Research Paper

JIS

Journal of Information Science
1–16
© The Author(s) 2019
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/0165551519877646
journals.sagepub.com/home/jis
SAGE

A deep learning-based quality assessment model of collaboratively edited documents: A case study of Wikipedia

Ping Wang
Center for the Studies of Information Resources, School of Information Management, Wuhan University, China

Xiaodan Li
Department of Electronic and Computer Engineering, Duke University, USA

Renli Wu
School of Information Management, Wuhan University, China

Abstract

Wikipedia is becoming increasingly critical in helping people obtain information and knowledge. Its leading advantage is that users can not only access information but also modify it. However, this presents a challenging issue: how can we measure the quality of a Wikipedia article? The existing approaches assess Wikipedia quality by statistical models or traditional machine learning algorithms. However, their performance is not satisfactory. Moreover, most existing models fail to extract complete information from articles, which degrades the model's performance. In this article, we first survey related works and summarise a comprehensive feature framework. Then, state-of-the-art deep learning models are introduced and applied to assess Wikipedia quality. Finally, a comparison among deep learning models and traditional machine learning models is conducted to validate the effectiveness of the proposed model. The models are compared extensively in terms of their training and classification performance. Moreover, the importance of each feature and the importance of different feature sets are analysed separately.

Keywords

Deep learning; feature framework; information quality assessment; Wikipedia

1. Introduction

Since 2001, Wikipedia has become the most popular web-based, collaboratively edited document repository. The quality improvement of articles has been the main concern for Wikipedia [1]. Currently, there are approximately 40 million articles in more than 270 languages. In contrast to traditional media, users can not only access the information but also edit Wikipedia content immediately and arbitrarily. There are few staff members who review users' modifications before they are published [1]. Due to its large volume and flexibility, Wikipedia has attracted substantial attention from academia and industry.

Recently, Wikipedia has grown considerably. However, its growth raises a serious challenge: How good is the information quality in Wikipedia? Another issue is that only 0.1% of the articles are of high quality [1,2]. Therefore, there is a strong demand to improve article quality. First, we need to measure article quality, but there are so many articles that it is infeasible to assess every article manually. Some researchers have proposed statistics or formulas to measure article

Corresponding author:
Ping Wang, School of Information Management, Center for the Studies of Information Resources, Wuhan University, 16 Luojia Hill Road, Wuhan, Hubei 430072, China.
Email: wangping@whu.edu.cn

quality [3,4]. However, some of these metrics are so oversimplified that they cannot accurately assess article quality. In addition, some methods are not automatic and need considerable human labour. Traditional machine learning algorithms, such as support vector regression (SVR) and *k*-nearest neighbours (KNN), have also been applied [5–11]. Although there are some automatic classification methods, their performances are unsatisfactory. In addition, some methods fail to use comprehensive feature sets when training models. Moreover, some models treat the quality classification problem as a one-class classification problem, which leads to a nonexclusive classification result. Since there is a specific classifier for each class, it might result in one sample belonging to multiple classes at the same time. To solve these problems, a deep learning-based multiclass quality assessment model with a comprehensive feature framework is proposed. Our contributions are as follows.

- To the best of our knowledge, this is the first comprehensive and extensive comparison of state-of-the-art deep learning models and traditional machine learning models such as the convolutional neural network (CNN), deep neural network (DNN), long short-term memory (LSTM), CNN-LSTM, bidirectional LSTM (biLSTM), stacked LSTM, KNN, support vector machine (SVM), naïve Bayes and decision tree in terms of assessing Wikipedia article quality.
- A comprehensive feature framework is proposed and used for deep learning models.
- A detailed and complete comparison of deep learning models and several traditional machine learning models is conducted from different dimensions, including classification performance and training performance.
- The importance of different features and feature sets is investigated separately, which can provide better guidelines for feature selection.

The remainder of this article is organised as follows. Section 2 investigates the related work. Section 3 summarises the different features that represent articles. Section 4 introduces the basic concept of deep learning models and how models handle features. Section 5 introduces the Wikipedia data set and discusses the experimental results. Finally, Section 6 concludes the article.

2. Related work

In this section, we discuss how to assess the quality of Wikipedia articles using different existing approaches, and we also analyse the previous feature framework and attach the contributions of the article at the end of the section.

Wikipedia has been the most popular online encyclopaedia and knowledge database. However, the quality of the articles on Wikipedia is a serious issue since any users can edit content in Wikipedia immediately and arbitrarily. It is infeasible to estimate each article quality manually, so an efficient and automatic approach is necessary. In this section, studies on the quality assessment of Wikipedia articles are discussed.

Previous studies employed formulas or statistics to assess article quality. Some studies hypothesised that the author was the critical factor affecting article quality. Hu et al. [12] assessed Wikipedia quality based on contributions and the authority of contributors. de La Robertie et al. [3] proposed a generic formulation between authors' interactions and article quality score. The model was tested by extracting the incorporating features from a coedit graph. Several studies have attempted to quantify author reputation. Adler et al. [13] used an improved algorithm to compute the quality of English Wikipedia articles based on the reputation of original authors through the revision history of each article. The authority of the reviewers was considered [12]. In Javanmardi et al. [14], the authors derived three computational models of user reputation according to user edit patterns and statistics [15]. However, the defect is that most of these methods need too much human effort. It is infeasible to manually estimate the quality of each article, so an efficient and automatic approach is necessary.

Article stability is also considered to evaluate article quality. The literature works [16–19] proposed persistent word revisions (PWR) to count the number of revisions that a word survives. Priedhorsky et al. [20] devised a similar index called persistent word view (PWV) by calculating the retention time of an author's contribution. Suzuki and Yoshikawa [21] extended this method and considered the impact of vandals who deliberately deleted good-quality texts; they proposed a computing method that took advantage of not only the text survival rate but also the editor's qualities, but this approach is not scalable. In addition, Nemoto et al. [22] considered the preexisting social capital of editors and thought it had a positive correlation with the article quality level. However, classification performance is not preferable, especially for high-quality articles. These methods failed to consider other important features, such as article content and structure.

Intuitively, article quality is directly related to the text, so features based on the wiki page were proposed. Hardik et al. [4] conducted a detailed analysis of Wikipedia documents with some big data techniques. There are usually two types of features that are commonly used in the evaluation of article quality [23,24]. One feature is content, such as the count of various types of sentences and words. For instance, Blumenstock [25] simply chose word count as the Wikipedia quality metric and obtained good results in detecting featured articles. The other feature is structural features, such as the number and ratio of pictures, sections, paragraphs and lists in the article. Dang and Ignat [8] presented an automatic assessment model that combined these two feature sets with some traditional readability indicators into the model, such as the Flesh–Kincaid grade level [26], the Smog index [27], the Coleman–Liau index [28] and the Linsear write formula [29]. Moreover, the writing style was proposed as a metric to measure textual information. Lipka and Stein [30] pioneered the application of writing style by employing various trigram vectors to describe featured articles. Xu and Luo [31] assumed that high-quality articles had more statistical features on lexical usage, so they selected eight basic metrics to measure article quality, including the numbers of verbs, nouns and sentences.

Website link analysis was applied to assess the quality of Wikipedia pages. Kamps and Koolen [32] introduced Wikipedia link analysis. Their work suggested a possible relationship among out-links, in-links and the importance of articles. In the work described in Pateman and Johnson [33], Wikipedia articles were evaluated and corrected using inherent links. Using the MapReduce-based link analysis system, Hardik et al. [4] applied a link-ability factor to describe the diversity and expandability of Wikipedia. de Ruvo and Santone [34] investigated the influence of the article network by PageRank. In addition, the author relationship network has attracted more attention because the editors' intensive cooperative behaviours lead to high-quality articles [35]. However, various collaboration patterns among contributors have a negative effect on article quality [36]. Li et al. [37] studied relationships among article editors and quality assessment. Bykau et al. [10] applied a novel multivariable algorithm that was based on the page revision history. Their experiments on the entire English Wikipedia data set suggested that the approach had higher precision and recall than conventional approaches. With regard to these approaches, researchers usually adopt few metrics to assess article quality. Few of them consider article quality from a comprehensive perspective.

To describe an article completely, comprehensive quality metric systems have been introduced. Anderka [1] constructed a multidimensional and multilevel wiki quality evaluation system. In Dalip et al. [6], the feature framework was organised into six views. It was reduced by using the SPEA2 multiobjective genetic algorithm. Warncke-Wang et al. [38] initially adopted 17 features. They used empirical research to simplify the model. Finally, an actional model with five dimensions, including Completeness, Informativeness, NumHeadings, ArticleLength and NumReferences/ArticleLength, was established. Halfaker [39] improved Warncke-Wang's actional model by examining the dynamics of Wikipedia quality at a finer granularity through historical versions of articles. In addition, some researchers simplified quality evaluation by merging relevant metrics. For example, based on peer-reviewed data, Suzuki [40] presented a Wikipedia article assessment method that combined the h-index with the p-ratio. Ofek and Rokach [41] proposed a set of indicators that referred to meta-content features and author-based features. This model could predict whether a Wikipedia biography would be accepted with nearly 97% AUC. However, these models are not completely automatic, and the assessment still needs considerable manual intervention.

Recently, machine learning models were applied to classify Wikipedia article quality. Some models were adopted, such as SVM, KNN, multinomial logistic regression and regression trees. Wang applied a decision tree and SVM to some actionable features [38]. Dalip et al. [23,42] applied SVR to classify Wikipedia article quality. The impact of features on assessment was studied in detail. However, this model failed to achieve good performance. In Dalip et al. [5], a general multiview framework that applied a meta learning method to obtaining features was developed. The quality was thought to be a continuous value. This framework was also extended to estimate the quality of Q&A forums. In contrast to previous works, Agrawal and DeAlfaro [43] developed a quality prediction model combining LSTM and neural networks (NN) that outperformed NN. Quang-Vinh Dang et al. assessed Wikipedia quality using content format features and readability scores. However, it considers only limited Wikipedia article information [44]. Kapugama et al. [9] categorised and labelled Wikipedia search results. In their methodology, K-means clustering, and agglomerative hierarchical clustering algorithms were used to group clusters. Then, the latent Dirichlet allocation was used for labelling groups.

There are some recent studies that introduce deep learning models to assess Wikipedia article quality. In Dang and Ignat [8], doc2vec was used to represent Wikipedia articles, and DNN was applied to classify article quality. Subsequently, scholars adopted a deep learning method based on a recurrent neural network (RNN) and LSTM to achieve higher accuracy and efficiency compared with previous approaches [45]. Moreover, Shen et al. [46] created a hybrid model that combined biLSTMs with hand-engineered features. However, they fail to use a comprehensive feature framework when classifying article quality.

Our research aims to fill the following gaps. (a) There are few papers that adopt a comprehensive feature framework. They usually only take advantage of certain aspects of Wikipedia articles. (b) There is still a lack of research that has applied deep learning models to Wikipedia article quality assessment. (c) There is no extensive performance comparison of various deep learning and conventional machine learning models to classify Wikipedia article quality. (d) There are few studies that discuss selecting a better feature set to achieve satisfactory classification performance. In our research, a complete performance comparison with a comprehensive feature framework was conducted. Seven different deep learning and four traditional machine learning models are adopted in the experiment. The importance of different features or feature sets is investigated separately, which can provide better guidelines for feature selection.

3. Representation of Wikipedia articles

In this section, the proposed representation of Wikipedia articles in this article is introduced. After the analysis and summarization of existing research, a comprehensive feature framework is presented with each feature and related studies listed and described.

Currently, there are many studies on how to conduct feature engineering for Wikipedia articles. However, most of this work focuses on only partial features. Few studies have analysed and summarised the existing work. In this section, we perform an extensive review of the existing feature frameworks [1,2,5,6,12,23–27,35,42,44,47–54] and propose a comprehensive feature framework as a representation of Wikipedia articles. Text statistics are indicators that measure basic article statistics [1,23], including word count and character count. Structural features describe how an article is organised. These features are summarised in the studies of Dalip et al. and Stvilia et al. [23,49]. Intuitively, the better the article structure is, the better the article will be. Writing styles quantify parts of speech and usages of different types of words [2,5,6]. The writing styles tend to represent the writing level achieved by the author. Therefore, it is highly related to article quality. Readability scores represent the grade level or education level that readers need to understand the texts [26,27,42,48,50–52]. Edit history represents the revision history for each Wikipedia article [12,23,24,35,42,49,53]. Usually, more editing will make Wikipedia articles more understandable and readable. Network features are based on a Wikipedia link graph [1,23,54]. Network features focus on relationships among different Wikipedia articles. The features are shown in Table 1. In Table 1, each feature is described and examples of research works which use the feature are listed. For each article, we extract features in Table 1 and concatenate them as a feature vector. Consequently, for each Wikipedia article, we obtain one feature vector including text statistics, article structure, writing styles, readability score, edit history and network features. Each feature for the Wikipedia article is a numeric value, so the feature vector is a vector of numeric values. This feature vector is input to machine learning models. Initially, we assume that each feature is equally important in our model, so the weights for all features are one by default. We conduct an importance analysis for features and feature sets given classification labels in the experiment section. This assigns different weights to each feature or feature set.

Table 1. Feature description.

Features	Description	Sources
Text statistics		
words_len	Total number of words in an article	[47]
sentences_len	Total number of sentences in an article	[25]
char_count	Total character count of an article	[25]
section_count	Total number of sections	[48]
subsections_count	Total number of subsections	[23]
mean_wds_per_paragraph	Average paragraph length based on word count	[54]
mean_section_len	Average section length based on number of characters	[23,54]
words_count_longest_sentence	Word count for the longest sentence in an article	[24]
abstract_len	The size of the abstract based on the character count	[54]
std_session_len	The standard deviation of the lengths of sections	[2,54]
size_largest_section	The length of the largest section based on the character count	[23,54]
size_shortest_section	The length of the shortest section based on the character count	[23,54]
Structure features		
mean_subsection_per_section	Average number of subsections in each section	[49,54]
citation_count_per_section	Average number of citations in each section	[49,54]
citation_count	Total number of citations	[23,49]
imgs_count_per_section	Average number of images in each section	[23,49]
citation_count_per_text_length	Average citation count per text length	[23,49]

(continued)

Table 1. Continued

Features	Description	Sources
links_per_section	Average link count for each section	[23,49]
links_per_text_length	Ratio between the total number of links and the text length	[23,49]
ext_links_count	This is total count of external links	[23,49]
Writing styles		
passive_sentence_count	Number of sentences written in passive voice in an article	[2,5,6]
questions_count	Number of questions in an article	
long_phrase_rate	Percentage of long phrases	[2,5,6,54]
short_phrase_rate	Percentage of short phrases	[2,5,6,42]
auxVerbs_count	Number of auxiliary verbs	[2,5,6,42]
conjunctions_rate	Ratio between numbers of conjunctions and words	[2,5,6]
sentence_count_pronoun_bgn	Number of sentences beginning with a pronoun	[2,5,6,54]
sentence_count_article_bgn	Number of sentences beginning with an article	[2,5,6]
sentence_count_conjunction_bgn	Number of sentences beginning with a conjunction	[2,5,6]
sentence_count_subconjunction_bgn	Number of sentences beginning with a subconjunction	[2,5,6]
sentence_count_interpronoun_bgn	Number of sentences beginning with an interrogative pronoun	[2,5,6,54]
nominalization_rate	Percentage of nominalizations	[2,5,6]
preposition_rate	Percentage of prepositions	[2,5,6]
to_be_rate	Percentage of 'to be' verbs	[2,5,6,54]
Readability scores		
Automated Readability Index (ARI)	Age needed to understand an article	[48,54]
Coleman–Liau Index (Coleman_Liau)	Reading level needed to understand an article	[42]
Flesch reading ease (flesh)	Difficulty of reading and comprehending an article	[50,54]
Flesch–Kincaid (Kincaid)	Modification of the Flesch reading score	[26]
Gunning Fog Index (fog)	The Score indicating how easy the text is	[51]
LIX (Lix)	The Score indicating the text difficulty in different languages	[52]
SMOG score (Smog)	Reading grade that readers must have in order to interpret the text	[27,54]
Edit history		
Age	Age of the article	[49,42]
Mean_time_between_two_review	Mean time between two reviews in the past 30 days	
mean_edit_per_user	Average number of edits per user	[23,42]
discussion_count	Total number of user discussions	[35]
ip_number	Total number of anonymous users as represented by IPs (Internet protocols)	[24]
review_number	Number of reviews made by registered/anonymous users	[53]
user_number	Number of registered users	[49]
modified_lines_percentage	Percentage of lines modified	[23]
Occasional_user_review_percentage	Percentage of reviews from occasional users	[23]
mean_review_rate	Average review rate for the last 3 months	[23]
most_active_user_review_percentage	Percentage of reviews made by the most active users	[42]
Quality_review	Quality of reviews based on the quality of reviewers	[12,42]
std_edit_per_usr	Standard deviation of edits made by a user	
reviews_number_per_day	Average number of reviews per day	[42,42]
Network features		
in_degree	Number of articles cited by one article	[1,23]
out_degree	Number that an article is cited by other articles	[1,23]
page_rank	Page rank of an article	[23,42]
Reciprocity	Number of articles that cite each other	[23,42]
Links	Number of links in an article	[23,42]
Clustering_coefficient	Ratio between the number of edges of a node and its n neighbouring nodes and the total number of edges	[23,54]
Translation_count	Count of translation the article to other languages	[23,54]

(continued)

Table 1. Continued

Features	Description	Sources
assortativity_in_degree_in_degree	Ratio between in degree of the article and in degree of articles' neighbours	[23,54]
assortativity_in_degree_out_degree	Ratio between in degree of the article and out degree of articles' neighbours	[23,54]
assortativity_out_degree_in_degree	Ratio between out degree of the article and in degree of articles' neighbours	[23,54]

4. Quality assessment models

In this section, we present quality assessment models in more detail. It starts with the introduction of input representation, followed by the description of seven deep learning models used in our experiment.

4.1. Input representation

For each Wikipedia article, the feature vector $\boldsymbol{x} \in \mathbf{R}^l$ (l is the feature size), which includes text statistics, structure features, writing style, edit history and network features, is derived. In equation (1), $\boldsymbol{x}$ is the concatenation of these features. $\oplus$ represents the concatenation operation, and x_i represents one feature from Section 3

$$\boldsymbol{x} = x_{1:l} = x_1 \oplus x_2 \oplus x_3 \ldots \oplus x_l \tag{1}$$

4.2. RNN

An RNN is a kind of NN that can memorise training information from previous time steps [55]. For each time step t, the input of an RNN includes input vector $\boldsymbol{x}_t$ at time t and hidden state vector $\boldsymbol{h}_{t-1} \in \boldsymbol{R}^m$ at time $t - 1$. $\boldsymbol{x}_t$ is the feature vector for the current Wikipedia article, while $\boldsymbol{h}_{t-1}$ is the processed feature vector from the last time step. For our training, at each time step, the feature vector for one article will be fed into the RNN. $\boldsymbol{h}_{t-1}$ is the result for the last Wikipedia article. The output of the RNN is a hidden state vector $\boldsymbol{h}_t \in \boldsymbol{R}^m$ at time t. $\boldsymbol{h}_t$ is the processed result for this article. The hidden states can be obtained by recursively applying equation (2)

$$\boldsymbol{h}_t = f(\boldsymbol{W}\boldsymbol{x}_t + \boldsymbol{U}\boldsymbol{h}_{t-1} + \boldsymbol{b}) \tag{2}$$

where $\boldsymbol{W} \in \boldsymbol{R}^{m \times l}$ is the input to the hidden states matrix. $\boldsymbol{U} \in \boldsymbol{R}^{m \times m}$ is hidden states-to-hidden states matrix. $\boldsymbol{b} \in \boldsymbol{R}^m$ is a bias vector. f is an elementwise nonlinear function. In equation (2), the RNN can take advantage of training information for the last time step. In our research, the RNN can not only consider features for this article but also features for previous articles. However, it fails to learn the long-term dependencies [55].

4.3. Bidirectional RNNs

The current RNN takes only training information from the past. However, regarding the language processing problem, the subsequent context is also important. Therefore, two RNNs are stacked. The forward layer processes the subsequent context, while the backward layer processes the past context [56]. $\boldsymbol{x}_t$ is a feature vector for each Wikipedia article. The forward layer processes Wikipedia article in the same way as an RNN, while the backward layer processes Wikipedia articles in a reverse order. Apart from the relationship between previous and current articles, this architecture can also consider the relationship from subsequent and current articles. Formulas (3)–(5) are shown as follows

$$\overrightarrow{\boldsymbol{h}}_t = \mathcal{H}\left(\boldsymbol{W}_{x\vec{\boldsymbol{h}}} x_t + \boldsymbol{W}_{\vec{\boldsymbol{h}}\vec{\boldsymbol{h}}} \vec{\boldsymbol{h}}_{t-1} + \boldsymbol{b}_{\vec{\boldsymbol{h}}}\right) \tag{3}$$

$$\overleftarrow{\boldsymbol{h}}_t = \mathcal{H}\left(\boldsymbol{W}_{x\overleftarrow{\boldsymbol{h}}} x_t + \boldsymbol{W}_{\overleftarrow{\boldsymbol{h}}\overleftarrow{\boldsymbol{h}}} \overleftarrow{\boldsymbol{h}}_{t-1} + \boldsymbol{b}_{\overleftarrow{\boldsymbol{h}}}\right) \tag{4}$$

$$\boldsymbol{y}_t = \boldsymbol{W}_{y\overleftarrow{\boldsymbol{h}}} \overleftarrow{\boldsymbol{h}} + \boldsymbol{W}_{y\vec{\boldsymbol{h}}} \overrightarrow{\boldsymbol{h}}_t + \boldsymbol{b}_y \tag{5}$$

where $\vec{\boldsymbol{h}}$ is the forward sequence. $\overleftarrow{\boldsymbol{h}}$ is the backward sequence. x is the input. y is the output. $\boldsymbol{W}$ is the corresponding matrix. $\mathcal{H}$ is the nonlinear activation function.

4.4. LSTM

To solve the long-term memory problem, LSTM is applied since it can process the training information at much earlier time steps. LSTM includes three gates: forget gate, input gate and output gate. The forget gate is used to decide which cell state is thrown away. The input gate controls the states to be updated. The output gate controls the results to be output. The output is based on the cell states with filtering conditions [57,58]. For each Wikipedia article, feature vector $\boldsymbol{x}_t$ is extracted. Then, $\boldsymbol{x}_t$ is used to obtain the input gate, output gate and forget gate to obtain representation for the current article. Equations (6)–(10) show the updates at time t. From the following equations, we find that the representation of this article includes features for previous articles

$$\boldsymbol{i}_t = \sigma\left(\boldsymbol{W}_i \boldsymbol{x}_t + \boldsymbol{U}_i \boldsymbol{h}_{t-1} + \boldsymbol{b}^{\boldsymbol{i}}\right) \tag{6}$$

$$\boldsymbol{f}_t = \sigma\left(\boldsymbol{W}_{\boldsymbol{f}} \boldsymbol{x}_t + \boldsymbol{U}_{\boldsymbol{f}} \boldsymbol{h}_{t-1} + \boldsymbol{b}^{\boldsymbol{f}}\right) \tag{7}$$

$$o_t = \sigma(\boldsymbol{W}_{\boldsymbol{o}} \boldsymbol{x}_t + \boldsymbol{U}_{\boldsymbol{o}} \boldsymbol{h}_{t-1} + \boldsymbol{b}^{\boldsymbol{o}}) \tag{8}$$

$$\boldsymbol{c}_t = \boldsymbol{f}_t \odot \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \odot tanh(\boldsymbol{W}_{\boldsymbol{c}} \boldsymbol{h}_{t-1} + \boldsymbol{U}_{\boldsymbol{c}} \boldsymbol{x}_t + \boldsymbol{b}^{\boldsymbol{g}}) \tag{9}$$

$$\boldsymbol{h}_t = o_t \odot tanh(\boldsymbol{c}_t) \tag{10}$$

where $\boldsymbol{i}_t$ is the input gate. $\boldsymbol{f}_t$ is the forget gate. o_t is the output gate. $\boldsymbol{W_i}$, $\boldsymbol{W_f}$, $\boldsymbol{W_o}$ and $\boldsymbol{W_c}$ are inputs to the hidden state matrixes. $\boldsymbol{U_i}$, $\boldsymbol{U_f}$, $\boldsymbol{U_o}$ and $\boldsymbol{U_c}$ are the hidden states to the hidden states matrixes. $\boldsymbol{b}^{\boldsymbol{i}}$, $\boldsymbol{b}^{\boldsymbol{f}}$, $\boldsymbol{b}^{\boldsymbol{o}}$ and $\boldsymbol{b}^{\boldsymbol{g}}$ are the bias vectors. σ is a sigmoid function, and *tanh* is a hyperbolic tangent function. $\odot$ is the elementwise multiplication [58].

4.5. Variants of LSTMs

4.5.1. BiLSTMs. BiLSTMs are combinations of bidirectional RNNs and LSTMs that use both past and future information. The extracted features for each article are fed into bidirectional LSTMs. The forward LSTM processes Wikipedia articles in the same way as LSTM, while backward LSTM processes Wikipedia articles in a reverse order. This model has two advantages. (a) It considers the dependence not only among previous Wikipedia articles and current Wikipedia articles but also subsequent Wikipedia articles and current Wikipedia articles. (b) It solves the long-term memory problem. When $\mathcal{H}$ in equations (3) and (4) is implemented as in equations (6)–(10), we can obtain biLSTMs [59]. In this model, one LSTM operates in the forward direction to obtain $\overrightarrow{\boldsymbol{h}_t}$, while another LSTM operates in the backward direction to obtain $\overleftarrow{\boldsymbol{h}_t}$. The hidden states from two LSTMs are concatenated as $\boldsymbol{h}_t = [\overrightarrow{\boldsymbol{h}_t}; \overleftarrow{\boldsymbol{h}_t}]$ [58].

4.5.2. Stacked LSTMs. Stacked LSTMs have multiple LSTMs. Unlike biLSTMs, all LSTMs in this model take only training information from previous time steps. The features for each Wikipedia article are put into stacked LSTMs. Due to a more sophisticated architecture, stacked LSTMs extract more important features based on the input from the Wikipedia article, which contributes to classification performance.

4.6. CNN

A CNN focuses on a data set with a grid-like topology. Usually, there are three different layers in a CNN: a convolution layer, pooling layer and classification layer [60]. The input vector $\boldsymbol{x}$ is the feature vector for each Wikipedia article. The filter $\boldsymbol{W} \in \boldsymbol{R}^{K \times B}$ (K is window size) convolutes K features as $x_{i:i+K-1}$ from the input vector $\boldsymbol{x}$ to obtain the local feature $\boldsymbol{lf}_i$ in equation (11). Equation (12) represents feature map $\boldsymbol{fm}$

$$\boldsymbol{lf}_i = f(\boldsymbol{W}.\boldsymbol{x}_{i:i+K-1} + \boldsymbol{b}) \tag{11}$$

where $\boldsymbol{b}$ is the bias vector and f is an elementwise nonlinear function

$$\boldsymbol{fm} = [lf_1, lf_2, lf_3, lf_4, \ldots, lf_{L-K+1}] \tag{12}$$

Then, we can conduct the pooling operation on the feature map. The pooling layer can effectively reduce the feature dimensions. Consequently, a CNN can yield high classification performance with less training time. The dense layer conducts classification based on the output from the pooling layer [61].

4.7. CNN-LSTM

In CNN-LSTM, the CNN is used to extract features from text representations for each Wikipedia article. Then, LSTM classifies article quality based on features from CNN [62]. Finally, the dense layer outputs the classification result. CNN-LSTM combines advantages from a CNN and LSTM, including considering long-term dependence among Wikipedia articles and better classification performance with less training time.

5. Experiments

In this section, we discuss our experiments and their results. First, we introduce the data set. Second, we review the seven deep learning models and compare four typical machine learning models in terms of classification and training performance. Finally, we present the most important features and feature set which contributes to the quality classification of Wikipedia articles.

5.1. Data set

Wikipedia articles in English are chosen for experiments. The source files are available from Wikimedia Downloads. The content and metadata are embedded in XML. To obtain text content, a wiki extractor is adopted (Wikipedia Extractor). The structural features are extracted from the source files. The revision history and network information can be obtained from the wiki data set website (Wikimedia Downloads). There are six Wikipedia quality levels: featured article (FA), A class (AC), good article (GA), B class (BC), start class (ST), and stub class (SB) [1]. After review of some samples in each class, we noticed that the differences among adjacent classes are not significant due to the manual classification. For instance, the quality of FA and AC articles is close, while the quality of AC and SB is very different. Therefore, to improve the classification performance, three quality classes, high, medium and low, are proposed. The high-quality class includes FA and AC. The medium class includes GA and BC. The low quality includes ST and SB. These three classes are distributed equally. In our experiment, only 3294 articles were selected. There are two reasons. First, some articles are too short to extract enough features. Second, the information such as edit history or reference/link relationship for many articles is incomplete. The data set is partitioned randomly into a training set and testing set with proportions of 60% and 40%, respectively. The experiment was repeated 20 times to obtain the average performance metrics.

5.2. Result analysis

In this section, the results are discussed. All experiments are implemented in Keras 2.0.8 and TensorFlow 1.1.0. For the parameters, epochs, batch size, and dropout rate are set as 15, 195 and 0.2, respectively.

5.2.1. Classification performance. Accuracy, precision, recall, F1-score and F-beta score are adopted to measure classification performance. These are informative and direct indicators of a model's performance [63,64]. Tables 2 and 3 show the classification performance of the deep learning models and conventional machine learning models, respectively.

Table 2 reports that stacked LSTMs acquire the best performance for all the metrics. This is due to its complicated model architecture. Three layers of LSTMs are stacked together, which can extract more minute but important patterns. Compared with basic LSTM, dropout leads to performance degradation. Dropout is specific for avoiding overfitting in the model [65]. It might make performance worse. Contrary to what we expect, the CNN performs the worst. In most cases, the CNN has high performance in learning relevant features and ruling out irrelevant features [54,62]. Moreover, after comparison of basic LSTM and CNN-LSTM, we find that the CNN degrades the model performance.

According to Table 3, we can determine that both the decision tree and SVM yield preferable classification performance. However, the decision tree is slightly better than SVM. The decision tree has a leading classification performance. It generates better scores than other models in terms of accuracy, precision, F1 and F-beta. Naïve Bayes is not suitable for classifying Wikipedia article quality. It has low F1 and F-beta scores.

LSTM and its variants are proficient in classifying Wikipedia article quality. Regarding conventional machine learning algorithms, both the decision tree and SVM perform better than most deep learning models except stacked LSTMs. Due to the complex model architecture, stacked LSTMs have slightly better performance than decision trees. In addition,

Table 2. Classification performance of deep learning models.

	Stacked LSTMs	DNN	CNN	CNN + LSTM	LSTM with dropout	Basic LSTM	Bidirectional LSTMs
Accuracy	**0.7185**	0.6866	0.6343	0.6730	0.6791	0.6904	0.6965
Precision	**0.7169**	0.6802	0.6087	0.6665	0.6696	0.6855	0.6905
F1	**0.7165**	0.6707	0.5973	0.6659	0.6660	0.6821	0.6880
F-beta	**0.7167**	0.6680	0.5989	0.6668	0.6658	0.6810	0.6876

LSTM: long short-term memory; DNN: deep neural network; CNN: convolutional neural network.
The bold values point out the best performance.

Table 3. Classification performance of conventional machine learning models.

	Decision tree	SVM	KNN	Naïve Bayes
Accuracy	**0.7109**	0.7079	0.6631	0.5986
Precision	**0.7172**	0.7095	0.6711	0.6460
F1	**0.7135**	0.7055	0.6659	0.5724
F-beta	**0.7156**	0.7070	0.6687	0.5948

SVM: support vector machine; KNN: *k*-nearest neighbours.
The bold values point out the best performance.

we compare our classification accuracy with other state-of-art models. The accuracy of our model is 12.3% better than the accuracy of the random forest, which is 64% [8].

In summary, stacked LSTMs are the best model among deep learning models, while the decision tree performs the best in traditional machine learning models. Except for precision, stacked LSTMs are slightly better than the decision tree for the other three metrics.

5.2.2. Confusion matrix. The confusion matrix for each model is reported in Figure 1. Generally, these models can identify low-quality articles very well, but they fail to identify the other two classes. This can be explained partially by the fact that the articles are originally classified manually. This classification is subjective, and the boundary between medium and good/low articles is not clear. The CNN and DNN perform poorly when classifying medium-quality articles. However, the DNN performs very well when classifying good-quality articles. Stacked LSTMs outperform other deep learning models when classifying medium-quality articles. However, traditional machine learning models are better at classifying medium-quality articles than deep learning models. Naïve Bayes yields preferable performance in classifying low-quality articles, while it fails to distinguish good-quality articles from others. Averagely, the decision tree has a balanced classification result, even though it is not very good at classifying low-quality articles. Comparing stacked LSTMs and decision trees, we find that stacked LSTMs are much better than decision trees when classifying low-quality Wikipedia articles, while decision trees have better performance when classifying good-quality articles. The accuracy is very close when classifying medium-quality articles for two models.

5.2.3. Training performance. In Figure 2, the accuracy of most models increases slowly at first. Then, it rises quickly and reaches a plateau. In addition, stacked LSTMs perform the best, while the CNN has the worst performance. Moreover, the training accuracy of basic LSTM is close to that of LSTM with dropout. This indicates that dropout has little impact on training performance. Figure 3 gives the training cross-entropy loss for each model. The cross-entropy is the metric that measures error probabilities in discrete classifications (Objectives). The training loss for most models decreases monotonically except for CNN-LSTM. Stacked LSTMs reach the lowest training loss, while the loss for the CNN is the highest. For CNN-LSTM, the training loss first decreases. However, its loss rises from epoch 10 to 11 and ends with a high value. Moreover, CNN performs the worst and reaches approximately 0.52. For LSTM with dropout and basic LSTM, their model training losses are almost the same. Compared with basic LSTM and LSTM with dropout, bidirectional LSTM performs better.

5.2.4. Feature importance analysis. In this section, we conduct feature importance analysis. Stacked LSTMs are applied due to their best performance. We run the model several times. Every time, we delete one feature and train models with

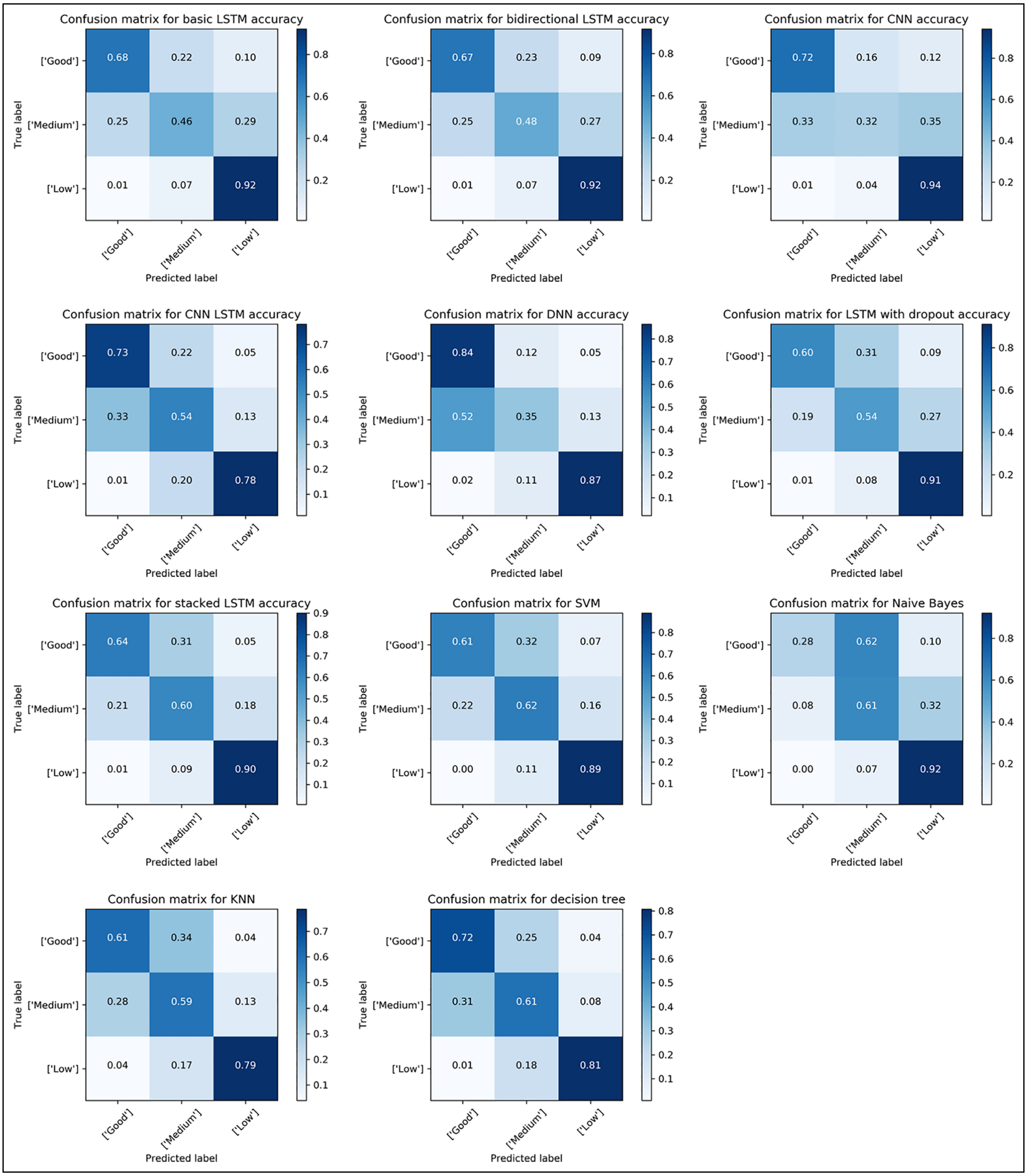

Figure 1. Confusion matrices for different models.

the other remaining features. The feature importance is different between the results of the stacked LSTMs with a complete feature set and those with a reduced feature set. The larger the feature importance is, the more important the feature is. A positive importance means that the feature contributes to the increase in classification accuracy, while a negative importance means that the feature worsens the performance.

Due to page restrictions, we show only the top ten features with the highest importance and lowest importance in Table 4. Conjunction, sub_conjunction and passive_sentence_count are the three most important features. This suggests

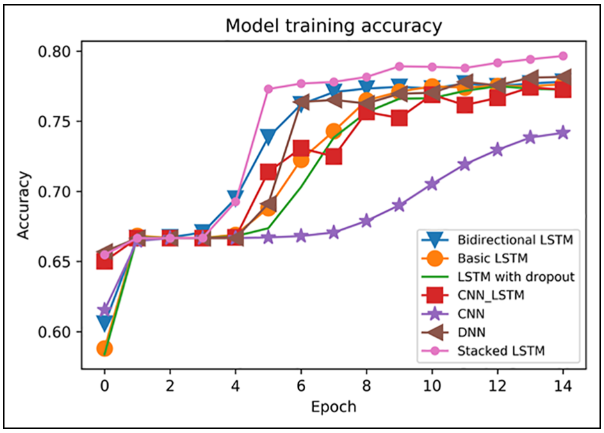

Figure 2. Model training accuracy.

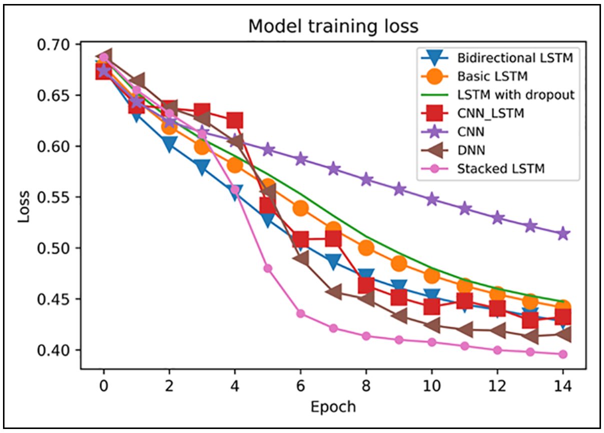

Figure 3. Model training loss.

Table 4. Feature importance.

Feature	Importance
conjunction	0.01971396
sub_conjunction	0.01592033
passive_sentence_count	0.01516161
read_smog	0.01516161
abstract_size	0.01440288
sections_count	0.01440288
pronouns	0.01440288
session_len	0.01364416
largest_section_size	0.01364416
ip_count	0.01364416
age	0.00302200
assortativity_out_in	0.00302200
read_ari	0.00226328
nominalization_rate	0.00150455
links_per_text_length	0.00074583
out_degree	0.00074583
wds_per_paragraph_avg	– 0.0007716
Clustering_coefficient	– 0.0007716
reciprocity	– 0.0007716
len_sentences	– 0.0015303

Table 5. Accuracy for each feature set.

Model	Writing style	Text statistics	Structural features	Readability scores	Network features	Edit history
Accuracy	0.6343	0.6912	0.6555	0.3414	0.6184	0.6032

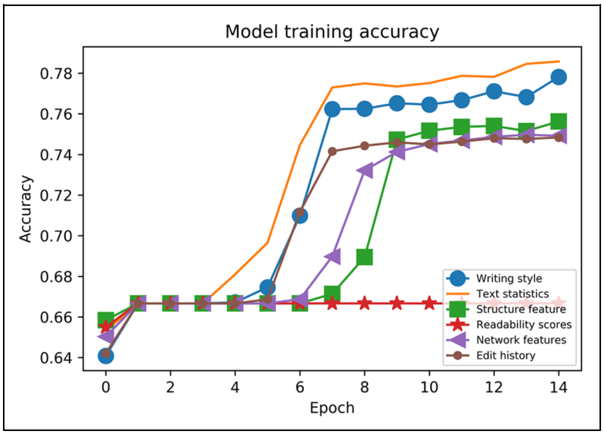

Figure 4. Model training accuracy.

that the writing style is highly related to article quality. If the author uses more conjunctions, subconjunctions and passive sentences, the article will be more sophisticated. However, len_sentences, reciprocity and clustering_coefficient are the features with the least importance. Len_sentences is the feature that contradicts our expectation. Intuitively, the better article usually has more long sentences. This is because Wikipedia is for the public with different knowledge backgrounds. The shorter sentence can also help the public to obtain knowledge much easier. Therefore, both long and short sentences are preferable for a good-quality article.

5.2.5. Feature set analysis. In this section, each feature set is analysed. The feature framework has six feature sets: writing style, text statistics, structural features, readability scores, network features and edit history. The stacked LSTMs are used due to their best classification performance.

In Table 5, we observe that text statistics outperform other feature sets. Intuitively, an article with more words tends to be of better quality. In addition, structural features yield good performance. Therefore, the length and structure of an article are highly related to its quality. However, readability scores perform the worst. This is because readability scores focus on how difficult an article is to understand and read. Generally, the feature framework with all feature sets performs better than only one feature set as a feature framework.

Figure 4 presents the model training accuracy for different feature sets. The accuracy for most models increases slowly at first and then reaches a plateau. After epoch three, the training accuracy increases sharply. It eventually achieves good training accuracy. However, unlike other feature sets, the training accuracy for the readability scores increases slowly and ends with a small value. The accuracy for text statistics is the highest in the end. Figure 5 shows that the loss for most models decreases significantly at first and reaches a plateau with a low training loss, while the loss for readability scores remains high and has only a slight decrease. The model with readability scores has a high training loss in the end. However, text statistics yield a preferable model training loss.

6. Conclusion and discussion

In this article, a novel deep learning-based quality assessment approach for Wikipedia is presented. First, based on related works, a comprehensive feature framework is summarised. The feature framework is fed into a deep learning model and traditional machine learning models: CNN, DNN, basic LSTM, LSTM with dropout, stacked LSTMs, biLSTMs, CNN-LSTM, SVM, naïve bayes, KNN and decision tree. Extensive and detailed experiments based on these models and feature frameworks are conducted. The results show that stacked LSTMs are the best model that can efficiently distinguish Wikipedia articles with different article qualities. Stacked LSTMs are the best model among deep learning models, while the decision tree performs the best in traditional machine learning models. Except for precision, stacked LSTMs are

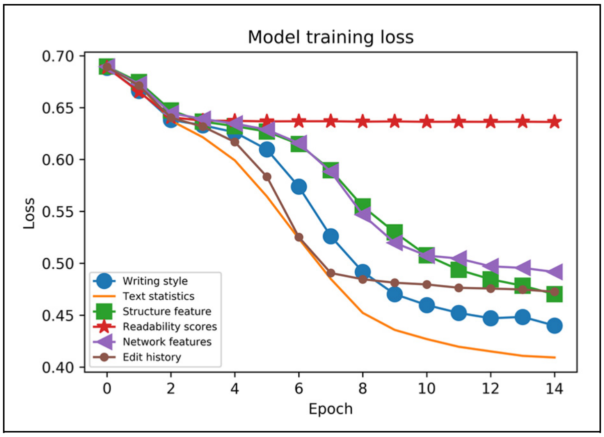

Figure 5. Model training loss.

slightly better than the decision tree for the other three metrics. According to the confusion matrix, stacked LSTMs yield much better performance when classifying low-quality articles. The decision tree has a significantly better accuracy when classifying good-quality articles. The accuracy is very close when classifying medium-quality articles with two models.

There are many theoretical and practical implications to our research. As far as we know, this is the first article that proposes an extensive comparison among different deep learning models to classify Wikipedia article quality using a comprehensive feature framework. The existing Wikipedia quality assessment models tend to adopt only partial Wikipedia features, such as content features or structural features. Our feature framework collects information for Wikipedia articles, such as content features, structural features, and author-related features. This framework can offer a more complete description of each Wikipedia article with a better article quality classification accuracy. Our findings also provide several important implications for practice. Currently, there are many Wikipedia articles. It is impossible to assess each article manually. The first practical implication is that our project proposes an automatic and practical quality classification method that can accelerate Wikipedia article quality assessments and save considerable human effort. Another critical implication is that our research has provided a guideline for the best deep learning model and feature sets to assess Wikipedia article quality. After the experiments, we found that stacked LSTMs are the best classification models. The readability scores contribute the least to the classification performance. Generally, our findings suggest that stacked LSTMs with all feature sets, including writing style, text statistics, structural features, network features and edit history, are the best classification methods for Wikipedia articles.

In terms of limitations, only approximately 3000 articles are selected. The size of the data set is insufficient for training for two reasons. First, some articles are too short to extract enough features. Second, the information related to many articles is incomplete. For example, some articles do not have a long enough edit history or network information to generate related features. Another limitation is that this classification framework is applicable to Wikipedia articles in only English since some features in text statistics and writing style are based on only English grammar.

In future work, to generalise the classification framework, we will further investigate feature design to enlarge the sample size. In addition, we will introduce some other state-of-the-art machine learning models, such as the attention model and transformer into our framework to improve the model classification performance.

Acknowledgements

The authors thank Daniel Hasan Dalip and Dr Pável Calado for their helpful guidance and support.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This research was supported by National Natural Science Foundation of China (No. 71774121) and the worldclass discipline "Library, Information and Data Science" by the Ministry of Education of the People’s Republic of China.

ORCID iD

Ping Wang https://orcid.org/0000-0003-0033-4150

References

[1] Anderka M. *Analyzing and predicting quality flaws in user-generated content: the case of Wikipedia*. PhD Thesis, Bauhaus-Universität, Weimar, 2013.

[2] Anderka M and Stein B. A breakdown of quality flaws in Wikipedia. In: *Proceedings of the 2nd Joint WICOW/AIRWeb workshop on web quality*, Lyon, 16 April 2012, pp. 11–18. New York: ACM.

[3] de la Robertie B, Pitarch Y and Teste O. Measuring article quality in Wikipedia using the collaboration network. In: *Proceedings of the 2015 IEEE/ACM international conference on advances in social networks analysis and mining*, Paris, 25 August 2015, pp. 464–471. New York: ACM.

[4] Hardik V, Anirudh V and Balaji P. Link analysis of Wikipedia documents using mapreduce. In: *Proceedings of the IEEE international conference on information reuse and integration*, San Francisco, CA, 13–15 August 2015, pp. 582–588. New York: IEEE.

[5] Dalip DH, Gonçalves MA, Cristo M et al. A general multiview framework for assessing the quality of collaboratively created content on web 2.0. *J Assoc Inf Sci Technol* 2017; 68: 286–308.

[6] Dalip DH, Lima H, Gonçalves MA et al. Quality assessment of collaborative content with minimal information. In: *Proceedings of the IEEE/ACM joint conference on digital libraries*, London, 8–12 September 2014, pp. 201–210. New York: IEEE.

[7] Sinanc D and Yavanoglu U. A new approach to detecting content anomalies in Wikipedia. In: *Proceedings of the 12th international conference on machine learning and applications*, Miami, FL, 4–7 Dec. 2013, pp. 288–293. Piscataway, NJ: IEEE.

[8] Dang Q and Ignat C. Measuring quality of collaboratively edited documents: the case of Wikipedia. In: *Proceedings of the IEEE 2nd international conference on collaboration and internet computing (CIC)*, Pittsburgh, PA, 1–3 November 2016, pp. 266–275. New York: IEEE.

[9] Kapugama KDCG, Lorensuhewa SAS and Kalyani MAL. Enhancing Wikipedia search results using text mining. In: *Proceedings of the 16th international conference on advances in ICT for emerging regions (ICTer)*, Negombo, Sri Lanka, 1–3 September 2016, pp. 168–175. New York: IEEE.

[10] Bykau S, Korn F, Srivastava D et al. Fine-grained controversy detection in Wikipedia. In: *Proceedings of the IEEE 31st international conference on data engineering*, Seoul, South Korea, 13–17 April 2015, pp. 1573–1584. New York: IEEE.

[11] Ganjisaffar Y, Javanmardi S and Lopes C. Based ranking of Wikipedia articles. In: *Proceedings of the international conference on computational aspects of social networks*, Fontainebleau, 24 June–27 June 2009, pp. 98–104. New York: IEEE.

[12] Hu M, Lim EP, Sun A et al. Measuring article quality in Wikipedia: models and evaluation. In: *Proceedings of the 16th ACM conference on information and knowledge management*, Lisbon, Portugal, 6–10 November 2007, pp. 243–252. New York: IEEE.

[13] Adler BT, Chatterjee K, de Alfaro L et al. Assigning trust to Wikipedia content. In: *Proceedings of the 4th international symposium on wikis*, Porto, Portugal, 8–10 September 2008, pp. 1–12. New York: ACM.

[14] Javanmardi S, Lopes C and Baldi P. Modeling user reputation in wikis. *Stat Anal Data Min* 2010; 3: 126–139.

[15] Javanmardi S and Lopes C. Statistical measure of quality in Wikipedia. In: *Proceedings of the first workshop on social media analytics*, Washington, DC, 25–28 July 2010, pp. 132–138. New York: ACM.

[16] Wöhner T and Peters R. Assessing the quality of Wikipedia articles with lifecycle based metrics. In: *Proceedings of the 5th international symposium on wikis and open collaboration*, Orlando, FL, 25–27 October 2009, pp. 1–10. New York: ACM.

[17] de la Calzada G and Dekhtyar A. On measuring the quality of Wikipedia articles. In: *Proceedings of the 4th workshop on information credibility*, Raleigh, NC, 27 April 2010, pp. 11–18. New York: ACM.

[18] Biancani S. Measuring the quality of edits to Wikipedia. In: *Proceedings of the international symposium on open collaboration*, Berlin, 27–29 August 2014, pp. 1–33. New York: ACM.

[19] Halfaker A, Kittur A, Kraut R et al. A jury of your peers: quality, experience and ownership in Wikipedia. In: *Proceedings of the 5th international symposium on wikis and open collaboration*, Orlando, FL, 25–27 October 2009, pp. 1–10. New York: ACM.

[20] Priedhorsky R, Chen J, Lam SK et al. Creating, destroying, and restoring value in Wikipedia. In: *Proceedings of the 2007 international ACM conference on supporting group work*, Sanibel Island, FL, 4–7 November 2007, pp. 259–268. New York: ACM.

[21] Suzuki Y and Yoshikawa M. Mutual evaluation of editors and texts for assessing quality of Wikipedia articles. In: *Proceedings of the 8th annual international symposium on wikis and open collaboration*, Linz, Austria, 27–29 August 2012, pp. 1–10. New York: ACM.

[22] Nemoto K, Gloor P and Laubacher R. Social capital increases efficiency of collaboration among Wikipedia editors. In: *Proceedings of the 22nd ACM conference on hypertext and hypermedia*, Eindhoven, 6–9 June 2011, pp. 231–240. New York: ACM.

[23] Dalip D, Gonçalves MA, Cristo M et al. Automatic quality assessment of content created collaboratively by web communities: a case study of Wikipedia. In: *Proceedings of the 9th ACM/IEEE-CS joint conference on digital libraries*, Austin, TX, 15–19 June 2009, pp. 295–304. New York: ACM.

[24] Anderka M, Stein B and Lipka N. Predicting quality flaws in user-generated content: the case of Wikipedia. In: *Proceedings of the 35th international ACM SIGIR conference on research and development in information retrieval*, Portland, OR, 12–16 August 2012, pp. 981–990. New York: ACM.

[25] Blumenstock JE. Size matters: word count as a measure of quality on Wikipedia. In: *Proceedings of the 17th international conference on World Wide Web*, Beijing, China, 21–25 April 2008, pp. 1095–1096. New York: ACM.

[26] Kincaid JP, Fishburne RP Jr, Rogers RL et al. *Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel*. Technical Report Research Branch Report 8-75. Report no. 56, February 1975. Millington, TN: Institute for Simulation and Training.

[27] Mc Laughlin GH. SMOG grading-a new readability formula. *J Read* 1969; 12: 639–646.

[28] Coleman M and Liau TL. A computer readability formula designed for machine scoring. *J Appl Psychol* 1975; 60: 283–284.

[29] Chen H-H. How to use readability formulas to access and select English reading materials. *J Educ Media Libr Sci* 2012; 50: 229–254.

[30] Lipka N and Stein B Identifying featured articles in Wikipedia: writing style matters. In: *Proceedings of the 19th international conference on World Wide Web*, Raleigh, NC, 26–30 April 2010, pp. 1147–1148. New York: ACM.

[31] Xu Y and Luo T. Measuring article quality in Wikipedia: lexical clue model. In: *Proceedings of the 3rd symposium on web society*, Port Elizabeth, South Africa, 26–28 October 2011, pp. 141–146. New York: IEEE.

[32] Kamps J and Koolen M. Is Wikipedia link structure different? In: *Proceedings of the second ACM international conference on web search and data mining*, Barcelona, 9–12 February 2009, pp. 232–241. New York: ACM.

[33] Pateman BM and Johnson C. Using the Wikipedia link structure to correct the Wikipedia link structure. In: *Proceedings of the 2nd workshop on the people's web meets NLP: collaboratively constructed semantic resources*, Beijing, China, August 2009, pp. 10–18. Stroudsburg, PA: ACL.

[34] de Ruvo G and Santone A. Analysing wiki quality using probabilistic model checking. In: *Proceedings of the IEEE 24th international conference on enabling technologies: infrastructure for collaborative enterprises*, Larnaca, Cyprus, 15 June–17 June 2015, pp. 224–229. New York: IEEE.

[35] Wilkinson DM and Huberman BA. Cooperation and quality in Wikipedia. In: *Proceedings of the 2007 international symposium on Wikis*, Montreal, QC, Canada, 21–25 October 2007, pp. 157–164. New York: ACM.

[36] Liu J and Ram S. Who does what: collaboration patterns in the Wikipedia and their impact on article quality. *ACM Trans Manag Inf Syst* 2011; 2: 1–23.

[37] Li X, Tang J, Wang T et al. Automatically assessing Wikipedia article quality by exploiting article–editor networks. In: *Advances in information retrieval* (eds Hanbury A, Kazai G, Rauber A et al.), Vienna, Austria, 2 April 2015, pp. 574–580. Cham: Springer International Publishing.

[38] Warncke-Wang M, Cosley D and Riedl J. Tell me more: an actionable quality model for Wikipedia. In: *Proceedings of the 9th international symposium on open collaboration*, Hong Kong, China, 5–7 August 2013, pp. 1–8. New York: ACM.

[39] Halfaker A. Interpolating quality dynamics in Wikipedia and demonstrating the keilana effect. In: *Proceedings of the 13th international symposium on open collaboration*, Galway, 23–25 August 2017, pp. 1–9. New York: ACM.

[40] Suzuki Y. Quality assessment of Wikipedia articles using h-index. *J Inf Process* 2015; 23: 22–30.

[41] Ofek N and Rokach L. A classifier to determine which Wikipedia biographies will be accepted. *J Assoc Inf Sci Technol* 2014; 66: 213–218.

[42] Dalip DH, Gonçalves MA, Cristo M et al. Automatic assessment of document quality in web collaborative digital libraries. *J Data Inf Qual* 2011; 2: 14.

[43] Agrawal R and DeAlfaro L. Predicting the quality of user contributions via LSTMs. In: *Proceedings of the 12th international symposium on open collaboration*, Berlin, 17–19 August 2016, pp. 1–10. New York: ACM.

[44] Dang Q and Ignat C. Quality assessment of Wikipedia articles without feature engineering. In: *Proceedings of the 2016 IEEE/ACM joint conference on digital libraries (JCDL)*, Newark, NJ, 19–23 June 2016, pp. 27–30. New York: IEEE.

[45] Dang Q-V and Ignat C-L. An end-to-end learning solution for assessing the quality of Wikipedia articles. In: *Proceedings of the 13th international symposium on open collaboration*, Galway, 23–25 August 2017, pp. 1–10. New York: ACM.

[46] Shen A, Qi J and Baldwin T. A hybrid model for quality assessment of Wikipedia articles. In: Proceedings of the Australasian language technology association workshop, Brisbane, QLD, Australia, December 2017, pp. 43–52, https://pdfs.semanticscholar.org/8946/03d927860010ed3554a9922a992838188d81.pdf?_ga=2.152178489.1047445110.1568451902-1540706140.1559042995

[47] Lee YW, Strong DM, Kahn BK et al. AIMQ: a methodology for information quality assessment. *Inf Manage* 2002; 40(2): 133–146.

[48] Blumenstock JE. Automatically assessing the quality of Wikipedia articles. School of Information, UC Berkeley, https://escholarship.org/uc/item/18s3z11b (2008, accessed 6 June 2018).

[49] Stvilia B, Michael BT, Linda CS et al. Assessing information quality of a community-based encyclopedia. In: *Proceedings of the international conference on information quality (ICIQ)*, Cambridge, MA, 10–12 November 2006, pp. 442–454. DBLP, http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.78.6243

[50] Flesch R. A new readability yardstick. *J Appl Psychol* 1948; 32(3): 221–233.

[51] Gunning R. The fog index after twenty years. *Int J Bus Commun* 1969; 6(2): 3–13.

[52] Bjômsson CH. *Lasbarhet* [Readability]. Stockholm: Liber, 1968.

[53] Lih A. Wikipedia as participatory journalism: reliable sources? Metrics for evaluating collaborative media as a news resource. *Nature* 2004; 3(1): 1–31.

[54] Wang P and Li X. Assessing the quality of information on Wikipedia: a deep-learning approach. *J Assoc Inf Sci Tech*. Epub ahead of print 8 April 2019. DOI: 10.1002/asi.24210.

[55] Abdul-Mageed M and Ungar L. Emonet: fine-grained emotion detection with gated recurrent neural networks. In: *Proceedings of the 55th annual meeting of the association for computational linguistics*, Vancouver, BC, Canada, 30 July–4 August 2017, pp. 718–728. Stroudsburg, PA: ACL.

[56] Ming H, Huang D, Xie L et al. Deep bidirectional LSTM modeling of timbre and prosody for emotional voice conversion. In: *Proceedings of the 17th annual conference of the international speech communication association*, San Francisco, CA, September 2016, pp. 2453–2457. Interspeech, https://www.isca-speech.org/archive/Interspeech_2016/abstracts/1053.html

[57] Olah C. Understanding LSTM networks, http://colah.github.io/posts/2015-08-Understanding-LSTMs/ (2015, accessed 5 June 2018).

[58] Yang B and Mitchell T. Leveraging knowledge bases in LSTMs for improving machine reading. In: *Proceedings of the 55th annual meeting of the association for computational linguistics*, Vancouver, BC, Canada, January 2017, (Volume 1: Long Papers) Jul 2017, pp. 1436–1446. Stroudsburg, PA: ACL.

[59] Fan B, Xie L, Yang S et al. A deep bidirectional LSTM approach for video-realistic talking head. *Multimed Tools Appl* 2016; 75: 5287–5309.

[60] Krizhevsky A, Sutskever I and Hinton GE. ImageNet classification with deep convolutional neural networks. In: *Proceedings of the 25th international conference on neural information processing systems*, Lake Tahoe, NV, 3–6 December 2012, pp. 1097–1105. Red Hook: Curran Associates, Inc.

[61] Oquab M, Bottou L, Laptev I et al. Learning and transferring mid-level image representations using convolutional neural networks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, Columbus, OH, 23–28 June 2014, pp. 1717–1724. New York: IEEE.

[62] Trusov R. Text classifier algorithms in machine learning, https://blog.statsbot.co/text-classifier-algorithms-in-machine-learning-acc115293278 (2017, accessed 5 June 2018)

[63] Feng W, Wu S, Li X et al. A deep belief network based machine learning system for risky host detection, 2017, https://arxiv.org/abs/1801.00025

[64] Davis J and Goadrich M. The relationship between Precision-Recall and ROC curves. In: *Proceedings of the 23rd international conference on machine learning*, Pittsburgh, PA, 25–29 June 2006, pp. 233–240. New York: ACM.

[65] Srivastava N, Hinton G, Krizhevsky A et al. Dropout: a simple way to prevent neural networks from overfitting. *J Mach Learn Res* 2014; 15: 1929–1958.