Deep Attention-based Model for Helpfulness Prediction of Healthcare Online Reviews

Sergio Consoli[1⋆][0000−0001−7357−5858], Danilo Dessì[2][0000−0003−3843−3285], Gianni Fenu[2][0000−0003−4668−2476], and Mirko Marras[2][0000−0003−1989−6057]

[1] Joint Research Centre European Commission, Ispra, Italy
sergio.consoli@ec.europa.eu
[2] University of Cagliari, Cagliari, Italy
{danilo_dessi, fenu, mirko.marras}@unica.it

Abstract. With tons of healthcare reviews being collected online, finding helpful opinions among this collective intelligence is becoming harder. Existing literature in this domain usually tackled helpfulness prediction with machine-learning models optimized for binary classification. While they can filter out a subset of reviews, users might be still overwhelmed if the number of reviews marked as helpful is high. In this paper, we design a new neural model optimized for predicting a continuous score that can be used to rank reviews based on their helpfulness. Given embedding representations of words in a review, the proposed model processes them through recurrent and attention-based layers to solve a helpfulness prediction task, modeled as a regression. Experiments on a real-world healthcare dataset show that the proposed model optimized for regression leads to accurate helpfulness prediction and better helpfulness-based rankings than models optimized for binary classification.

Keywords: Machine Learning · Deep Learning · Ranking · Recommendation · Healthcare · Helpfulness Prediction · Review Usefulness.

1 Introduction

Online platforms are fostering more and more interactions among individuals and groups in various domains (e.g., politics, education, health). Users have started writing and sharing their opinions online, and often base their choices on collective reasoning [27]. However, due to the increasing amount of reviews being collected online, finding helpful opinions among this collective intelligence is becoming harder, thus limiting the power of this source of information [13].

Intelligent systems are expected to offer promising solutions in order to support individuals in many existing real-world situations, both online and onlife [18, 28, 9, 8]. Current implementations have reached impressive accuracy for several tasks, but it still remains under-explored their application for assisting people who need opinions on a specific topic (e.g., a product, a disease, and so on).

⋆ All authors contributed equally to this research.

Tackling this goal is requiring to create novel strategies for understanding texts and learning patterns that characterize helpful reviews [14, 3].

When such intelligent interfaces target individuals who are looking for opinions on healthcare topics, filtering timely and valuable user-generated content is even more critical, given its heterogeneity and inner-complexity [20]. Before approaching doctors, individuals tend to search symptoms and cares online, and often struggle to detect accurate and valid health opinions. Recent studies report that around 1% of Google queries refer to symptoms questions [25]. As healthcare online reviews promise to play a primary role, it becomes imperative for intelligent systems to be able to suggest reviews based on their helpfulness.

In general, this underlying non-personalized recommender would suggest reviews whose helpfulness is high. Existing literature in healthcare usually tackled helpfulness prediction as a binary classification task, i.e., 1 if the review is considered helpful, 0 otherwise [2, 19, 22]. While optimizing models for such a task might help to filter out unhelpful reviews, they still lead to overwhelming lists, if the number of reviews marked as helpful becomes large. Moreover, the probability scores returned by the binary classifier can be hardly exploitable to create helpfulness-based rankings, as a binary classifier does not have knowledge on the difference in helpfulness among reviews marked with the same label.

In this paper, we introduce a new deep learning model optimized for predicting a continuous helpfulness score for healthcare reviews. For each review, the proposed model retrieves the corresponding word embedding representations, that are then manipulated by consecutive recurrent and attention-based layers. The resulting predicted score can be used to rank reviews about healthcare topics based on their helpfulness. While deep learning for natural language regression has been widely investigated for predicting sentiment scores or ratings as examples [5], there is limited knowledge on how it can be applied on helpfulness targets in the healthcare domain. Hence, the contribution of this paper is four-fold:

- We propose a deep learning architecture tailored for helpfulness prediction of healthcare reviews.
- We enable an optimization of the proposed architecture for regression, predicting a continuous score of helpfulness.
- We experiment with various pre-trained word embeddings created through *Word2Vec*, *GloVe*, and *FastText* algorithms.
- We show on a real-world healthcare dataset that optimizing the proposed architecture for regression leads to accurate prediction and better helpfulness-based rankings.

The rest of this paper is organized as follows. Section 2 discusses related work in the healthcare domain. Section 3 formalizes the problem we deal with. The proposed architecture, the underlying word embedding representations, and workflow followed to built the model is presented in Section 4. Results and discussion are reported in Section 5. Section 6 provides remarks and future works.

2 Related Work

Learning to rank items is a common task investigated by the recommender system community, which is not new to studies carried out in the healthcare domain. For instance, in the last four years, the ACM Conference on Recommender Systems *RecSys*[3] hosted a workshop targeting healthcare, *Health RecSys*[4]. The proposed recommenders mainly dealt with clinical data, and aimed to suggest to users health services based on their symptoms [7], well-being activities depending on their attitudes [1], food choices for a healthier diet [17], as examples.

In the healthcare domain, Electronic Health Records (EHR) are usually mined to model users' characteristics and deliver insightful recommendations to them [4]. Such systems are often designed to handle ambiguous medical situations raised by the variety of decisions that healthcare providers may take. For instance, the authors in [15] developed a patient-focused recommender system aimed to rank and suggest therapies based on collective EHR. To this end, they adopted an ensemble model that combined a Bayesian Network (BN) and a Random Forest (RF), as learning algorithms. Similarly, textual descriptions were leveraged by [6] for providing medical recommendations. They used Cognitive Computing and Semantic Web tools to mine features from clinical notes, which are then fed into clustering algorithms. Given a new clinical note, the resulting model can retrieve the most suitable diagnosis for the input clinical patient' description. Based on medical claims, demographics, symptoms, and specific users' information, other learning-to-rank models delivered personalized therapy recommendations. Online content related to user-doctor interactions was mined by [29, 11] to recommend the most appropriate doctors for a patient.

The aforementioned recommender systems usually combined knowledge from past interactions among users and items and the related descriptive attributes in order to anticipate the future interest of users (i.e., since you have interacted with these items, you might be interested in these); this would clearly not support us towards the goal of predicting the helpfulness of a review. What triggers our non-personalized recommendation is the degree of helpfulness of a review, not the social relation between users and items. Our method aims to predict the helpfulness score of a review to create helpfulness-based rankings, that may be used by intelligent systems (e.g., healthcare conversational interfaces).

Existing approaches that predict review helpfulness usually embed a binary classification model that receives features peculiar of the applicative domain [21, 26, 13, 24]. For instance, predicting e-commerce review helpfulness capitalized on user-item predispositions, user-reviewer idiosyncrasy, product nature, social network connections, and so on [23]. Similarly, other works integrated helpfulness score prediction as a preliminary task that served to improve classical user-item recommendation [10]. While all the approaches were proved to be accurate to some extent, their core principles and features tailored to the e-commerce domain

[3] https://recsys.acm.org/
[4] https://healthrecsys.github.io/

would have limited applicability in the healthcare domain. This motivated us to investigate approaches for helpfulness prediction in this under-explored domain.

3 Problem Formalization

In this section, we provide details on how we tackled the challenge of ranking healthcare reviews based on their potential helpfulness for users. We also describe how to build a machine learning model able to rank a large collection of online reviews about healthcare topics. More precisely, given a ground-truth ranking of reviews L sorted by descending true number of helpfulness votes, our goal is to find a model that produces a helpfulness-based ranking $\tilde{L}$ by maximizing the ranking quality function $nDCG(L, \tilde{L})$, where $nDCG$ (normalized Discounted Cumulative Gain) measures the quality of a ranking based on a ground-truth ranking [16]. To achieve this goal, we could deal with models optimized for two main tasks, namely *Binary Classification* and *Regression*.

Binary Classification. With this traditional task, we seek to rank reviews based on the output of a model able to categorize each review as *helpful* (label 1) or *unhelpful* (label 0). Formally, let $R \in W^*$ be the domain of reviews with unknown length, with W as a vocabulary of considered words. We apply a two-step processing pipeline with a function $f_\theta : R \rightarrow P \in [0, 1]$ that computes the class probability of a review $r \in R$ to belong to the class *helpful*, and a function $g_\theta : P \in [0, 1] \rightarrow C \in \{0, 1\}$ that assigns a binary class label to the review $r \in R$.

Regression. With this under-explored task, we seek to rank reviews based on the output of a model able to predict the number of helpfulness votes that users assign to each review. Formally, let $R \in W^*$ be the domain of reviews with unknown length, with W as a vocabulary of considered words. We consider a function $f_\theta : R \rightarrow S \in [0, S_{max}]$, where $s \in S$ represents the predicted number of helpfulness votes that users would assign to $r \in R$ in the range $[0, S_{max}]$.

For each of the two tasks, the set of all reviews $\{r_0, \ldots, r_n\}$ and the set of predicted helpfulness scores $\{f(r_0), \ldots, f(r_n)\}$ are used to build a helpfulness-based ranking $\tilde{L} = [r_i, \ldots, r_j]$, where the score $f(r_k)$ of review r_k at position k is equal or higher than the score $f(r_{k+1})$ of review r_{k+1} at position $k + 1$.

4 The Proposed Approach

In this section, we describe the deep learning model we propose to compute helpfulness binary labels or continuous scores. Moreover, we detail all the steps performed to pre-process reviews and map them to the corresponding word embeddings. Our approach is summarized in Fig. 1.

4.1 Pre-processing Module

The *Pre-processing Module* has the purpose to prepare the input texts to be fed into the deep learning model. Its input is a set of pairs $R = \{(r_0, s_0), \ldots, (r_n, s_n)\}$

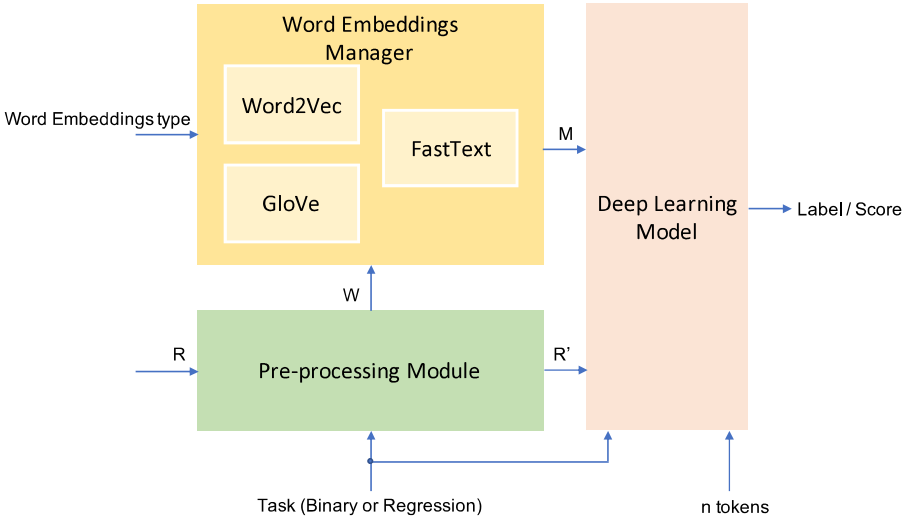

Fig. 1. The proposed approach for helpfulness prediction.

where r_i represents a textual review, and s_i is its helpfulness score. The module returns a set of pairs $R' = \{(r'_0, s_0), \ldots, (r'_n, s_n)\}$ where each r'_i is an integer-encoded review. More precisely, let W be the set of all words of the input corpus. The module first builds a function ϕ to map each word $w \in W$ to a unique integer number p, where $p \in \{0, \ldots, |W|\}$. Then, ϕ is applied on each word of a review yielding the integer-encoded representations. For example, consider the sentence *"My doctor suggested me mirtazapine"* and assume to have a toy function ϕ_{toy} that maps "my" to "11", "doctor" to "19", "suggested" to "41", "me" to "16", and "mirtazapine" to "31". Then, the integer-encoded sentence is the list $[11, 19, 41, 16, 31]$. In case of a *Binary Classification* task, the helpfulness scores are converted into binary labels, i.e., the module uses an input threshold th where scores equal or higher than th become 1, and all the others become 0.

4.2 Word Embeddings Manager

The *Word Embeddings Manager* aims to build a matrix M, where each row corresponds to the word embedding of a word $w \in W$. The module can load three types of pre-trained word embeddings generated by the following algorithms:

– *Word2Vec*[5] aims to detect the semantics of words by exploiting neural networks that mine co-occurrences of words in a given corpus. We used pre-trained word embeddings of size 300 trained on the Google News dataset.

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

– *GloVe*[6] builds a co-occurrence matrix for the input corpus and, then, a factorization approach yields the output word embeddings. Our module adopts word embeddings of size 300 with a vocabulary of 400 thousand words.
– *FastText*[7] does not use words as the smallest item of a text, but considers words as n-gram representations while learning word representations. Our module employed word embeddings of size 300 trained on a Wikipedia dataset with a vocabulary of 1 million thousand words.

In order to build M, the module receives the type of word embeddings and the set of all words W. Then, it builds a matrix M with shape $(|W|, 300)$ where each row with index $\phi(w) \mid w \in W$, i.e., $row_{\phi(w)}$, contains the word embedding of the word w. If a word w is not present in the considered input resource, then $row_{\phi(w)}$ is a row with all entries to 0.

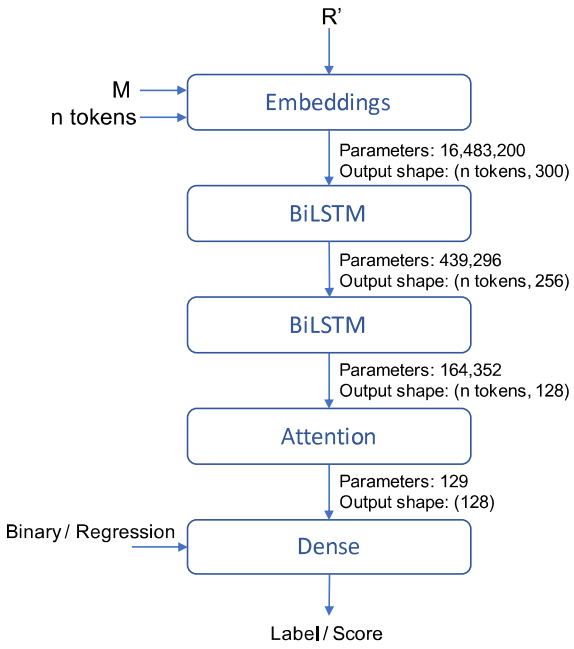

Fig. 2. The architecture of the proposed deep learning model. Layers input parameters are reported on the left. The number of inner parameters and the output shape of each layer are reported between layer blocks.

[6] https://nlp.stanford.edu/projects/GloVe/
[7] https://s3-us-west-1.amazonaws.com/FastText-vectors/wiki.en.vec

4.3 Deep Learning Model

The *Deep Learning Model* employed an underlying architecture inspired by [5], where remarkable performances were obtained for the task of sentiment prediction. Its structure is depicted in Fig. 2. After an *Input layer* that accepts integer-encoded reviews of a given maximum length, the model presents an *Embedding layer*, whose weights were initialized with the matrix M. Such weights were frozen, i.e., the initialized weights were not updated during model training. The *Embeddings layer* receives each of the integer-encoded reviews R', and returns a two-dimensional matrix, where each row is the word embedding vector of a word w in the input integer-encoded review $r \in R'$.

Then, the model mounts two *Bidirectional Long Short-Term Memory (BiLSTM) layers*. Each BiLSTM layer trains two LSTM networks by parsing the input in both forward and backward directions. This allows to have a more complete understanding of patterns behind the input data in order to deliver the final predictions. Each BiLSTM layer returns as output the concatenation of the outputs of the inner LSTM networks. By stacking two BiLSTM layers consecutively, the complexity of the model is gradually reduced. This helps us to capture knowledge at different levels of granularity.

Then, a sequential *Attention layer* is integrated, so that we consider the contextual dependencies within the BiLSTM output vector. More precisely, we employed an attention mechanism[8] that computes an additive attention, considering long-range dependencies within hidden states of the BiLSTM layer on top of it. Finally, the model presents a *Dense layer*, which is a fully-connected neural network that receives the output of the attention mechanism and delivers the model predictions. It can be configured to provide binary labels or continuous scores based on the task being approached.

5 Experimental Evaluation

In this section, we empirically evaluate the effectiveness of the proposed model on predicting the helpfulness of a review and on ranking reviews based on their potential helpfulness. We aim to answer three key research questions:

RQ1 How does our deep learning architecture perform on the tasks aimed at predicting a score / label of helpfulness?

RQ2 Which is the impact of the training optimization task (binary classification or regression) on the quality of the resulting helpfulness-based ranking of reviews?

RQ3 On the task aimed at ranking reviews based on their helpfulness, how does the model ranking performance change while considering different helpfulness thresholds th?

[8] https://pypi.org/project/keras-self-attention/

5.1 Experimental Setup

Dataset Even though health research is receiving great attention, technological, societal, and legislative constraints are currently limiting the flow of health data for research purposes. To the best of our knowledge, Drugs.com is among the first data sets that attaches helpfulness scores to health online reviews [12]. Collected from a well-known pharmaceutical website, it provides 215,063 reviews on drugs with related condition and a score (avg: 28; std. dev. 36; min: 1; max: 1,291) representing the number of users who mark a given review as useful. In our experiments, to partially limit the fact that new reviews may achieve a small number of marks even if they convey useful information, we discarded reviews published from less than one year (i.e., after Jan 2016). This helped us to keep comparable the helpfulness scores without loosing in generalizability. The distribution of the number of helpfulness votes per review is depicted in Fig. 3.

Protocols Following a 5-fold cross-validation setup, each of the considered models was trained on 80% of the reviews part of the train set, and tested on the remaining 20% included in the test set, for each fold. Indeed, 10% of the reviews in the train set were used for validation. Each model was served with batches of 1024 (review, helpfulness score) pairs for 50 epochs. For the regression task, the helpfulness scores were normalized in the range [0,1] through min-max normalization. For the binary classification task, the helpfulness scores were binarized based on a threshold of $th = 15$. This value made it possible to get the number of samples per label balanced. Early stopping procedures monitoring validation metrics (i.e., mean squared error for regression and accuracy for binary classification) were applied in order to avoid overfitting, with a patience of 5 epochs.

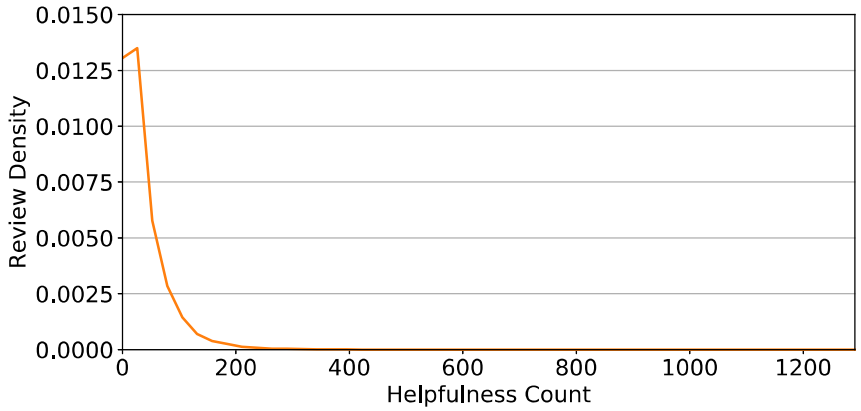

Fig. 3. The distribution of the number of helpfulness votes each review has received in the *Drugs.com* dataset.

The optimizer used for gradient update was Adam configured with a learning of 0.01 and a momentum of 0.99. Models were coded in Python on top of Keras, and trained on NVIDIA GPUs.

Metrics Once trained, each model was tested in order to assess its effectiveness on predicting the helpfulness score / label of a given review. To this end, for the binary classification task, we measured the base performance of the classifier through precision, recall, and f-measure, computed as follows:

$$P = \frac{TP}{TP+FP} \; R = \frac{TP}{TP+FN} \; F = 2\frac{R \cdot P}{R+P}$$

where TP (True Positives) is the number of helpful reviews correctly classified as helpful, FP (False Positives) is the number of unhelpful reviews that have been incorrectly classified as helpful, and FN (False Negatives) is the number of helpful reviews that are classified as unhelpful. It follows that, while comparing two models, the higher the precision, recall, f-measure, the better the model effectiveness.

For the regression task, we measured the mean squared error (MSE) and the mean absolute error (MAE) achieved by each model, as follows:

$$MAE(y, \hat{y}) = \frac{1}{n} \cdot \sum_{i=0}^{n-1} |y_i - \hat{y}_i| \; MSE(y, \hat{y}) = \frac{1}{n} \cdot \sum_{i=0}^{n-1} (y_i - \hat{y}_i)^2$$

where y_i is the true helpfulness score, $\hat{y}_i$ is the predicted helpfulness score, and n is the number of samples in the test set. It follows that, while comparing two models, the lower the MSE and MAE, the better the model effectiveness.

Performing well on helpfulness score prediction does not imply that a model is good on ranking them based on helpfulness. Therefore, in order to evaluate how each model performs on the ranking task, we measured the normalized discounted cumulative gain (nDCG), derived from the discounted cumulative gain (DCG). Through a graded relevance scale of reviews in a ranking, DCG would measure the gain of a review based on its position in the ranking. The gain is accumulated from the top to the bottom of the ranking, with the gain of each review discounted at lower ranks. First, all reviews are sorted by relative helpfulness, producing the maximum possible DCG, i.e., Ideal DCG (IDCG). The normalized discounted cumulative gain is then computed as follows:

$$DCG@k = \sum_{p=1}^{k} \frac{2^{y@p}-1}{log_2(p+1)} \; nDCG@k = \frac{DCG@k}{IDCG@k}$$

where k is the length of the considered rankings, $y@p$ is the helpfulness score of the review at position p, and $IDCG@k$ is calculated with reviews sorted by decreasing helpfulness.

Models We ran several instances of our regression model combined with 300-sized pre-trained state-of-the-art word embeddings, i.e., *Word2Vec*, *GloVe* and *FastText*. The Embedding layer of each instance was initialized with a different set of word embeddings, and its weights were kept frozen while training. This led to the following proposed models:

- **Word2Vec_Regression**: our model architecture trained on regression and initialized with *Word2Vec* word embeddings at the Embedding layer.
- ***GloVe*_Regression**: our model architecture trained on regression and initialized with *GloVe* word embeddings at the Embedding layer.
- ***FastText*_Regression**: our model architecture trained on regression and initialized with *FastText* word embeddings at the Embedding layer.

The proposed regression models were accompanied by models trained on binary classification, i.e., the common setup for predicting review helpfulness. We performed comparative analysis against the following baselines:

- **Word2Vec_Binary**: our model architecture trained on binary classification and initialized with *Word2Vec* word embeddings at the Embedding layer.
- ***GloVe*_Binary**: our model architecture trained on binary classification and initialized with *GloVe* word embeddings at the Embedding layer.
- ***FastText*_Binary**: our model architecture trained on binary classification and initialized with *FastText* word embeddings at the Embedding layer.

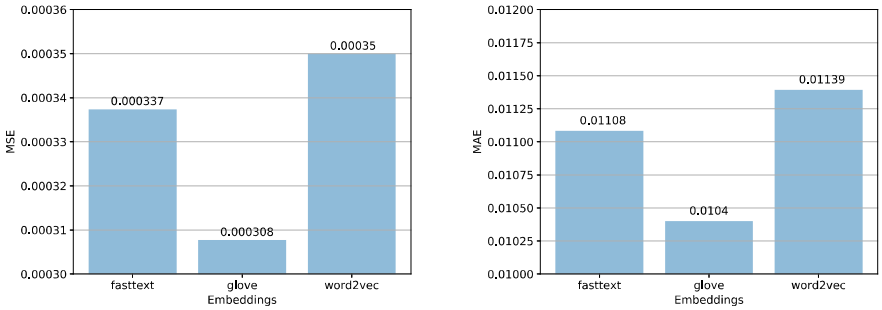

Fig. 4. Effectiveness of the propose model optimized for regression on helpfulness score prediction under different word embedding setups, measured through mean squared error (**left**) and mean absolute error (**right**). The lower the better.

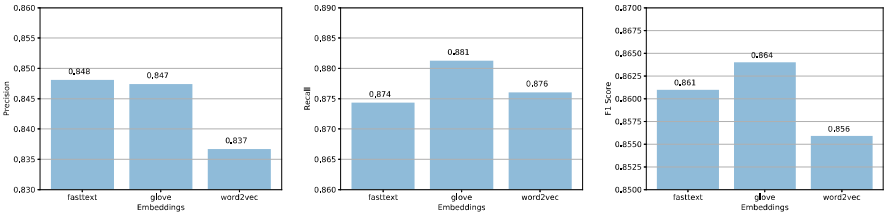

Fig. 5. Effectiveness of the proposed model optimized for binary classification on helpfulness label prediction under different word embedding setups, measured through precision (**left**), recall (**center**), and f-measure (**right**). The higher the better.

5.2 RQ1: Effectiveness on Helpfulness Prediction

In this subsection, we present results from experiments conducted to assess whether the proposed model has coherent helpfulness prediction performance. We also include an ablation experiment for assessing the impact of the word embedding collection in prediction performance. A comparison between *Word2Vec*-, *GloVe*-, and *FastText*-based models is provided in order to answer to RQ1.

Results on MSE and MAE of our models optimized for regression are shown in Figure 4. We can observe that all designed configurations had a low MSE and MAE prediction error, showing up that the proposed models can coherently quantify the helpfulness score of a review based on its content. However, based on the word embedding collection, the relative gap in prediction error among models was statistically significant ($p - value = 0.05$). Our model initialized with *GloVe* embeddings achieved the lowest prediction error (MSE=0.000308; MAE=0.01134), with a relative improvement of 10% against *FastText* and 15% against Word2Vec on both metrics. Surprisingly, this finding was in contrast with previous studies from other domains (e.g., education) and tasks (e.g., sentiment prediction), where it was proved that the best performers usually integrate Word2Vec word embeddings. It follows that, compared with Word2Vec and *FastText*, *GloVe* pre-trained word embeddings might better fit the helpfulness prediction task, and can achieve good results with health data.

Results on precision, recall and f-measure of our models optimized for binary classification are shown in Figure 5. It can be observed that all the models achieved comparable performance. As for the regression task, *Glove* word embeddings enable our model architecture to reach better performance than the other word embeddings. Models trained with *Glove* and *FastText* word embeddings outperformed the model trained on *Word2Vec* word embeddings of around 3% in terms of precision and recall. The *Glove*-based model achieved significantly better f-measure than the other ones, meaning that it might be exploited in the healthcare domain even for binary classification tasks related to helpfulness.

To have a more detailed picture, we analyzed the distribution of the helpfulness scores the models predicted (Fig. 6). As per construction, helpfulness scores predicted by regression-based models tend to follow a distribution similar to the one of the helpfulness scores in the original dataset. This might be considered as

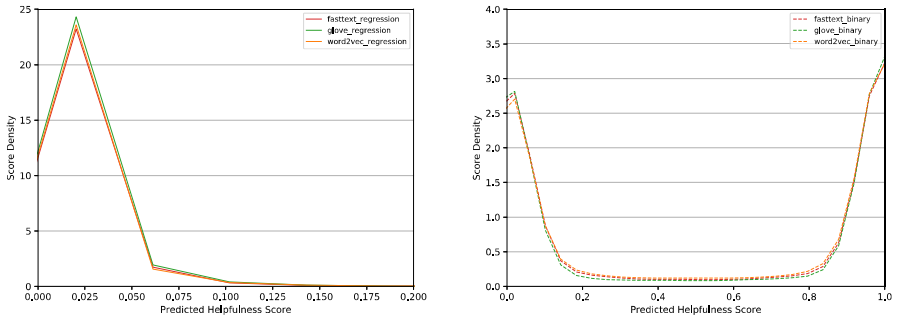

Fig. 6. The distribution of the helpfulness scores predicted under regression (left) and binary classification (right) scenarios.

a positive signal towards good helpfulness-based rankings. On the other hand, while models optimized for binary classification clearly separate scores between helpful and unhelpful reviews, they might fail to rank reviews based on helpfulness as they have no knowledge about helpfulness differences on reviews marked with the same label. Overall, the proposed architecture successfully supports both regression and binary classification tasks.

5.3 RQ2: Effectiveness on Helpfulness-based Ranking

We investigated here how effectiveness in helpfulness prediction translates to good helpfulness-based rankings. To answer this question, we report the normalized discounted cumulative gain for different ranking lengths and models in Fig. 7. For the sake of readability, to compute Ideal nDCG, we considered reviews with at least $th = 15$ votes as helpful; other reviews were marked as unhelpful.

From Figure 7, the proposed models (straight lines) can achieve almost full nDCG for small cut-offs (e.g., 10, 20, 50), and beat binary-based models (dashed lines) by large extent. It follows that our proposed models optimized for regression might bring to more effective rankings of reviews based on their helpfulness. Furthermore, while nDCG scores linearly went down at increasing cut-offs for regression-based models, binary-based models achieved stable results across them. Interestingly, we can observe that regression- and binary-based models achieved comparable results at quite large cut-offs, i.e., $4,657$, and binary-based models started to perform better than regression-based ones for higher cut-offs. It is worth to note that, from a practical user's perspective, a model that performs well at smaller cut-offs is preferable to models that work better at large cut-offs; users generally inspect few pages of reviews, each with 10-20 reviews.

It should be noted that our models can be also applied in real contexts for ranking reviews about drugs related to rare conditions, which may achieve a small number of marks even if they convey useful information in the training and testing datasets. To do this, as an example, reviews could be first filtered

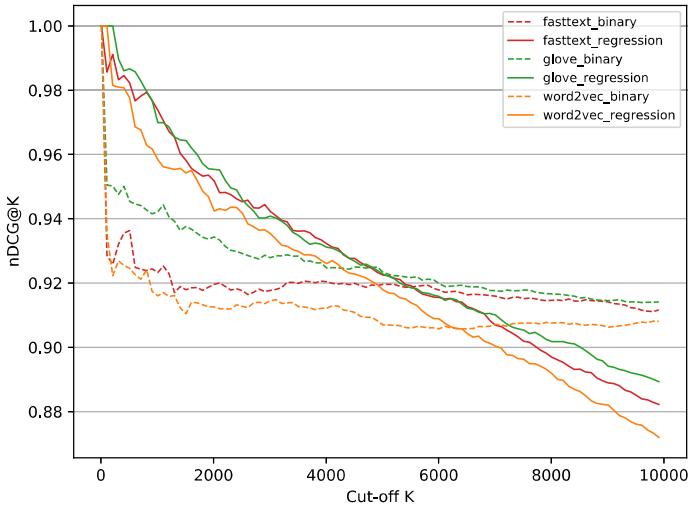

Fig. 7. Effectiveness of the models on helpfulness-based review ranking, measured through normalized discounted cumulative gain at increasing cut-offs. Regression models are identified by straight lines, while binary classification models are marked with dashed lines. The higher the nDCG, the more effective the model.

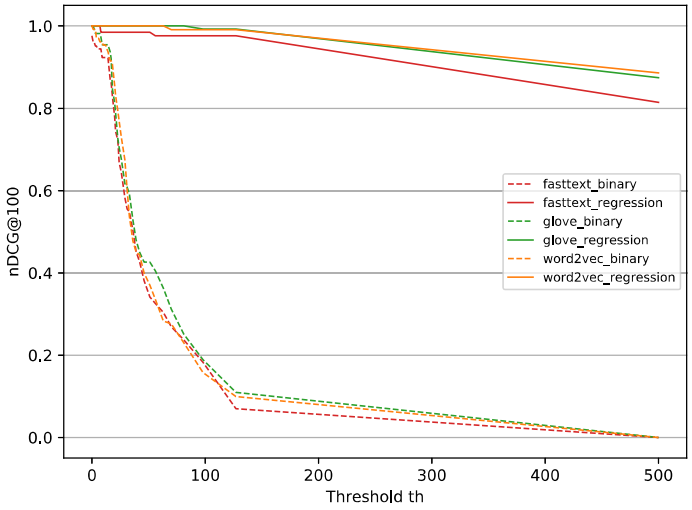

Fig. 8. Normalized discounted cumulative gain with cut-off $k = 100$, varying the helpfulness threshold th.

based on the rare condition of interest for the user, then our model can be used to predict helpfulness and rank such filtered reviews only.

5.4 RQ3: Impact of Helpfulness Threshold

We investigated how the models perform when we vary the threshold th used to separate reviews between helpful and unhelpful while computing Ideal nDCG. For conciseness, we only reported experimental results on a cut-off of 100, but note that the results on other cut-offs had similar patterns.

We vary the helpfulness threshold, and plot the results on nDCG@100 in Figure 8. The x-axis coordinates indicates the value of the helpfulness threshold, while the y-axis shows the value measured for nDCG at a cut-off of 100 for that threshold. Results plotted in the figure showed that, with larger helpfulness thresholds, model effectiveness drastically decreased for binary-based models, while it remained stable across regression-based models. It follows that the latter models better learn helpfulness-related patterns within review texts. The curves of binary-based models highlight the fact that such models are failing in predicting helpfulness relevance, especially for those reviews with a lot of votes.

6 Conclusions

In this paper, we introduced a deep learning model aimed to predict the helpfulness of a health-related review. This model mainly manipulated word embedding representations of words in a review through two BiLSTM and one Attention layer, learning patterns suitable for helpfulness prediction. Experiments on a real-world data set demonstrated that the proposed model can accurately assign helpfulness scores to reviews and be used to build rankings based on review helpfulness. Based on the results, we can conclude that:

- The proposed model leads to low errors while predicting healthcare review helpfulness, and is a feasible solution to be embedded into intelligent interfaces for supporting users in quantifying the helpfulness of reviews they look for.
- Pre-trained *GloVe* word embeddings enabled the proposed model to predict more accurate scores of helpfulness, compared with *Word2Vec* and *FastText* word embeddings.
- The proposed model optimized for regression leads to good-quality ranking of healthcare reviews based on decreasing helpfulness.
- Models optimized for regression can provide good-quality rankings for small cut-offs, and this would better fit real-world scenarios where users explores only the top-ranked results.
- By increasing the helpfulness threshold used for computing Ideal nDCG, models optimized for regression keep stable ranking performance, meaning that they put reviews with a high number of votes in top positions.

In our next steps, we are interested in temporal-based and domain-specific helpfulness prediction, which respectively take the time a review is released and

the specific healthcare domain of the review into account, when treating helpfulness. In addition, we would also investigate deeply new ways to manage reviews about topics related to rare conditions, which may achieve a small number of marks even if they convey useful information. To this aim, we may include ontological resources to embed background knowledge within our methodology in order to weigh usefulness scores. We also plan to conceive more fine-grained helpfulness predictions that can be tuned on specific target healthcare categories and can detect the specific parts of a review triggering helpfulness. On the side of applied technologies, we aim to investigate more how the employed word embedding models perform and come up with a comprehensive understanding on why GloVe returned the lowest prediction error and, overall, the best performance when compared to Word2Vec and FastTex representations. Finally, it is planned to embed the models within assistant robots working in real-world scenarios, and query them through conversational interfaces.

Acknowledgement

The research leading to these results has received funding from the EU's Marie Curie training network PhilHumans - Personal Health Interfaces Leveraging HUman-MAchine Natural interactionS under grant agreement 812882.

References

1. Alcaraz-Herrera, H., Palomares, I.: Evolutionary approach for'healthy bundle'wellbeing recommendations (2019)
2. Alodadi, N., Zhou, L.: Predicting the helpfulness of online physician reviews. In: 2016 IEEE International Conference on Healthcare Informatics (ICHI). pp. 1–6. IEEE (2016)
3. Chua, A.Y., Banerjee, S.: Understanding review helpfulness as a function of reviewer reputation, review rating, and review depth. Journal of the Association for Information Science and Technology **66**(2), 354–362 (2015)
4. Consoli, S., Recupero, D.R., Petkovic, M. (eds.): Data Science for Healthcare - Methodologies and Applications. Springer (2019). https://doi.org/10.1007/978-3-030-05249-2
5. Dessì, D., Dragoni, M., Fenu, G., Marras, M., Recupero, D.R.: Evaluating neural word embeddings created from online course reviews for sentiment analysis. In: ACM/SIGAPP Symposium on Applied Computing. pp. 2124–2127. ACM (2019)
6. Dessì, D., Recupero, D.R., Fenu, G., Consoli, S.: A recommender system of medical reports leveraging cognitive computing and frame semantics. In: Machine Learning Paradigms, pp. 7–30. Springer (2019)
7. Duan, L., Street, W.N., Xu, E.: Healthcare information systems: data mining methods in the creation of a clinical recommender system. Enterprise Information Systems **5**(2), 169–181 (2011)
8. Fenu, G., Garau, P.: Rfid-based supply chain traceability system. In: 2009 35th Annual Conference of IEEE Industrial Electronics. pp. 2672–2677. IEEE (2009)
9. Fenu, G., Nitti, M.: Strategies to carry and forward packets in vanet. In: International Conference on Digital Information and Communication Technology and Its Applications. pp. 662–674. Springer (2011)

10. Ge, S., Qi, T., Wu, C., Wu, F., Xie, X., Huang, Y.: Helpfulness-aware review based neural recommendation. CCF Tran. on Pervasive Computing and Interaction **1**(4), 285–295 (2019)
11. Glinos, M., Dahlberg, S., Tselas, N., Papapetrou, P.: Findmydoc: a p2p platform disrupting traditional healthcare models and matching patients to doctors. In: Proceedings of the 9th ACM International Conference on PErvasive Technologies Related to Assistive Environments. p. 53. ACM (2016)
12. Gräßer, F., Kallumadi, S., Malberg, H., Zaunseder, S.: Aspect-based sentiment analysis of drug reviews applying cross-domain and cross-data learning. In: Proc. of the 2018 Intern. Conference on Digital Health. pp. 121–125. ACM (2018)
13. Hong, H., Xu, D., Wang, G.A., Fan, W.: Understanding the determinants of online review helpfulness: A meta-analytic investigation. Decision Support Systems **102**, 1–11 (2017)
14. Huang, A.H., Chen, K., Yen, D.C., Tran, T.P.: A study of factors that contribute to online review helpfulness. Computers in Human Behavior **48**, 17–27 (2015)
15. Jamshidi, S., Torkamani, A., Mellen, J., Jhaveri, M., Pan, P., Chung, J., Kardes, H.: A hybrid health journey recommender system using electronic medical records. Age **46**(55), 56–65 (2018)
16. Järvelin, K., Kekäläinen, J.: Cumulated gain-based evaluation of ir techniques. ACM Transactions on Information Systems (TOIS) **20**(4), 422–446 (2002)
17. Khan, M.A., Rushe, E., Smyth, B., Coyle, D.: Personalized, health-aware recipe recommendation: An ensemble topic modeling based approach. preprint arXiv:1908.00148 (2019)
18. Klopfenstein, L.C., Delpriori, S., Malatini, S., Bogliolo, A.: The rise of bots: A survey of conversational interfaces, patterns, and paradigms. In: Proceedings of the 2017 Conference on Designing Interactive Systems. pp. 555–565. ACM (2017)
19. Krishnamoorthy, S.: Linguistic features for review helpfulness prediction. Expert Systems with Applications **42**(7), 3751–3759 (2015)
20. Laranjo, L., Dunn, A.G., Tong, H.L., Kocaballi, A.B., Chen, J., Bashir, R., Surian, D., Gallego, B., Magrabi, F., Lau, A.Y., et al.: Conversational agents in healthcare: a systematic review. Journal of the American Medical Informatics Association **25**(9), 1248–1258 (2018)
21. Lee, S., Choeh, J.Y.: Predicting the helpfulness of online reviews using multilayer perceptron neural networks. Expert Systems with App. **41**(6), 3041–3046 (2014)
22. Malik, M., Hussain, A.: Helpfulness of product reviews as a function of discrete positive and negative emotions. Computers in Human Behavior **73**, 290–302 (2017)
23. Mukherjee, S., Popat, K., Weikum, G.: Exploring latent semantic factors to find useful product reviews. In: Proceedings of the 2017 SIAM International Conference on Data Mining. pp. 480–488. SIAM (2017)
24. O'Mahony, M.P., Cunningham, P., Smyth, B.: An assessment of machine learning techniques for review recommendation. In: Irish Conference on Artificial Intelligence and Cognitive Science. pp. 241–250. Springer (2009)
25. Pinchin, V.: I'm feeling yucky: Searching for symptoms on google. Online at https://www. blog. google/products/search/im-feeling-yucky-searching-for-symptoms (2016)
26. Tang, J., Gao, H., Hu, X., Liu, H.: Context-aware review helpfulness rating prediction. In: Proceedings of the 7th ACM conference on Recommender systems. pp. 1–8. ACM (2013)
27. Wang, M., Lu, Q., Chi, R.T., Shi, W.: How word-of-mouth moderates room price and hotel stars for online hotel booking an empirical investigation with expedia data. Journal of Electronic Commerce Research **16**(1), 72 (2015)

28. Yan, H., Xu, L.D., Bi, Z., Pang, Z., Zhang, J., Chen, Y.: An emerging technology–wearable wireless sensor networks with applications in human health condition monitoring. Journal of Management Analytics **2**(2), 121–137 (2015)
29. Zhang, Y., Chen, M., Huang, D., Wu, D., Li, Y.: idoctor: Personalized and professionalized medical recommendations based on hybrid matrix factorization. Future Generation Computer Systems **66**, 30–35 (2017)