A Hybrid Model for Quality Assessment of Wikipedia Articles

Aili Shen **Jianzhong Qi** **Timothy Baldwin**
School of Computing and Information Systems
The University of Melbourne
Victoria, Australia
ailis@student.unimelb.edu.au
jianzhong.qi@unimelb.edu.au tb@ldwin.net

Abstract

The task of document quality assessment is a highly complex one, which draws on analysis of aspects including linguistic content, document structure, fact correctness, and community norms. We explore the task in the context of a Wikipedia article assessment task, and propose a hybrid approach combining deep learning with features proposed in the literature. Our method achieves 6.5% higher accuracy than the state of the art in predicting the quality classes of English Wikipedia articles over a novel dataset of around 60k Wikipedia articles. We also discuss limitations with this task setup, and possible directions for establishing more robust document quality assessment evaluations.

1 Introduction

With the advent of Web 2.0, it has become much easier to collaboratively write and distribute documents, such as Wikipedia articles, or questions and answers in StackOverflow. The quality of such documents, however, varies greatly, ranging from well-written documents to poorly written documents with little if any content.

Automatic quality assessment is needed for the following reasons. First, the number of digital documents being generated is huge and continues to grow. It is infeasible to assess the quality of all documents manually and in a timely manner. Second, even given the same grading scheme, people assess quality of documents in a subjective way, which makes it difficult to reach consensus among raters and assign a proper quality class to a document. An automatic labeling approach, however, can enable immediate feedback to both contributors and readers, ideally with a justification for why a given label has been assigned. A high quality class assignment can give users greater trust in the document, while a low-quality class assignment can direct the efforts of contributors to improve certain articles.

In the absence of a general-purpose document quality assessment dataset, we use and expand on a document quality dataset sourced from English Wikipedia. The quality of Wikipedia articles is inconsistent, for reasons including: (1) not all contributors (i.e., users who edit a Wikipedia article) are experts in the area of the articles they edit, and different contributors have different writing styles; (2) some articles receive more attention than others, resulting in imbalances in the level of peer; (3) there is article vandalism (Wikipedia Vandalism) that lowers quality of Wikipedia articles.

Officially, there are six quality classes of Wikipedia articles, which are (in descending order of quality): *Featured Article* (FA), *Good Article* (GA), *B-class Article* (B), *C-class Article* (C), *Start Article* (Start), and *Stub Article* (Stub). The quality class of an article is determined by Wikipedia reviewers, and any registered user can become a reviewer. A general description of the criteria of different quality classes can be found in the Wikipedia grading scheme page.[1] The difference between different quality classes is subjective and ambiguous, especially for adjacent classes. This presents significant challenges in assigning quality classes consistently. Furthermore, there maybe some qualitative differences between different datasets. If there are more articles whose quality is at the boundary of adjacent classes, these articles are more likely to be misclassified into their adjacent classes. Lastly, the quality labels assigned to Wikipedia articles are not always trustworthy. For example, there are some noisy data

[1] https://en.wikipedia.org/wiki/Template:Grading_scheme

Aili Shen, Jianzhong Qi and Timothy Baldwin. 2017. A Hybrid Model for Quality Assessment of Wikipedia Articles. In *Proceedings of Australasian Language Technology Association Workshop*, pages 43–52.

points (e.g., an empty article is labeled as an FA article in the 30K dataset — see Section 5.1).

In this paper, we formulate the assessment of Wikipedia article quality as a classification problem and propose a hybrid model combining deep learning and hand-engineered features. We show that *long short-term memories* (LSTMs) (Hochreiter and Schmidhuber, 1997) generate effective document representations for article quality classification. We also show that there is benefit to supplementing the document embedding with hand-engineered features, to better capture the subtleties of the task.

This paper makes the following contributions:

(i) We formulate the quality assessment of Wikipedia articles as a classification problem, and propose a novel approach which combines LSTMs with hand-engineered features.

(ii) We construct a large-scale Wikipedia article dataset with quality class labels, by combining an existing dataset with newly-crawled data; this will be released for public use on acceptance of this paper.[2]

(iii) We report empirical results of the proposed model on three datasets, and show that the proposed model achieves state-of-the-art results in quality classification accuracy.

2 Related Work

A number of approaches for quality assessment of Wikipedia articles have been proposed, which can be classified into three categories: (i) meta-data based approaches; (ii) article internal feature-based approaches; and (iii) meta-data and article internal feature-based approaches. In addition to quality assessment of Wikipedia articles, there is some work measuring essays written by (second) language learners and content quality in community question answering (cQA). As our main focus is on assessing quality of Wikipedia articles, this will be the focus of the literature review, with a brief mention of work on automatic essay scoring and cQA content quality assessment at the end of the section.

Meta-data based approaches use the meta-data of Wikipedia articles, e.g., contributors of the articles, to perform quality classification. For example, Stein and Hess (2007) and Adler et al. (2008) use the authority of the contributors to measure the quality of Wikipedia articles. Suzuki (2015) uses h-index and p-ratio to measure editor quality and uses editor quality to assess the quality of Wikipedia articles. Li et al. (2015) use article–editor networks and PageRank to assess Wikipedia article quality. These approaches require collecting large amounts of meta-data about the articles such as their edit history, and article quality prediction is indirect, i.e., based on external evidence such as article contributors instead of the article content itself.

Article internal features refer to features derived from the articles themselves. Various such features have been used for Wikipedia article quality assessment. Blumenstock (2008) uses article length as a metric to assess the quality of Wikipedia articles. A high accuracy is achieved in separating featured articles from non-featured articles despite the simplicity of this approach. Lipka and Stein (2010) use writing style, which is represented by exploiting binarized character trigram features, to identify featured articles. Warncke-Wang et al. (2013) propose a model to assess article quality, which includes five features extracted from article text: completeness, informativeness, number of headings, and ratio of number of references to article length. Later, Warncke-Wang et al. (2015) propose a model including 11 structural features (such as number of references) and use a random forest ("RF") to classify Wikipedia articles by quality. Dang and Ignat (2016a) further add nine readability metrics (such as Flesch reading-ease score) to the structural features, and use a RF to classify Wikipedia articles based on their quality. Based on these last two studies, an online Objective Revision Evaluation Service (ORES) has been built to measure the quality of Wikipedia articles (Halfaker and Taraborelli, 2015). ORES requires the revision ID of a Wikipedia article as its input parameter. Our experimental datasets do not contain such information. Thus, we compare with the model proposed by Dang and Ignat (2016a) instead of ORES in the experiments.

Recently, Dang and Ignat (2016b) use a distributed memory version of *Paragraph Vector* (Le and Mikolov, 2014) to learn document embeddings, which they fed into a four-layer neural network to classify articles for quality. This study does not consider the order of sentences, which

[2] https://bitbucket.org/unimelb_nlp/wiki60k

may affect article quality: if the sentences in an article are not ordered in a logical way, it is more difficult for reviewers and readers to understand the article, and the quality will be lower. Our approach differs from that of Dang and Ignat (2016b) in that we use an LSTM, which captures the order between sentences, to learn a high-level representation of Wikipedia articles. Furthermore, although neural networks can learn features from the article content, they require a large training dataset. Due to the limited availability of labeled Wikipedia articles, we supplement the document embedding with hand-engineered features, which can lead to better quality prediction even with a relatively small volume of training data. Cheng et al. (2016) adopt a similar idea in an app recommendation scenario.

There are also hybrid approaches that use both meta-data and article internal features for quality assessment. Stvilia et al. (2005) present seven Information Quality metrics based on article features and edit history of 834 articles. Dalip et al. (2009) analyze the effect of different feature sets on Wikipedia article quality assessment. The feature sets considered include article text, revision history, and citation network (where nodes are articles and edges are citations between them). A regression model is proposed for quality class prediction. Dalip et al. (2009) find that textual features extracted from articles are the best indicators to distinguish articles of different quality classes.

For the related task of automatic essay scoring, the following dimensions are often captured: topic relevance, organization and coherence, word usage and sentence complexity, and grammar and mechanics. To measure whether an essay is relevant to its "prompt" (i.e., the description of the essay topic), lexical overlap and semantic overlap between an essay and its corresponding prompt can be used (Phandi et al., 2015; Persing and Ng, 2014). Lexical overlap and semantic similarity features are exploited to measure coherence between different discourse elements, sentences, and paragraphs (Higgins et al., 2004; McNamara et al., 2015). Attali and Burstein (2004) explore word features, such as the number of verb formation errors, average word frequency, and average word length, to measure word usage and lexical complexity. Intelli-Metric (Rudner et al., 2006) uses sentence structure features, such as syntactic variety and readability, to measure sentence variety and complexity. The effects of grammatical and mechanics errors on the quality of an essay are measured via word and POS n-gram features and "mechanics" features (e.g., spelling, capitalization, and punctuation), respectively (Persing and Ng, 2013; Higgins et al., 2004).

To measure content quality in cQA, researchers exploit various features from different sources, such as the content itself, the user's profile, asking and answering interaction among users, and usage of the content. The most common feature used is the content length (Jeon et al., 2006; Suryanto et al., 2009). Agichtein et al. (2008) explore syntactic and semantic complexity features, such as the entropy of word lengths and various readability scores. Le et al. (2016) exploit user's characteristic features (e.g., the grade level or the rank of the user in cQA) and the user's historical features (e.g., the number of questions asked by the user, and the number of answers given by the user). Suryanto et al. (2009) and Jurczyk and Agichtein (2007) exploit asking and answering expertise features, which can be computed through the HITS algorithm (Kleinberg, 1999). Asking expertise (hub values) of a user is derived from the answering expertise of other users answering questions posted by this user. A user's answering expertise (authority score) is derived from the asking expertise of other users posting questions answered. Usage features, such as the number of clicks (views), are also beneficial in measuring content quality in cQA (Burel et al., 2012).

3 Problem Definition

We formulate quality assessment of Wikipedia articles as a multi-class classification problem.

The input of the problem is a set of Wikipedia articles denoted by $\mathbb{D}$. Each article is denoted as a tuple $\langle a, c \rangle$, where a represents the article content, and c is a latent true quality class of the article. The value of c belongs to a set $\mathbb{C}$ of quality classes: $\mathbb{C} = \{\mathsf{FA}, \mathsf{GA}, \mathsf{B}, \mathsf{C}, \mathsf{Start}, \mathsf{Stub}\}$.

We aim to predict a quality class $\hat{c}$ for each article, such that $\hat{c}$ is as close as possible to the true latent quality class c of the article. Our classification model to achieve this purpose is essentially a mapping function: $f : \mathbb{D} \rightarrow \mathbb{C}$. Here, the optimization goal of the mapping function is to minimize the difference between the predicted quality class $\hat{c}$ and the true latent quality class c of an article.

4 The Proposed Hybrid Model

The proposed classification model is a hybrid model that integrates neural network document embeddings and hand-engineered features. In this section, we first describe the LSTM-based model to document embeddings of Wikipedia articles, then we present the hand-engineered features, and finally we describe how we combine the two.

4.1 Document Embedding Learning

We adopt a bidirectional LSTM model to generate document embeddings of Wikipedia articles; we will refer to this model as Bi-LSTM. The input of Bi-LSTM is the text of an article, and the output is a document embedding, which we later integrate with hand-engineered features.

We explain our model in detail as illustrated in Fig. 1. First, an average-pooling layer is applied to word embeddings within a sentence to obtain a sentence embedding. Each word is represented as a word embedding (Bengio et al., 2003), which is a continuous, real-valued vector.

Second, we use a bidirectional LSTM to generate a document embedding over the sentence embeddings. Suppose that an article contains n sentences: $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n$, the bidirectional LSTM contains a forward LSTM which reads an article from sentence $\boldsymbol{s}_1$ to $\boldsymbol{s}_n$ and a backward LSTM which reads an article from sentence $\boldsymbol{s}_n$ to $\boldsymbol{s}_1$. Given a sentence $\boldsymbol{s}_j$, we can obtain its hidden state $\boldsymbol{h}_j$ of $\boldsymbol{s}_j$ by concatenating the forward output $\overrightarrow{\boldsymbol{h}}_j$ and the backward output $\overleftarrow{\boldsymbol{h}}_j$, i.e., $\boldsymbol{h}_j = [\overrightarrow{\boldsymbol{h}}_j, \overleftarrow{\boldsymbol{h}}_j]$.

Last, a max-pooling layer is applied to select the most salient features among the component sentences. Then the output of the max-pooling layer is fed into a feedforward neural network with ReLU as the activation function, which produces our neural network learned high-level representation $\boldsymbol{f}_l$.

4.2 Hand-Engineered Features

Following Dang and Ignat (2016a), we use structural features and readability scores as the hand-engineered features for quality class prediction. The structural features can capture the structure information of articles and the readability scores can reflect writing styles. These features are listed in Table 1.

The structural features reflect article quality in different ways. For example, *article length* captures how much content an article contains (with the expectation that articles that do not contain much content are usually of low quality). The *number of references*, *number of links to other Wikipedia pages*, and *number of citation templates* show how the article editors support their content by using information from different sources, which makes the article more reliable and of higher quality. The *number of level 2 and level 3+ headings* reflect how the content is organized. Usually, Wikipedia articles of high quality have appropriate *number of level 2 and level 3+ headings*.

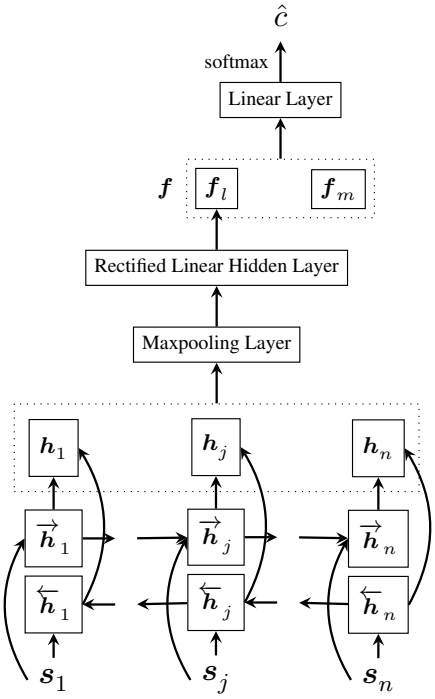

Figure 1: Overview of the proposed model.

Readability scores reflect the use of language and how easy to read an article is. *Flesch reading score*, *Flesch-Kincaid grade level*, *Smog index*, and *Linsear write formula* use the average syllable per word or the number of polysyllables with different weight values to measure how difficult a text is to understand. Both *Coleman-Liau index* and *Automated readability index* use the average word length with different weight values to measure the readability of texts. *Difficult words*, *Dale-Chall score*, and *Gunning-Fog index* use the number of difficult words or percentage of difficult words to measure the comprehension difficulty of a text. Here, a word is considered difficult if it is not in a list of 3000 common English words that fourth-grade American students can reliably un-

Structural Features	Readability Scores
Article length in bytes	Flesch reading score (Kincaid et al., 1975)
Number of references	Flesch-Kincaid grade level (Kincaid et al., 1975)
Number of links to other Wikipedia pages	Smog index (Mc Laughlin, 1969)
Number of citation templates	Coleman-Liau index (Coleman and Liau, 1975)
Number of non-citation templates	Automated readability index (Senter and Smith, 1967)
Number of categories linked in the text	Difficult words (Chall and Dale, 1995)
Number of images / length of the article	Dale-Chall score (Dale and Chall, 1948)
Information noise score (Zhu and Gauch, 2000)	Linsear write formula (Chen, 2012)
Article having an infobox or not	Gunning-Fog index (Gunning, 1969)
Number of level 2 headings	
Number of level 3+ headings	

Table 1: Hand-engineered features.

derstand.

Hand-engineered features are extracted using the open-source packages *wikiclass*[3] and *textstat*:[4] *wikiclass* is used to extract structural features and *textstat* is used to compute readability scores.

4.3 The Proposed Hybrid Model Bi-LSTM$^+$

The proposed hybrid model, denoted as Bi-LSTM$^+$, concatenates the neural network learned high-level representation $\boldsymbol{f}_l$ with hand-engineered features $\boldsymbol{f}_m$, which results in the combined feature vector $\boldsymbol{f}$. The combined features $\boldsymbol{f}$ are used to classify Wikipedia articles according to their quality. This is done by feeding the combined feature vector $\boldsymbol{f}$ into a linear layer and softmax layer to predict the probability distribution over quality classes. We use the cross-entropy between ground truth distribution and predicted distribution as the loss function to train our model.

5 Experiments

We report the results of an empirical study on the proposed hybrid model in this section.

5.1 Datasets

In our experiments, we use three different datasets with different numbers of Wikipedia articles: 20K dataset,[5] 30K dataset,[6] and 60K dataset. The Wikipedia articles in these datasets contain manually labeled quality classes, which are used as the ground truth in our experiments. The 20K dataset is provided by Warncke-Wang et al. (2015). The 30K dataset is provided by the Wikimedia Foundation. The 60K dataset is obtained by combining the 30K dataset with newly crawled articles of different quality classes. We wrote a Python script to crawl articles from each quality class, and eliminate talk pages from the crawled data, resulting in about 5K articles from each quality class. We crawl about 5K articles from each quality class because there are only about 5K FA articles from the featured article category and we want the dataset to be evenly distributed (among the labeled Wikipedia articles, 71% is labeled as Stub and 0.096% is labeled as FA). Table 2 summarizes the quality class distributions of the three datasets.

	20K	30K	60K
FA	2414	4920	9908
GA	3160	4891	9898
B	3201	4913	9913
C	3318	4907	9907
Start	4096	4910	9910
Stub	4243	4915	9915
Total	20432	29456	59451

Table 2: Datasets used in our experiments.

5.2 Experimental Setting

We divide a Wikipedia article into sentences and tokenize them using *NLTK* (Bird, 2006; Bird et al., 2010). Words appearing more than 20 times are retained in building the vocabulary. All low frequency words are replaced by the special `UNK` token. We use the pre-trained *GloVe* (Pennington et al., 2014) 50-dimensional word embeddings to

[3] https://github.com/wiki-ai/wikiclass

[4] https://pypi.python.org/pypi/textstat/0.3.1

[5] http://figshare.com/articles/English_Wikipedia_Quality_Assessment_Dataset/1375406

[6] https://datasets.wikimedia.org/public-datasets/enwiki/

represent words found in the GloVe dataset. For words that cannot be found in GloVe, word embeddings are randomly initialized based on sampling from a uniform distribution $U(-1, 1)$. All word embeddings are updated in the training process.

For evaluation, we perform 10-fold cross-validation over the three datasets, using 90% as training data (10% of which is in turn used as the development set for early stopping), and the rest test data. We report the *average classification accuracy* (combined across the cross validation folds).

Hyper-parameters of the proposed model are tuned on the development set for a given iteration of cross validation. We set the word embedding dimension to 50 and the hidden size of Bi-LSTM$^+$ to 256. Then the concatenation of the forward and backward LSTMs gives us 512 dimensions for the document embedding. The feedforward neural network produces the output $\boldsymbol{f}_l$, which is a real-valued vector with 40 dimensions. Concatenating with hand-engineered features $\boldsymbol{f}_m$, which is a real-valued normalized vector with 20 dimensions, we obtain the combined features of $\boldsymbol{f}$ with 60 dimensions for each article, which are used as the features for Wikipedia article quality classification. A linear layer and softmax layer are applied on the combined features $\boldsymbol{f}$, which produces the predicted distribution $\hat{\boldsymbol{c}}$. To save training time, articles with more than 350 sentences are clipped and only the first 350 sentences are used. During training, we use a mini-batch size of 128. We use the Adam optimizer (Kingma and Ba, 2014) to train the model with a learning rate of 0.001. Dropout layers are applied to the input of Bi-LSTM$^+$ and the neural network learned high-level representation $\boldsymbol{f}_l$ with a dropout probability of 0.5.

5.3 Experimental Results

We compare the proposed model Bi-LSTM$^+$ with two state-of-the-art approaches RF (Dang and Ignat, 2016a) and Doc2Vec (Dang and Ignat, 2016b). RF only uses the structural features and readability scores as features to build a random forest. Doc2Vec uses Paragraph Vectors to learn document embeddings, and builds a classification model on top of this. The hyper-parameters of RF and Doc2Vec are set as described in the corresponding papers. We also compare with a model using only Bi-LSTM learned document embeddings, denoted as Bi-LSTM.

	20K	30K	60K
RF	**63.70%**	58.63%	61.71%
Doc2Vec	59.84%	54.98%	61.46%
Bi-LSTM	56.04%	54.36%	65.16%
Bi-LSTM$^+$	63.59%	**58.98%**	**68.17%**†

Table 3: Results.

Table 3 shows the experimental results. We see that on the 20K and 30K datasets, Bi-LSTM$^+$ and RF have very close performance: RF has a 0.11% higher accuracy on the 20K dataset while Bi-LSTM$^+$ has a 0.35% higher accuracy on the 30K dataset. Wilcoxon signed-rank test demonstrates that the performance difference of RF and Bi-LSTM$^+$ is not significant over the 20K and 30K datasets. However, on the larger 60K dataset, Bi-LSTM$^+$ gains a 6.5% higher accuracy than that of RF. The performance gain of Bi-LSTM$^+$ is statistically significant ($p < 0.01$) on the 60K dataset, which is emphasized using a † symbol. Doc2Vec and Bi-LSTM have a lower accuracy than that of Bi-LSTM$^+$ on all three datasets.

6 Analysis and Discussion

In this section, the performance of the hybrid model Bi-LSTM$^+$ is analyzed, and we discuss the task of of quality assessment of Wikipedia articles.

6.1 Analysis

Impact of hand-engineered features on dataset of different sizes. Bi-LSTM$^+$ and RF have very close performance on the 20K and 30K datasets, which shows the effectiveness of hand-engineered features in article quality classification over smaller datasets. Meanwhile, the better performance of Bi-LSTM$^+$ on the 60K dataset highlights the advantage of a neural network based model when there is more training data. Further, by comparing Bi-LSTM with Bi-LSTM$^+$, we find that the improvement gained by adding hand-engineered features decreases as the dataset size gets larger: the hand-engineered features produce an accuracy improvement of 7.55% on the 20K dataset, 4.62% on the 30K dataset, and 3.01% on the 60K dataset. This suggests that as the dataset size increases, the neural network can learn more robust features directly from the document content, and hence the performance improvement of Bi-LSTM$^+$ from the hand-engineered features decreases.

Quality	FA	GA	B	C	Start	Stub	Class Total	Accuracy
FA	880	55	44	6	0	6	991	88.80%
GA	111	701	137	34	7	0	990	70.81%
B	45	67	619	197	48	16	992	62.40%
C	13	50	205	598	96	29	991	60.34%
Start	4	1	71	191	513	211	991	51.77%
Stub	1	0	9	24	144	814	992	82.06%
Predicted Total	1054	874	1085	1050	808	1076	5947	69.36%

Table 4: Confusion matrix of Bi-LSTM⁺ on a test set of the 60K dataset. The last column is the accuracy for each class. Diagonal elements (gray cells) of the matrix are correct predictions. Rows are actual quality classes, and columns are the predicted quality classes.

Different classification accuracy of different quality classes. To further analyze the performance of our model Bi-LSTM⁺, we dive into the classification results of each quality class. We show the confusion matrix of the experiment on one fold test set of the 60K dataset in Table 4. A total of 5947 articles are used for testing in this experiment. In the matrix, the diagonal elements show the number of correct predictions for each class. For example, 880 FA articles have been predicted as FA correctly. Each row in the matrix shows the prediction result for the articles of a certain class. For example, the first row shows that among the 991 FA articles, 880, 55, 44, 6, 0, and 6 articles are predicted to be FA, GA, B, C, Start, and Stub, respectively. Each column shows the numbers of articles of different classes that have been predicted to be a certain class. For example, the first column shows that there are 880, 111, 45, 13, 4, and 1 articles of different classes predicted to be FA.

It is more difficult to classify articles at adjacent quality classes. In Table 4, for example, there are 191 and 211 Start articles that have been misclassified as C and Stub articles, respectively, which are adjacent classes of Start articles. The low accuracy of B, C, and Start articles demonstrates that they are more difficult to classify correctly. The proposed model Bi-LSTM⁺ achieves a higher accuracy on both FA and GA, which can be explained by that both FA and GA pass an official review and that the difference between them is clearer. In fact, it is difficult to classify articles in adjacent classes even for a human reader, e.g., B article *Wave Hill Station*[7] and C article *Ivanhoe*,[8] are difficult for humans to assign the correct label to. If we allow an article to be classified into adjacent classes (e.g., it is regarded as a correct classification if a GA article is classified into its adjacent classes FA or B.), then the overall accuracy of the proposed model Bi-LSTM⁺ will increase to 93.14% from 69.36%.[9]

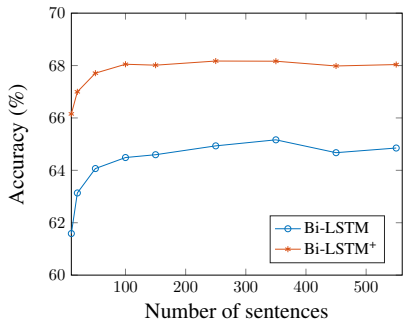

Figure 2: Performance of Bi-LSTM and Bi-LSTM⁺ on the 60K dataset varying number of sentences.

Impact of hand-engineered features on models using different number of sentences in each article. To further justify the use of hand-engineered features, we vary the number of sentences fed into Bi-LSTM⁺ and Bi-LSTM, and compare the accuracy of the two models on the 60K dataset. We feed from the first 10 to the first 550 sentences of each article into Bi-LSTM and Bi-LSTM⁺, respectively. From Fig. 2, we can see that the performance gain of Bi-LSTM⁺ over Bi-LSTM is most significant (up to 4.57%) when the number of sentences is below 100. As the number of sentences continues to increase, the performance gain becomes smaller. This suggests that when the number of sentences per article is small, there are less features that can be learned by the neural network. The performance of Bi-LSTM⁺

[7] https://en.wikipedia.org/w/index.php?title=Wave_Hill_Station&oldid=767773441

[8] https://en.wikipedia.org/w/index.php?title=Ivanhoe&oldid=802355224

[9] This number is higher than 68.17% shown in Table 3 because this is the result of one fold of 10-fold cross-validation.

can be better compensated by the hand-engineered features. As we train on more sentences, more features are learned by the neural network, and the contributions of hand-engineered features become less. We also see that, beyond 350 sentences, the performance of Bi-LSTM$^+$ does not increase any more. This is because the neural network may forget the afore-learned features, which are more important, if there is too much content to learn from. Thus, we have used 350 sentences per article as our default setting in the experiments.

6.2 Discussion

We also ran an experiment where we formulated the quality assessment task as a regression problem. The dependent variable is the quality class, and the independent variables are the combined features $\boldsymbol{f}$. We convert the quality class to integers: Stub to 0, Start to 1, C to 2, B to 3, GA to 4, and FA to 5. After we get the predicted quality value of the test data using the regression model, we convert back to the quality class by rounding (and truncating at either end of the scale).

When we regard the quality assessment of Wikipedia articles as a regression problem, the classification accuracy is poor, and we thus do not report the results here. This may be because it is difficult to learn when the quality assessment of Wikipedia articles is regarded as a regression problem. In the future, we will formulate the quality assessment of Wikipedia articles as an ordinal regression problem where only the relative ordering between different quality classes is important.

It is inappropriate to compare performance across different datasets, even with ones of the same size. We perform an experiment on the newly crawled 30K dataset. The accuracy of RF, Doc2Vec, Bi-LSTM, and Bi-LSTM$^+$ is 65.53%, 67.46%, 75.34%, and 76.33%, respectively. The accuracy of all approaches on the newly crawled 30K dataset is higher (ranging from 6.9% to 20.98%) than that on the 30K dataset provided by Wikimedia Foundations. One reason for the performance difference is that there are noisy data points in the 30K dataset provided by Wikimedia Foundations, even after removing articles with obvious problems (e.g., an empty article is labeled as an FA article). Another reason is that there may be some qualitative differences between the 30K dataset provided by Wikimedia Foundations and our newly crawled 30K dataset. For example, there may be more articles whose quality is at the boundary of adjacent classes in the 30K dataset provided by the Wikimedia Foundation than those in our newly crawled 30K dataset and more articles being misclassified into their adjacent classes, which results in poor performance of all approaches over the 30K dataset provided by the Wikimedia Foundation.

Returning to our original objective of general-purpose document quality assessment, the most commonly used quality factors across different domains include: grammaticality, readability, stylistics, structure, correctness, and technical depth. These quality factors, however, have different impact on document quality across different domains. For example, people emphasize grammaticality more in Wikipedia articles and essays written by (second) language learners than in the case of cQA posts. Features used in automated essay scoring and quality assessment in cQA, which are applicable to assessing the quality of Wikipedia articles, will be exploited in the future. We also expressed misgivings about the quality of the labels in the Wikipedia dataset. We did not perform inter-annotator agreement analysis since each article only has a single quality class label assigned by the Wikipedia community. In the future, we want to validate our proposed approach on datasets with quality ratings from different annotators on each of the aforementioned six dimensions, for a more robust evaluation.

7 Conclusions

We propose a hybrid model to classify Wikipedia articles based on their quality, which integrates Bi-LSTM learned document embeddings with hand-engineered features for article quality classification. As part of this, we construct a novel dataset. Experimental results show that the proposed model achieves a 6.5% higher accuracy than state-of-the-art approaches over a set of 60K Wikipedia articles. The results also show that hand-engineered features play an important role in obtaining the correct classification, which justifies the use of such features, especially when the size of the training data is limited. Further, the quality of documents should be assessed from different dimensions and different annotators should be employed to alleviate subjectivity of assessing document quality.

References

B. Thomas Adler, Krishnendu Chatterjee, Luca De Alfaro, Marco Faella, Ian Pye, and Vishwanath Raman. 2008. Assigning trust to wikipedia content. In *Proceedings of the 4th International Symposium on Wikis*. pages 26:1–26:14.

Eugene Agichtein, Carlos Castillo, Debora Donato, Aristides Gionis, and Gilad Mishne. 2008. Finding high-quality content in social media. In *Proceedings of the International Conference on Web Search and Web Data Mining*. pages 183–194.

Yigal Attali and Jill Burstein. 2004. Automated essay scoring with e-rater® v. 2.0. *ETS Research Report Series* 2004(2).

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of Machine Learning Research* 3(Feb):1137–1155.

Steven Bird. 2006. NLTK: the natural language toolkit. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*. pages 69–72.

Steven Bird, Ewan Klein, and Edward Loper. 2010. Natural language processing with python: analyzing text with the natural language toolkit. *Language Resources and Evaluation* 44(4):421–424.

Joshua E. Blumenstock. 2008. Size matters: word count as a measure of quality on wikipedia. In *Proceedings of the 17th International Conference on World Wide Web*. pages 1095–1096.

Grégoire Burel, Yulan He, and Harith Alani. 2012. Automatic identification of best answers in online enquiry communities. In *The Semantic Web: Research and Applications - 9th Extended Semantic Web Conference*. pages 514–529.

Jeanne Sternlicht Chall and Edgar Dale. 1995. *Readability revisited: The new Dale-Chall readability formula*. Brookline Books.

Hai-Hon Chen. 2012. How to use readability formulas to access and select english reading materials. *Journal of Educational Media & Library Sciences* 50(2):229–254.

Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*. pages 7–10.

Meri Coleman and T. L. Liau. 1975. A computer readability formula designed for machine scoring. *Journal of Applied Psychology* 60(2):283.

Edgar Dale and Jeanne S. Chall. 1948. A formula for predicting readability: Instructions. *Educational Research Bulletin* pages 37–54.

Daniel H. Dalip, Marcos A. Gonçalves, Marco Cristo, and Pável Calado. 2009. 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*. pages 295–304.

Quang-Vinh Dang and Claudia-Lavinia Ignat. 2016a. Measuring quality of collaboratively edited documents: the case of wikipedia. In *The 2nd IEEE International Conference on Collaboration and Internet Computing*. pages 266–275.

Quang-Vinh Dang and Claudia-Lavinia Ignat. 2016b. Quality assessment of wikipedia articles without feature engineering. In *Proceedings of the 16th ACM/IEEE-CS Joint Conference on Digital Libraries*. pages 27–30.

Robert Gunning. 1969. The fog index after twenty years. *International Journal of Business Communication* 6(2):3–13.

Aaron Halfaker and Dario Taraborelli. 2015. Artificial intelligence service ores gives wikipedians x-ray specs to see through bad edits. *[online] Available: https://blog.wikimedia.org/2015/11/30/artificial-intelligence-x-ray-specs* .

Derrick Higgins, Jill Burstein, Daniel Marcu, and Claudia Gentile. 2004. Evaluating multiple aspects of coherence in student essays. In *Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*. pages 185–192.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9(8):1735–1780.

Jiwoon Jeon, W. Bruce Croft, Joon Ho Lee, and Soyeon Park. 2006. A framework to predict the quality of answers with non-textual features. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. pages 228–235.

Pawel Jurczyk and Eugene Agichtein. 2007. Discovering authorities in question answer communities by using link analysis. In *Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management*. pages 919–922.

J. Peter Kincaid, Robert P. Fishburne Jr., Richard L. Rogers, and Brad S. Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. Technical report, DTIC Document.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR* abs/1412.6980. http://arxiv.org/abs/1412.6980.

Jon M. Kleinberg. 1999. Authoritative sources in a hyperlinked environment. *J. ACM* 46(5):604–632.

Long T. Le, Chirag Shah, and Erik Choi. 2016. Evaluating the quality of educational answers in community question-answering. In *Proceedings of the 16th ACM/IEEE-CS on Joint Conference on Digital Libraries*. pages 129–138.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning*. volume 14, pages 1188–1196.

Xinyi Li, Jintao Tang, Ting Wang, Zhunchen Luo, and Maarten De Rijke. 2015. Automatically assessing wikipedia article quality by exploiting article–editor networks. In *37th European Conference on Information Retrieval*. pages 574–580.

Nedim Lipka and Benno Stein. 2010. Identifying featured articles in wikipedia: writing style matters. In *Proceedings of the 19th International Conference on World Wide Web*. pages 1147–1148.

G. Harry Mc Laughlin. 1969. Smog grading-a new readability formula. *Journal of Reading* 12(8):639–646.

Danielle S McNamara, Scott A Crossley, Rod D Roscoe, Laura K Allen, and Jianmin Dai. 2015. A hierarchical classification approach to automated essay scoring. *Assessing Writing* 23:35–59.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*. volume 14, pages 1532–1543.

Isaac Persing and Vincent Ng. 2013. Modeling thesis clarity in student essays. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*. pages 260–269.

Isaac Persing and Vincent Ng. 2014. Modeling prompt adherence in student essays. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*. pages 1534–1543.

Peter Phandi, Kian Ming Adam Chai, and Hwee Tou Ng. 2015. Flexible domain adaptation for automated essay scoring using correlated linear regression. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. pages 431–439.

Lawrence M Rudner, Veronica Garcia, and Catherine Welch. 2006. An evaluation of intellimetric essay scoring system. *The Journal of Technology, Learning and Assessment* 4(4).

R. J. Senter and E. A. Smith. 1967. Automated readability index. Technical report, DTIC Document.

Klaus Stein and Claudia Hess. 2007. Does it matter who contributes: a study on featured articles in the german wikipedia. In *Proceedings of the 18th Conference on Hypertext and Hypermedia*. pages 171–174.

Besiki Stvilia, Michael B. Twidale, Linda C. Smith, and Les Gasser. 2005. Assessing information quality of a community-based encyclopedia. In *Proceedings of the 2005 International Conference on Information Quality*. pages 442–454.

Maggy Anastasia Suryanto, Ee-Peng Lim, Aixin Sun, and Roger H. L. Chiang. 2009. Quality-aware collaborative question answering: methods and evaluation. In *Proceedings of the Second International Conference on Web Search and Web Data Mining*. pages 142–151.

Yu Suzuki. 2015. Quality assessment of wikipedia articles using h-index. *Journal of Information Processing* 23(1):22–30.

Morten Warncke-Wang, Vladislav R. Ayukaev, Brent Hecht, and Loren Terveen. 2015. The success and failure of quality improvement projects in peer production communities. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*. pages 743–756.

Morten Warncke-Wang, Dan Cosley, and John Riedl. 2013. Tell me more: an actionable quality model for wikipedia. In *Proceedings of the 9th International Symposium on Open Collaboration*. pages 8:1–8:10.

Wikipedia Vandalism. Accessed Jun., 2017. Wikipedia vandalism. https://en.wikipedia.org/wiki/Wikipedia:Vandalism.

Xiaolan Zhu and Susan Gauch. 2000. Incorporating quality metrics in centralized/distributed information retrieval on the world wide web. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. pages 288–295.