures have been developed to get more efficiency, where two channels have been employed that formed CNN-Two-Channel models and three channels as in CNN-three-Channel models as well they were also vital to improving the performance. These four CNN-based models have been implemented on both and S-G and CBOW techniques. The dimensionality of pre-trained vectors was '300'.

More powerful models have been conducted to increase the accuracy and reduce overfitting by concatenating the vectors generated from both previous language models (CBOW and S-G). The concatenated vectors have formed the input to the CNN-based models mentioned above, where the most outstanding benefit comes from these models achieving the objective of this research.

The implementation of the different proposed CNN-based models is firstly classified into two methods (random initialization and pre-trained vectors) as mentioned above. Secondly, the method based on pre-trained vectors can be categorized into two techniques (S-G pre-trained vectors and CBOW pre-trained vectors). Furthermore, a new language model is proposed based on the concatenation method of pre-computed vectors, in which these two techniques are incorporated. Each one of the presented language models has been respectively implemented on the four proposed constructions of CNN-based models as given below:

CNN-RIM: This is the baseline model using 'random initialization' with an integer assigned to each word in the corpus.

CNN-Static-SG: In this model, the words are represented as vectors as these vectors are trained using the S-G model. During training, all the word embeddings were kept static while the remaining parameters are allowed to learn.

CNN-Fine-Tune-SG: This is similar to the previous model except for a fine-tuning that is applied to each task of the model to update the word combinations during the training process.

CNN-Two-Channel-SG: In this model, two groups of word embeddings are used with each group considered a channel. The filters are convolved over both channels. In the back-propagation process, the gradients are performed through one channel and the fine-tuning is also carried out, while the second group is left static.

CNN-Three-Channel-SG: Similar to the previous model but with improvements to the CNN structure and the introduction of a new third group. Concerning the gradients in back-propagation and fine-tuning, which is only implemented through two channels, with the third channel remaining static.

CNN-Static-CBOW: It is similar to the CNN-Static-SG model, except for the word embeddings based on pre-trained vectors that were trained with CBOW model.

CNN-Fine-Tune-CBOW: Partially similar to the previous model, but fine-tuning is applied to every task while training the model.

CNN-Two-Channel- CBOW: A cupule of channels were combined to form this model through the use of the vectors of CBOW model. Several filters with different window sizes were employed on the built-in channels. Only one channel performs gradients, while another channel remains constant.

CNN-Three-Channel- CBOW: Compared to the previous model, it joins three channels, and filters are applied to each channel. Moreover, gradients are implemented in back-propagation only through two channels, while the last channel is kept without updating.

CNN-Static-Concatenation: This model employs a new method for creating the word pre-computed vectors. The concatenation of the vectors produced by both S-G and CBOW models has been used to create the word embeddings of this model for each word. The gradients are kept static, while the rest parameters are allowed to learn.

CNN-Fine-Tune-Concatenation: The construction of this model is identical to the CNN-Static-Concatenation model, while gradients are allowed to update.

CNN-Two-Channel-Concatenation: The model was implemented by a two-channel procedure. Filters and kernels are applied to both. The back-propagation is only through one channel and the second is kept static.

CNN-Three-Channel-Concatenation: Much like the previous model that is evolved by adding three groups to the CNN architecture, with each group forming a channel. One channel is kept constant, while the rest performers update the gradients in back-propagation.

The proposed CNN-based models described above have been validated by training with the public dataset "Stanford Sentiment Treebank" (SST-1); produced with the following splits; train/dev/test, which is labeled (very negative, negative, neutral, positive, very positive), relabeled in [73]. The same has been trained on ABHR.

As per experiments, the self-designed CNN-based models were trained and tested on various parameters, while their determinants are reported based on numerous validations utilizing the language models based on SG and CBOW, as stated in Table 2.

These CNN-based models were implemented by Python Programming using the "Keras Package" whereas the "Genism Package" was utilized for the word embedding methods with the parameters shown in Table 3. Our experiments revealed the findings of the obtained word representations with the best vector size of 300.

5. Results and discussion

The proposed system ELHRS uses knowledge-based techniques to capture learners' preferences by deducing the semantic learner profile automatically based on the learner's behaviors. Semantic similarity with WordNet depending on the ontological DBpedia knowledge base has been utilized for expanding the initial terms to enrich the semantic linking between e-learning categories and

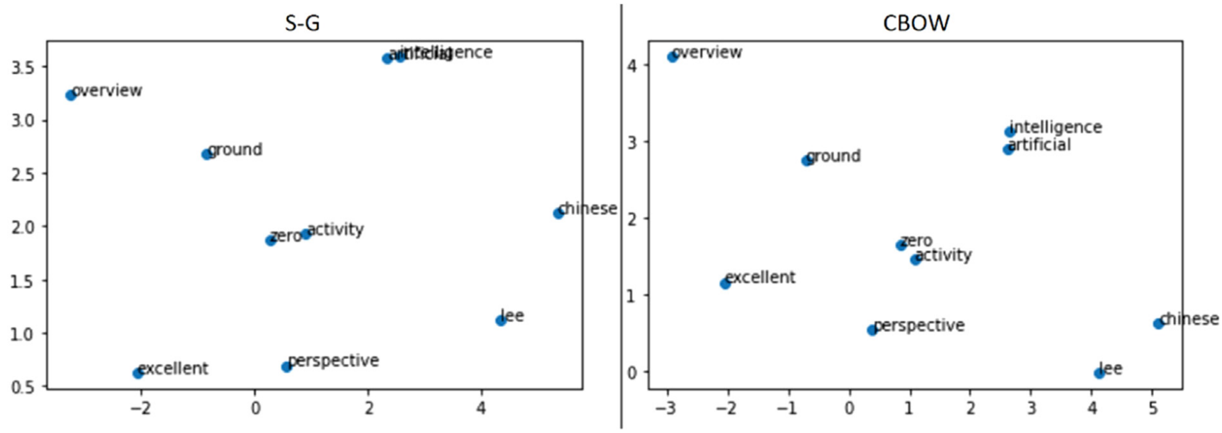

Fig. 5. Word embeddings of S-G and CBOW representation in low-dimension space.

Table 2
Training parameters of the CNN-based models.

Number of convolutional layers	4
Filter sizes	3,4,5,6
Feature maps	300
L2-norm constraint	2
Dropout ratio	0.5
Batch size	55
“dev set” size	10% out of the training set is randomly selected.
Training the model	Stochastic gradient descent was used over shuffled mini-batches.
Optimizer	“Adam Optimization” [74]

groups of materials. The learner profile is updated according to continuous observation and navigation logs linked to different browsing actions. That is efficiently reflected on meaningful building the dynamic learner profile to represent the learner interests more expressively. More semantic and lexical information was incorporated in the proposed approach for term expansion eliminating sparse data in the semantic matrix.

Using a lexical database alone cannot find semantically related words. It may find lexically similar words that share some connotations with the original terms but will not get a competent list of related concepts. The findings have shown that semantics-aware term expansion using WordNet to add weighted related concepts from a global knowledge base containing data from almost all fields provides more accurate results than the statistical manner.

It is more effective to enrich the model by expanding the terms from the field (domain) of the original term based on the semantic similarity, especially in recommending e-learning content. So semantic representations based on the ontology would help better identify terms relevant to the learning domain. Moreover, adding more specific concepts evolves information retrieval exceedingly that assists in understanding the learner's interests well, thus reducing overfitting and improving the recall and precision of the system.

As the dynamic learner data changes over time, automatic updating of the learner profile is obtained using retrieved information and semantic representation using WordNet and DBpedia ontologies. Thus, the captured learner's interests are suitability used as an input to the hybrid recommendation engine resulting in more accurate predictions of e-learning resources to the learner. Moreover, using the semantic-based methods incorporated with DL-based models can tackle the ‘Sparsity’ and ‘Cold-start’ problems from which the traditional recommender systems suffer.

Furthermore, the possibility of adjustment to generalize the learner profile model can be produced as a web service for automatic user profiling service to the websites and would suit many cases on the web.

Besides, the significance of the proposed recommendation methodology is to achieve improved results from constructing superior CNN-based models for affective sentence classification to determine the sentiment from the reviews. These variant models were used to experiment with alternate methods (as described in Table 4) by using ABHR and measuring accuracy to estimate the outcomes of the fine-grained sentiment analysis models. The comparison is made between the variant CNN-based models demonstrated in Section 4.4. The baseline model is CNN-RIM model where all words are initialized randomly which performed an accuracy of 69.5% (Table 4).

The remaining experiments were conducted to compare CBOW and S-G in terms of word initialization where the text of the reviews from ABHR formed the input to CBOW and S-G language models. These two methods have been trained on ABHR very well, as shown in Fig. 5 which depicts the findings of applying CBOW and S-G methods for extracting word embeddings. It shows the representation of the first 10 words in ABHR where words are mapped into low-dimension space. The small distance between the two vectors of the two words (‘artificial’ and ‘intelligence’) is noted. That means these two words are very close to each other.

Performance evaluation of the CNN-based models in terms of Accuracy, Precision, Recall, and F1 Measure are listed in Table 4. The table also shows a performance comparison of our models applied to ABHR against other models conducted in [33] and the result of training ‘SST-1’ on our CNN-based models. The highest performance based on these experimental results is presented in bold.

The experimental results clearly show that the static models for the three methods (CBOW, S-G, Concatenation) gave low accuracy while keeping the overfitting in low proportion, whereas the Fine-Tune models perform with better accuracy while bringing more overfitting. Thus, to reduce the overfitting and acquire higher accuracy, models with two and three channels are implemented that showed higher performance measures over the previous models (Fig. 6).

In order to get a better representation of word embeddings, the concatenation method with the Three-channel model was conducted where it obtained the lowest overfitting with the most remarkable accuracy by comparing with Static, Fine-Tune, and Two-channel models. In light of brilliant findings, it is observed that the concatenation method outweighs the S-G and CBOW methods when each of them is applied separately on CNN-based models and presents satisfactory results regarding reducing the overfitting and producing higher accuracy. This method

Table 3
Parameters used to train word embeddings.

Model	Dimensionality	Window	Sampling	Negative	Min_Count	Alpha	Min_Alpha	Iterations
S-G	300	20	6e−5	15	5	0.03	0.0007	50
CBOW	300	20	6e−5	15	5	0.03	0.0007	50

Table 4
Performance comparison of our CNN-based models implemented on ABHR with different initializations of word embeddings against other methods.

Datasets	Word Embeddings	Model	Accuracy	Precision	Recall	F1
ABHR	Random initialization	CNN-RIM	69.5	68.3	70.1	69.1
	S-G	CNN-Static-SG	33.5	33.1	33.4	32.4
		CNN-Fine-Tune-SG	72.6	71.4	73.1	72.2
		CNN-Two-Channel-SG	74.0	72.9	74.5	73.5
		CNN-Three-Channel-SG	76.2	75.2	76.1	76.0
	CBOW	CNN-Static-CBOW	34.5	37.1	34.9	35.9
		CNN-Fine-Tune-CBOW	75.7	75.5	75.4	74.9
		CNN-Two-CBOW	77.5	77.5	77.3	76.6
		CNN-Three-Channel-CBOW	84.9	86.3	85.7	85.6
	Concatenation between the CBOW and S-G vectors	CNN-Static-Concatenation	49.8	50.0	49.8	49.5
		CNN-Fine-Tune-Concatenation	86.0	86.3	86.0	86.0
		CNN-Two-Concatenation	87.1	87.2	87.0	87.0
		CNN-Three-Channel-Concatenation	**89.1**	**88.1**	**87.9**	**87.9**
SST-1 [73]	Concatenation between the CBOW and S-G vectors	CNN-Static-Concatenation	34.7	30.9	32.8	30.3
		CNN-Fine-Tune-Concatenation	55.6	55.4	54.4	54.5
		CNN-Two-Concatenation	54.3	53.6	54.8	53.8
		CNN-Three-Channel-Concatenation	57.3	57.8	58.9	57.3
	'Mikolov' [62]	CNN-non-static [33]	48.0	–	–	–
		CNN-multichannel [33]	47.4	–	–	–

resulted in the robust performance of the CNN-Three-Channel-Concatenation model with an accuracy of 89.1%. Moreover, the process of CNN has not been penalized, and complexity has not taken place during the training process. Conversely, the performance of state-of-the-art models based on CNN has been exceeded by the CNN-Three-Channel-Concatenation model when addressing best pre-computed vector representations for fine-grained sentiment analysis. Fig. 6 shows that better experimental results of conducting the CNN-Concatenation-based models where the best is the CNN-Three-Channel-Concatenation model in terms of Accuracy and Loss.

Also, all the models based on the concatenation method have been trained on ABHR and SST-1. From the experimental results shown in Table 4, training 'SST-1' on the CNN-Three-Channel-Concatenation model gave a better accuracy of 57.3%, in comparison with training it on the rest of the CNN-Concatenation-based models. The performance is also higher to training it on other models conducted in [33] that acquire 48.0%. Lower results have been found in general by applying these CNN-Concatenation-based models to 'SST-1' compared to those applied to ABHR respectively. This justifies word embeddings relying on the dataset created for a specific domain rather than a public domain and gives proof epistemic power of ABHR. Also, the result proves the validity and robustness of our CNN-Concatenation-based models against other models implemented in [33] when applied to the same dataset 'SST-1'.

We compared the experimental results of our various CNN-based models with some sentiment analysis models that have been implemented for sentiment multi-classification relying on five classes in the literature. The public dataset 'Mikolove' [62] for word vector initialization has been utilized to extract pre-trained word vectors in [33]. The proposed language models based on S-G and CBOW using ABHR resulted in the best vocabulary representation to create the pre-computed word vectors dataset leading to greater efficiency by the concatenation method. That is possible due to the word embeddings created from data for a specific domain to evolve the performance of our models which is reflected in the results making the difference obvious. Moreover, the training parameters of our CNN-based models have been effectively learned which are different from what they used, and make our models reports so better results.

In [33], CNN-non-static model has given a better accuracy of 48.0% in multi-classification of five classes rather than the CNN-multichannel model that resulted in less overfitting and an accuracy of 47.4 %. Our experiments have shown an improvement in our models, especially the results of CNN-Three-Channel-Concatenation in terms of accuracy that significantly exceeds the results obtained in [33] as shown in Fig. 7 with lesser overfitting.

Despite applying sentimental classification on text reviews labeled with five classes, the empirical findings indicate increased performance with promising outcomes. This research presents an improved and more forward-looking scope of fine-grained sentiment analysis, whereas most existing research has been carried out to merely determine the positive and negative polarity of sentiment.

To summarize, the CNN-based models built rely on different ways: by keeping the weights frozen to compose the static models or leaving the gradients updated during the training process as in Fine-Tune models. Furthermore, multi adjustments are made on CNN architecture by constructing models with two and three channels. The results of the various applied CNN-based models were compared with each other, and with other CNN models targeting only 5 classes performed in [33] as well. The comparison between the models focused on increasing the accuracy and solving the issue of overfitting for a better operation. These models have implemented word embeddings based on S-G, CBOW, and Concatenation methods respectively, where each model uniquely qualifies the entire recommender system.

6. Conclusion

Enhancing the e-learning system requires hybridization methods utilizing emerging techniques such as sentiment analysis and deep machine learning that are reflected in the current work making the system an intelligent portal. The literature study presented an analytical review on these state-of-the-art techniques

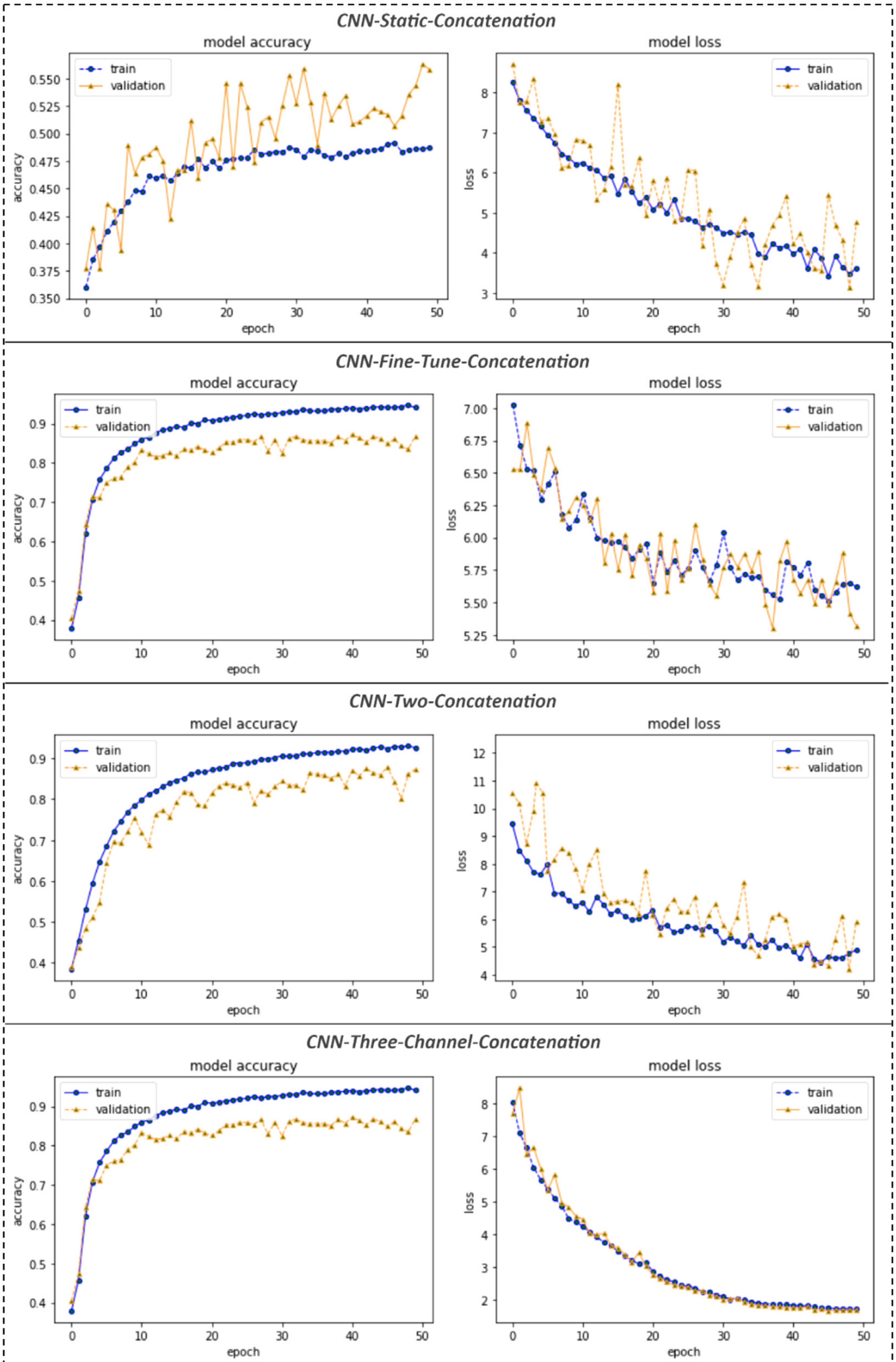

Fig. 6. Accuracy and Loss of CNN-Concatenation-based models.

and how the researchers treat the recommendation as a concept integrated with the adaptive e-learning process. Many studies addressed this concept from a narrower scope disregarding several influential factors and other necessary components.

The proposed framework for designing ELHRS is based on integrating the automatic learner profiling model and various fine-grained sentiment analysis models. It attempts to predict the relevant e-learning materials with the highest ratings linked to

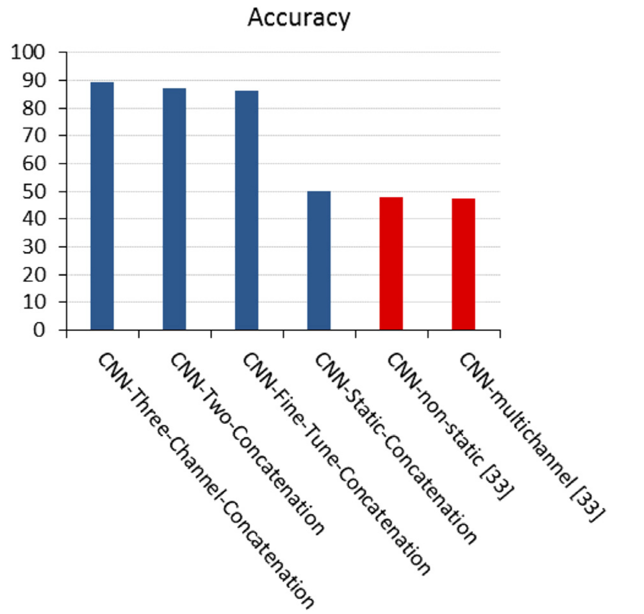

Fig. 7. Accuracy comparison of our CNN-Concatenation-based models against other models.

the learner preferences leveraging CNN-based models combined with NLP techniques for sentiment analysis. The semantic learner profile has been built automatically depending on the learner's behavior (actions). The semantic relations matrix linked between the e-learning categories and the materials group was generated using DBpedia and WordNet ontologies which are utilized with the suggested association rules. It tries to overcome the lack of building an adaptive learner profile model and bringing individualization by providing tailor-made services during the learning process. This displayed significantly improved performance and resulted in meaningful outcomes compared with other literature models.

As part of ELHRS, proposed thirteen different CNN-based models for sentiment analysis from book reviews towards the learning materials were evaluated on ABHR and SST-1 data sets and used in the hybrid recommendation process. Improved results were obtained by performing fine-grained sentiment classification on sentence-level as compared with other methods. Practically, better findings were achieved by the four CNN-based models using the concatenation of the two methods (S-G and CBOW) where CNN-Three-Channel-Concatenation model performed the best accuracy 89.1%. In contrast, most cases of published research focus on the two-polarity classification, and for good reason, fine-grained sentiment analysis is a significantly more challenging task.

The learner profiling model is scalable even in evolving the association relations among learner profile attributes themselves. Further, Fuzzy logic can be utilized in the current ontology-based probabilistic model to reflect the real world more precisely. Research can also be conducted on multichannel with "POS tag" along with hybrid deep network models for fine-grained sentiment classification. Furthermore, the datasets could be planned to be built by automotive methods at the time of insertion using cloud storage. Augmentation techniques may also be tried to generate some training data. For doing so, Generative Adversary Network (GAN) could be explored.

CRediT authorship contribution statement

Hadi Ezaldeen: Conceptualization, Design of study, Methodology, Acquisition and curation of data, Analysis of data, Software, Resources, Validation, Formal analysis, Investigation, Drafting the manuscript, Visualization, Revising the technical part, Revising the manuscript critically for important intellectual content, Review editing, Supervision. **Rachita Misra:** Conceptualization, Design of study, Methodology, Revising the manuscript critically for important intellectual content, Review editing, Supervision. **Sukant Kishoro Bisoy:** Conceptualization, Design of study, Methodology, Revising the manuscript critically for important intellectual content, Review editing, Supervision. **Rawaa Alatrash:** Acquisition and curation of data, Software, Resources, Revising the technical part. **Rojalina Priyadarshini:** Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

Approval of the version of the manuscript to be published: H. Ezaldeen, R. Misra, S. K. Bisoy, R. Alatrash, R. Priyadarshini.

References

[1] I. Uddin, A.S. Imran, K. Muhammad, N. Fayyaz, M. Sajjad, A systematic mapping review on MOOC recommender systems, IEEE Access 9 (2021) 118379–118405, http://dx.doi.org/10.1109/ACCESS.2021.3101039.
[2] S. Knight, A. Gibson, A. Shibani, Implementing learning analytics for learning impact: Taking tools to task, Internet High. Educ. 45 (2020) http://dx.doi.org/10.1016/j.iheduc.2020.100729.
[3] R. Burke, Hybrid web recommender systems, in: The Adaptive Web, Springer, Berlin, Heidelberg, 2007, pp. 377–408, http://dx.doi.org/10.1007/978-3-540-72079-9_12.
[4] J. Xiao, M. Wang, B. Jiang, J. Li, A personalized recommendation system with combinational algorithm for online learning, J. Ambient Intell. Humaniz. Comput. 9 (3) (2018) 667–677, http://dx.doi.org/10.1007/s12652-017-0466-8.
[5] J.K. Tarus, Z. Niu, G. Mustafa, Knowledge-based recommendation: a review of ontology-based recommender systems for e-learning, Artif. Intell. Rev. 50 (1) (2018) 21–48, http://dx.doi.org/10.1007/s10462-017-9539-5.
[6] Z. Han, J. Wu, C. Huang, Q. Huang, M. Zhao, A review on sentiment discovery and analysis of educational big-data, Wiley Interdiscip. Rev.: Data Min. Knowl. Discov. 10 (1) (2020) http://dx.doi.org/10.1002/widm.1328.
[7] L. Zhang, S. Wang, B. Liu, Deep learning for sentiment analysis: A survey, Wiley Interdiscip. Rev.: Data Min. Knowl. Discov. 8 (4) (2018) e1253.
[8] E. Cambria, Affective computing and sentiment analysis, IEEE Intell. Syst. 31 (2) (2016) 102–107, http://dx.doi.org/10.1109/MIS.2016.31.
[9] S. Poria, E. Cambria, R. Bajpai, A. Hussain, A review of affective computing: From unimodal analysis to multimodal fusion, Inf. Fusion 37 (2017) 98–125, http://dx.doi.org/10.1016/j.inffus.2017.02.003.
[10] W. Peng, X. Hong, G. Zhao, Adaptive modality distillation for separable multimodal sentiment analysis, IEEE Intell. Syst. 36 (3) (2021) 82–89, http://dx.doi.org/10.1109/MIS.2021.3057757.
[11] L. Stappen, A. Baird, L. Christ, L. Schumann, B. Sertolli, E.M. Messner, E. Cambria, G. Zhao, B.W. Schuller, The MuSe 2021 multimodal sentiment analysis challenge: Sentiment, emotion, physiological-emotion, and stress, 2021, arXiv preprint arXiv:2104.07123.
[12] F. Alattar, K. Shaalan, A survey on opinion reason mining and interpreting sentiment variations, IEEE Access 9 (2021) 39636–39655, http://dx.doi.org/10.1109/ACCESS.2021.3063921.
[13] H. Tang, Y. Mi, F. Xue, Y. Cao, Graph domain adversarial transfer network for cross-domain sentiment classification, IEEE Access 9 (2021) 33051–33060, http://dx.doi.org/10.1109/ACCESS.2021.3061139.
[14] D. Wang, et al., Coarse alignment of topic and sentiment: A unified model for cross-lingual sentiment classification, IEEE Trans. Neural Netw. Learn. Syst. 32 (2) (2021) 736–747, http://dx.doi.org/10.1109/TNNLS.2020.2979225.
[15] F. Leão, K. Revoredo, F. Baião, Extending WordNet with UFO foundational ontology, J. Web Semant. 57 (2019) 100499, http://dx.doi.org/10.1016/j.websem.2019.02.002.

[16] K. Singh, I. Lytra, A.S. Radhakrishna, S. Shekarpour, M.E. Vidal, J. Lehmann, No one is perfect: Analysing the performance of question answering components over the dbpedia knowledge graph, J. Web Semant. 65 (2020) 100594, http://dx.doi.org/10.1016/j.websem.2020.100594.

[17] Y. Ma, X. Geng, J. Wang, A deep neural network with multiplex interactions for cold-start service recommendation, IEEE Trans. Eng. Manage. 68 (1) (2021) 105–119, http://dx.doi.org/10.1109/TEM.2019.2961376.

[18] F. Martin, D. Markant, Adaptive learning modules, in: M. David, M. Amey (Eds.), The SAGE Encyclopedia of Higher Education, vol. 1, SAGE Publications, Inc., Thousand Oaks, CA, 2020, pp. 79–80, http://dx.doi.org/10.4135/9781529714395.n31.

[19] F. Martin, Y. Chen, R.L. Moore, C.D. Westine, Systematic review of adaptive learning research designs, context, strategies, and technologies from 2009 to 2018, Educ. Technol. Res. Dev. 68 (4) (2020) 1903–1929, http://dx.doi.org/10.1007/s11423-020-09793-2.

[20] S.M. Aslam, A.K. Jilani, J. Sultana, L. Almutairi, Feature evaluation of emerging E-learning systems using machine learning: An extensive survey, IEEE Access 9 (2021) 69573–69587, http://dx.doi.org/10.1109/ACCESS.2021.3077663.

[21] N.B.A. Normadhi, L. Shuib, H.N.M. Nasir, A. Bimba, N. Idris, V. Balakrishnan, Identification of personal traits in adaptive learning environment: Systematic literature review, Comput. Educ. 130 (2019) 168–190, http://dx.doi.org/10.1016/j.compedu.2018.11.005.

[22] Y. Zhou, C. Huang, Q. Hu, J. Zhu, Y. Tang, Personalized learning full-path recommendation model based on LSTM neural networks, Inform. Sci. 444 (2018) 135–152, http://dx.doi.org/10.1016/j.ins.2018.0002.053.

[23] K.R. Premlatha, B. Dharani, T.V. Geetha, Dynamic learner profiling and automatic learner classification for adaptive e-learning environment, Interact. Learn. Environ. 24 (6) (2016) 1054–1075, http://dx.doi.org/10.1080/10494820.2014.948459.

[24] I. Azzi, A. Jeghal, A. Radouane, A. Yahyaouy, H. Tairi, A robust classification to predict learning styles in adaptive e-learning systems, Educ. Inf. Technol. 25 (1) (2020) 437–448, http://dx.doi.org/10.1007/s10639-019-09956-6.

[25] S. Graf, S.R. Viola, T. Leo, Kinshuk, In-depth analysis of the Felder-Silverman learning style dimensions, J. Res. Technol. Educ. 40 (1) (2007) 79–93, http://dx.doi.org/10.1080/15391523.2007.10782498.

[26] S.M. Nafea, F. Siewe, Y. He, A novel algorithm for dynamic student profile adaptation based on learning styles, in: Y. Bi, R. Bhatia, S. Kapoor (Eds.), Intelligent Systems and Applications. IntelliSys 2019. Advances in Intelligent Systems and Computing, Vol. 1037, Springer, Cham, 2020, pp. 29–51, http://dx.doi.org/10.1007/978-3-030-29516-5_4.

[27] S. Jagadeesan, J. Subbiah, Real-time personalization and recommendation in adaptive learning management system, J. Ambient Intell. Humaniz. Comput. (2020) 1–11, http://dx.doi.org/10.1007/s12652-020-01729-1.

[28] M. Rani, R. Nayak, O.P. Vyas, An ontology-based adaptive personalized e-learning system, assisted by software agents on cloud storage, Knowl.-Based Syst. 90 (2015) 33–48, http://dx.doi.org/10.1016/j.knosys.2015.10.002.

[29] S. Ouf, M. Abd Ellatif, S.E. Salama, Y. Helmy, A proposed paradigm for smart learning environment based on semantic web, Comput. Hum. Behav. 72 (2017) 796–818, http://dx.doi.org/10.1016/j.chb.2016.08.030.

[30] T. Sheeba, R. Krishnan, Semantic predictive model of student dynamic profile using fuzzy concept, Procedia Comput. Sci. 132 (2018) 1592–1601, http://dx.doi.org/10.1016/j.procs.2018.05.124.

[31] T. Hamim, F. Benabbou, N. Sael, Toward a generic student profile model, in: M. Ben Ahmed, A. Boudhir, D. Santos, M. El Aroussi, İ. Karas (Eds.), Innovations in Smart Cities Applications Edition 3. SCA 2019. Lecture Notes in Intelligent Transportation and Infrastructure, Springer, Cham, 2020, pp. 200–214, http://dx.doi.org/10.1007/978-3-030-37629-1_16.

[32] M. Birjali, M. Kasri, A. Beni-Hssane, A comprehensive survey on sentiment analysis: Approaches, Chall. Trends Knowl.-Based Syst. (2021) 107134, http://dx.doi.org/10.1016/j.knosys.2021.107134.

[33] Y. Kim, Convolutional neural networks for sentence classification, 2014, arXiv:1408.5882.

[34] L. Kong, C. Li, J. Ge, F. Zhang, Y. Feng, Z. Li, B. Luo, Leveraging multiple features for document sentiment classification, Inform. Sci. 518 (2020) 39–55, http://dx.doi.org/10.1016/j.ins.2020.01.012.

[35] K. Wang, Y. Zhang, Topic sentiment analysis in online learning community from college students, J. Data Inf. Sci. 5 (2) (2020) 33–61, http://dx.doi.org/10.2478/jdis-2020-0009.

[36] Y. Zhang, Z. Zhang, D. Miao, J. Wang, Three-way enhanced convolutional neural networks for sentence-level sentiment classification, Inform. Sci. 477 (2019) 55–64, http://dx.doi.org/10.1016/j.ins.2018.10.030.

[37] Q. Jiang, L. Chen, W. Zhao, M. Yang, Toward aspect-level sentiment modification without parallel data, IEEE Intell. Syst. 36 (1) (2021) 75–81, http://dx.doi.org/10.1109/MIS.2021.3052617.

[38] E. Cambria, D. Das, S. Bandyopadhyay, A. Feraco, Affective computing and sentiment analysis, in: E. Cambria, D. Das, S. Bandyopadhyay, A. Feraco (Eds.), A Practical Guide To Sentiment Analysis. Socio-Affective Computing, vol 5, Springer, Cham, 2017, http://dx.doi.org/10.1007/978-3-319-55394-8_1.

[39] B.M. Hsu, Comparison of supervised classification models on textual data, Mathematics 8 (5) (2020) 851, http://dx.doi.org/10.3390/math8050851.

[40] G.A. Ruz, P.A. Henríquez, A. Mascareño, Sentiment analysis of Twitter data during critical events through Bayesian networks classifiers, Future Gener. Comput. Syst. 106 (2020) 92–104, http://dx.doi.org/10.1016/j.future.2020.01.005.

[41] M. Dragoni, M. Federici, A. Rexha, An unsupervised aspect extraction strategy for monitoring real-time reviews stream, Inf. Process. Manage. 56 (3) (2019) 1103–1118, http://dx.doi.org/10.1016/j.ipm.2018.04.010.

[42] Z. Kastrati, A.S. Imran, A. Kurti, Weakly supervised framework for aspect-based sentiment analysis on students' reviews of MOOCs, IEEE Access 8 (2020) 106799–106810, http://dx.doi.org/10.1109/ACCESS.2020.3000739.

[43] N. Majumder, S. Poria, H. Peng, N. Chhaya, E. Cambria, A. Gelbukh, Sentiment and sarcasm classification with multitask learning, IEEE Intell. Syst. 34 (3) (2019) 38–43, http://dx.doi.org/10.1109/MIS.2019.2904691.

[44] X. Wang, M. Tang, T. Yang, Z. Wang, A novel network with multiple attention mechanisms for aspect-level sentiment analysis, Knowl.-Based Syst. 227 (2021) 107196, http://dx.doi.org/10.1016/j.knosys.2021.107196.

[45] J. Su, J. Tang, H. Jiang, Z. Lu, Y. Ge, L. Song, D. Xiong, L. Sun, J. Luo, Enhanced aspect-based sentiment analysis models with progressive self-supervised attention learning, Artificial Intelligence 296 (2021) 103477, http://dx.doi.org/10.1016/j.artint.2021.103477.

[46] A. Onan, Sentiment analysis on massive open online course evaluations: a text mining and deep learning approach, Comput. Appl. Eng. Educ. 29 (3) (2021) 572–589, http://dx.doi.org/10.1002/cae.22253.

[47] Y. Susanto, E. Cambria, B.C. Ng, A. Hussain, Ten years of sentic computing, Cogn. Comput. (2021) 1–19, http://dx.doi.org/10.1007/s12559-021-09824-x.

[48] D. Mauro, I. Donadello, E. Cambria, OntoSenticNet 2: Enhancing reasoning within sentiment analysis, IEEE Intell. Syst. 36 (5) (2021).

[49] L. Zhuang, K. Schouten, F. Frasincar, SOBA: SEmi-automated ontology builder for aspect-based sentiment analysis, J. Web Semant. 60 (2020) 100544, http://dx.doi.org/10.1016/j.websem.2019.100544.

[50] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, J. Gao, Deep learning–based text classification: A comprehensive review, ACM Comput. Surv. 54 (3) (2021) 1–40, http://dx.doi.org/10.1145/3439726.

[51] R. Alatrash, H. Ezaldeen, R. Misra, R. Priyadarshini, Sentiment analysis using deep learning for recommendation in E-learning domain, in: C.R. Panigrahi, B. Pati, B.K. Pattanayak, S. Amic, K.C. Li (Eds.), Progress in Advanced Computing and Intelligent Engineering. Advances in Intelligent Systems and Computing, vol. 1299, Springer, Singapore, 2021, http://dx.doi.org/10.1007/978-981-33-4299-6_10.

[52] M.E. Basiri, S. Nemati, M. Abdar, E. Cambria, U.R. Acharya, ABCDM: AN attention-based bidirectional CNN-RNN deep model for sentiment analysis, Future Gener. Comput. Syst. 115 (2021) 279–294, http://dx.doi.org/10.1016/j.future.2020.08.005.

[53] D. Meškelė, F. Frasincar, ALDONAR: A hybrid solution for sentence-level aspect-based sentiment analysis using a lexicalized domain ontology and a regularized neural attention model, Inf. Process. Manage. 57 (3) (2020) 102211, http://dx.doi.org/10.1016/j.ipm.2020.102211.

[54] G. Alexandridis, J. Aliprantis, K. Michalakis, K. Korovesis, P. Tsantilas, G. Caridakis, A knowledge-based deep learning architecture for aspect-based sentiment analysis, Int. J. Neural Syst. 2021 (2021) 2150046, http://dx.doi.org/10.1142/S0129065721500465.

[55] Y. Fu, J. Liao, Y. Li, S. Wang, D. Li, X. Li, Multiple perspective attention based on double BiLSTM for aspect and sentiment pair extract, Neurocomputing 438 (2021) 302–311, http://dx.doi.org/10.1016/j.neucom.2021.01.079.

[56] Y. Lin, Y. Fu, Y. Li, G. Cai, A. Zhou, Aspect-based sentiment analysis for online reviews with hybrid attention networks, World Wide Web 24 (2021) 1215–1233, http://dx.doi.org/10.1007/s11280-021-00898-z.

[57] E. Cambria, Y. Li, F.Z. Xing, S. Poria, K. Kwok, SenticNet 6: Ensemble application of symbolic and subsymbolic AI for sentiment analysis, in: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, 2020, pp. 105–114, http://dx.doi.org/10.1145/3340531.3412003.

[58] L. Mou, Z. Lu, H. Li, Z. Jin, Coupling distributed and symbolic execution for natural language queries, Int. Conf. Mach. Learn. 70 (2017) 2518–2526.

[59] Y. Zhang, Z. Teng, Natural Language Processing: A Machine Learning Perspective, Cambridge University Press, Cambridge, 2021, http://dx.doi.org/10.1017/9781108332873.

[60] K. Panesar, Conversational artificial intelligence-demystifying statistical vs linguistic NLP solutions, J. Computer-Assisted Linguist. Res. 4 (1) (2020) 47–79, http://dx.doi.org/10.4995/jclr.2020.12932.

[61] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, 2013, arXiv:1301.3781.

[62] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Advances in neural information processing systems, 2013, pp. 3111–3119.

[63] A.R. Pathak, B. Agarwal, M. Pandey, S. Rautaray, Application of deep learning approaches for sentiment analysis, in: B. Agarwal, R. Nayak, N. Mittal, S. Patnaik (Eds.), Deep Learning-Based Approaches for Sentiment Analysis. Algorithms for Intelligent Systems, Springer, Singapore, 2020, pp. 1–31, http://dx.doi.org/10.1007/978-981-15-1216-2_1.

[64] H. Xia, C. Ding, Y. Liu, Sentiment analysis model based on self-attention and character-level embedding, IEEE Access 8 (2020) 184614–184620, http://dx.doi.org/10.1109/ACCESS.2020.3029694.

[65] M. Gimenez, J. Palanca, V. Botti, Semantic-based padding in convolutional neural networks for improving the performance in natural language processing. A case of study in sentiment analysis, Neurocomputing 378 (2020) 315–323, http://dx.doi.org/10.1016/j.neucom.2019.08.096.

[66] D. Dessí, M. Dragoni, G. Fenu, M. Marras, D.R. Recupero, Deep learning adaptation with word embeddings for sentiment analysis on online course reviews, in: B. Agarwal, R. Nayak, N. Mittal, S. Patnaik (Eds.), Deep Learning-Based Approaches for Sentiment Analysis. Algorithms for Intelligent Systems, Springer, Singapore, 2020, pp. 57–83, http://dx.doi.org/10.1007/978-981-15-1216-2_3.

[67] H. Ezaldeen, R. Misra, R. Alatrash, R. Priyadarshini, Semantically enhanced machine learning approach to recommend e-learning content, Int. J. Electron. Bus. 15 (4) (2020) 389–413, http://dx.doi.org/10.1504/IJEB.2020.111095.

[68] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, Z. Ives, Dbpedia: A nucleus for a web of open data, in: K. Aberer, et al. (Eds.), The Semantic Web. ISWC 2007, ASWC 2007. Lecture Notes in Computer Science, Vol. 4825, 2007, Springer, Berlin, Heidelberg, 2007, pp. 722–735, http://dx.doi.org/10.1007/978-3-540-76298-0_52.

[69] G.A. Miller, WordNet: A lexical database for English, Commun. ACM 38 (11) (1995) 39–41, http://dx.doi.org/10.1145/219717.219748.

[70] G.E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, R.R. Salakhutdinov, Improving neural networks by preventing co-adaptation of feature detectors, 2012, arXiv:1207.0580.

[71] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (1) (2014) 1929–1958.

[72] H. Ezaldeen, R. Alatrash, ABHR, GitHub, 2021, Retrieved from: https://github.com/HEzaldeen/ABHR.

[73] R. Socher, A. Perelygin, J. Wu, J. Chuang, C.D. Manning, A.Y. Ng, C. Potts, Recursive deep models for semantic compositionality over a sentiment treebank, in: Proceedings of the 2013 conference on empirical methods in natural language processing, 2013, pp. 1631–1642.

[74] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, arXiv:1412.6980.