A Joint Model for Multimodal Document Quality Assessment

Aili Shen, Bahar Salehi, Timothy Baldwin, Jianzhong Qi
School of Computing and Information Systems
The University of Melbourne
ailis@student.unimelb.edu.au,{salehi.b,tbaldwin,jianzhong.qi}@unimelb.edu.au

ABSTRACT

The quality of a document is affected by various factors, including grammaticality, readability, stylistics, and expertise depth, making the task of document quality assessment a complex one. In this paper, we explore this task in the context of assessing the quality of Wikipedia articles. Observing that the visual rendering of a document can capture implicit quality indicators that are not present in the document text — such as images, font choices, and visual layout — we propose a joint model that combines the text content with a visual rendering of the document for document quality assessment. The experimental result over a Wikipedia dataset reveals that textual and visual features are complementary, achieving state-of-the-art results. Further experiments on an Peer Review dataset verify the general applicability of our proposed model.

CCS CONCEPTS

• **Information systems → Digital libraries and archives**.

KEYWORDS

Document quality assessment, Multimodal, Visual embeddings, Textual embeddings, Wikipedia articles

ACM Reference Format:
Aili Shen, Bahar Salehi, Timothy Baldwin, Jianzhong Qi. 2019. A Joint Model for Multimodal Document Quality Assessment. In *Proceedings of (JCDL'19).* , 4 pages.

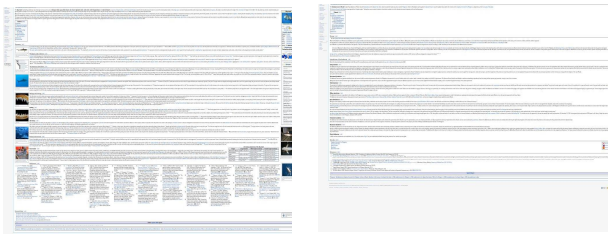
1 INTRODUCTION

Quality assessment of Wikipedia articles is a task that assigns a quality class label to a given Wikipedia article, mirroring the quality assessment process that the Wikipedia community carries out manually. Automatic quality assessment has obvious benefits in terms of time savings and tractability in contexts where the volume of documents is large. In the case of dynamic documents (possibly with multiple authors), such as with Wikipedia, it is particularly pertinent, as any edit potentially has implications for the quality label of that document (and around 1.8 English Wikipedia documents are edited per second[1]). Furthermore, when the quality assessment task is decentralized (as in the case of Wikipedia), quality criteria are often applied inconsistently by different people, where an automatic document quality assessment system could potentially reduce inconsistencies and enable immediate author feedback.

Current studies on document quality assessment mainly focus on textual features. For example, Warncke-Wang et al. [22] examine features such as the article length and the number of headings

[1]https://en.wikipedia.org/wiki/Wikipedia:Statistics

JCDL'19, June 2-6, 2019, Urbana-Champaign, Illinois, USA
2019.



Featured article | Lower quality article

Figure 1: Visual renderings of two example Wikipedia documents with different quality labels (not intended to be readable).

to predict the quality class of a Wikipedia article. In contrast to these studies, in this paper, we propose to combine textual features with visual features, based on a visual rendering of the document. Figure 1 illustrates our intuition, relative to Wikipedia articles. Without being able to read the text, we can tell that the article in the left has higher quality than the one in the right, as it has a detailed infobox, extensive references, and a variety of images. Based on this intuition, we aim to answer the following question: *Can we achieve better accuracy on document quality assessment by complementing textual features with visual features?*

Our visual model is based on fine-tuning an Inception V3 model [20] over visual renderings of documents, while our textual model is based on a bidirectional LSTM [9]. We further combine the two into a joint model. We perform experiments on a Wikipedia dataset. Experimental results on the visual renderings of documents show that implicit quality indicators, such as images and visual layout, can be captured by an image classifier, at a level comparable to a text classifier. When we combine the two models, we achieve state-of-the-art results over the Wikipedia dataset. Further experiments over an Peer Review dataset show that our proposed model is applicable to not only Wikipedia articles but also other types of documents.

This paper makes the following contributions:

(i) this is the first study to use combined text and visual renderings of documents to capture implicit quality indicators not present in the document text, such as document visual layout.
(ii) we further propose a joint model to predict document quality combining visual and textual features; we observe further improvements on the Wikipedia dataset indicating that visual and textual features are complementary.
(iii) we perform additional experiment over the Peer Review dataset and show the general applicability of our proposed model to assess the quality of documents.
(iv) we construct a large-scale Wikipedia dataset with full textual data, visual renderings, and quality class labels; we also supplement the existing Peer Review dataset with visual renderings of each document.

(v) All code and data associated with this research will be released on publication.

2 RELATED WORK

A variety of approaches has been proposed for assessing the quality of Wikipedia articles. Among these approaches, some use hand-crafted features while others use neural networks to automatically learn features from documents.

Many approaches have been proposed that use features from the article itself, meta-data features (e.g., the editors and Wikipedia article revision history), or a combination of the two. Article-internal features capture information such as whether an article is properly organized, with supporting evidence, and with appropriate terminology. For example, Lipka and Stein [14] use writing styles represented by binarized character trigrams to identify featured articles. Warncke-Wang et al. [22, 23] explore the number of headings, images, and references. Dang and Ignat [6] use nine readability scores, such as the percentage of difficult words, to measure article quality. Meta-data features, which are indirect indicators of article quality, are usually extracted from revision history and the interaction between editors and articles, an example of which is that higher-quality articles have more edits [3, 4]. Wang and Iwaihara [21] use the percentage of registered editors and the total number of editors of an article. Article–editor dependencies have also been explored. For example, Stein and Hess [19] use the authority of editors to measure the quality of Wikipedia articles, where the authority of editors is determined by the articles they edit.

Deep learning approaches to predicting Wikipedia article quality have also been proposed. For example, Dang and Ignat [7] use *doc2vec* [13] to represent articles, and feed the document embeddings into a four hidden layer neural network. Shen et al. [17] first obtain sentence representations by averaging words within a sentence, and then apply a biLSTM [9] to learn a document-level representation, which is combined with hand-crafted features as side information. Dang and Ignat [5] exploit two stacked biLSTMs to learn document representations.

As our main focus is to assess the quality of Wikipedia articles based on the article itself, we do not explore meta information (such as the revision history) of the articles.

3 THE PROPOSED JOINT MODEL

Following previous studies [5, 17, 22], we treat document quality assessment as a classification problem, i.e., given a document, we predict its quality class (e.g., which quality class should be assigned to an unseen Wikipedia article). In this section, we first present the details of the visual and textual embeddings, then describe how we combine the two.

3.1 Visual Embedding Learning

A wide range of models have been proposed to tackle the image classification task, such as *VGG* [18], *ResNet* [8], *Inception V3* [20], and *Xception* [2]. The only work we are aware of that has used visual renderings of documents to assess document quality is the very recent arXiv paper of Huang [10], which uses visual features only (similar to our Inception baseline in Section 4.3), to predict whether a paper is a conference or workshop paper. In this paper, we use Inception V3 pretrained on ImageNet[2] ("Inception" hereafter) to obtain visual embeddings of documents, noting that any image classifier could be applied to our task. The input to Inception is a visual rendering (screenshot) of a document, and the output is a visual embedding, which we will later integrate with our textual embedding.

3.2 Textual Embedding Learning

We adopt a bi-directional LSTM model to generate textual embeddings for document quality assessment, following the method of Shen et al. [17] ("biLSTM" hereafter).[3] The input to biLSTM is word embeddings of a textual document, and the output is a textual embedding, which will later integrate with the visual embedding.

3.3 The Joint Model

The proposed joint model ("Joint" hereafter) combines the visual and textual embeddings (output of Inception and biLSTM) via a simple feed-forward layer and softmax over the document label set. We optimize our model based on cross-entropy loss.

4 EXPERIMENTS

In this section, we first describe the Wikipedia dataset used in our experiments. Then, we report the experimental details and results.

4.1 Wikipedia Dataset

The Wikipedia dataset consists of articles from English Wikipedia, with quality class labels assigned by the Wikipedia community. Wikipedia articles are labelled with one of six quality classes, 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"). A description of the criteria associated with the different classes can be found in the Wikipedia grading scheme page.[4] We randomly sampled 5,000 articles from each quality class and removed all redirect pages, resulting in a dataset of 29,794 articles. As the wikitext contained in each document may contain markup relating to the document category such as *{Featured Article}* or *{geo-stub}*, which reveals the label, we remove such information. We randomly partitioned this dataset into training, development, and test splits based on a ratio of 8:1:1.

We generate a visual representation of each document via a 1,000×2,000-pixel screenshot of the article via a PhantomJS script over the rendered version of the article,[5] ensuring that the screenshot and wikitext versions of the article are the same version. Any direct indicators of document quality (such as the FA indicator, which is a bronze star icon in the top right corner of the webpage) are removed from the screenshot.

4.2 Experimental Setting

As discussed above, our model has two main components — biLSTM and Inception— which generate textual and visual representations,

[2] http://www.image-net.org/
[3] We did try a hierarchical attention network [24], but it didn't give better results than a vanilla LSTM. We adopt a vanilla LSTM as one of our baselines.
[4] https://en.wikipedia.org/wiki/Template:Grading_scheme
[5] https://github.com/ariya/phantomjs/blob/master/examples/rasterize.js

respectively. For the BILSTM component, the documents are preprocessed as described in Shen et al. [17], where an article is divided into sentences and tokenized using *NLTK* [1]. Words appearing more than 20 times are retained when building the vocabulary. All other words are replaced by the special UNK token. We use the pretrained *GloVe* [15] 50-dimensional word embeddings to represent words. For words not 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. We set the LSTM hidden layer size to 256. The concatenation of the forward and backward LSTMs thus gives us 512 dimensions for the document embedding. A dropout layer is applied at the sentence and document level, respectively, with a probability of 0.5.

For INCEPTION, we adopt data augmentation techniques in the training with a "nearest" filling mode, a zoom range of 0.1, a width shift range of 0.1, and a height shift range of 0.1. As the original screenshots are 1,000×2,000 pixels, they are resized to 500×500 to feed into INCEPTION. A dropout layer is applied with a probability of 0.5. Then, an average pooling layer is applied, which produces a 2,048 dimensional representation.

For the JOINT model, we get a representation of 2,560 dimensions by concatenating the 512 dimensional representation from the BILSTM with the 2,048 dimensional representation from INCEPTION. The dropout layer is applied to the two components with a probability of 0.5. For BILSTM, we use a mini-batch size of 128 and a learning rate of 0.001. For both INCEPTION and JOINT, we use a mini-batch size of 16 and a learning rate of 0.0001. All hyper-parameters were set empirically over the development data, and the models were optimized using the Adam optimizer [12].

In the training phase, the weights in INCEPTION are initialized by parameters pretrained on ImageNet, and the weights in BILSTM are randomly initialized (except for the word embeddings). We train each model for 50 epochs. To prevent overfitting, we adopt early stopping and stop training if the model performance on the development set does not improve for 20 epochs. For evaluation, we use (micro-)accuracy, following previous studies [6, 11].

4.3 Baseline Approaches

We compare our models against the following five baselines:

- MAJORITY: label test samples with the majority class from the training data.
- BENCHMARK: a benchmark method from the literature [6] that uses structural features (e.g., article length and the number of references) and readability scores as features in a random forest classifier.
- DOC2VEC: a 4-layer feed-forward classification model that uses doc2vec [13] to learn document embeddings.
- BILSTM (textual features only): generate document representations via a bidirectional LSTM (described in Section 2).
- BILSTM$^+$ [17]: supplementation of BILSTM with hand-crafted features.
- INCEPTION$_{\text{FIXED}}$ (visual features only): the frozen INCEPTION model, where only parameters in the last layer are fine-tuned during training. The INCEPTION$_{\text{FIXED}}$ model can reveal how much information an Inception V3 network can learn without updating its parameters (except for the last layer).

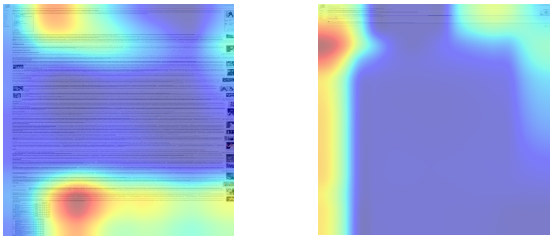


Figure 2: Heatmap overlapped onto screenshots of FA and Stub. Best viewed in color.

The hyper-parameters of BENCHMARK, DOC2VEC, BILSTM, and BILSTM$^+$ are based on the corresponding papers except that we fine-tune the feed forward layer of DOC2VEC on the development set and train the model 300 epochs on Wikipedia.

4.4 Experimental Results

Table 1 shows the model performance over the Wikipedia dataset, in terms of the average accuracy on the test set (along with the standard deviation) over 10 runs, with different random initializations. From Table 1, we observe that BILSTM, INCEPTION, and JOINT outperform all four baselines. INCEPTION achieves 2.9% higher accuracy than BILSTM. The performance of JOINT achieves an accuracy of 59.4%, which is 5.3% higher than using textual features alone (BILSTM) and 2.4% higher than using visual features alone (INCEPTION). Based on a one-tailed Wilcoxon signed-rank test, the performance of JOINT is statistically significant ($p < 0.05$). This shows that the textual and visual features complement each other, achieving state-of-the-art results in combination.

Table 2 shows the confusion matrix of JOINT on Wikipedia. We can see that more than 50% of documents for each quality class are correctly classified, except for the C class where more documents are misclassified into B. Analysis shows that when misclassified, documents are usually misclassified into adjacent quality classes, which can be explained by the Wikipedia grading scheme, where the criteria for adjacent quality classes are more similar.[6]

To better understand the performance of INCEPTION, we generated the gradient-based class activation map [16], by maximizing the outputs of each class in the penultimate layer, as shown in Figure 2. We can see that INCEPTION identifies the two most important regions (one at the top corresponding to the table of contents, and the other at the bottom, capturing both document length and references) that contribute to the FA class prediction; it also finds that (the lack of) images/the link bar down the left side of the document are the most important for Stub class prediction.

4.5 Experiment on Peer Review Dataset

To further verify the general applicability of our proposed model, we perform experiments on the Peer-Review-based dataset (Peer Review hereafter) of Kang et al. [11]. The Peer Review dataset consists of three subsets of academic articles, from the three subject areas of: Artificial Intelligence (cs.ai), Computation and Language (cs.cl), and Machine Learning (cs.lg). We use the pre-defined data

[6] Suggesting that ordinal regression should boost accuracy, but preliminary experiments with various methods led to no improvement over simple classification.

		Majority	Benchmark	Doc2Vec	Inception$_{fixed}$	biLSTM	biLSTM$^+$	Inception	Joint
Wikipedia		16.7%	46.7±0.34%	23.2±1.41%	43.7±0.51	54.1±0.47%	57.2±0.48%	57.0±0.63%	**59.4±0.47%**†
Peer Review	cs.ai	92.2%	92.6%	73.3±9.81%	92.3±0.29	91.5±1.03%	92.1±1.06%	92.8±0.79%	**93.4±1.07%**†
	cs.cl	68.9%	75.7%	66.2±8.38%	75.0±1.95	76.2±1.30%	76.8±1.67%	76.2±2.92%	**77.1±3.10%**
	cs.lg	67.9%	70.7%	64.7±9.08%	73.9±1.23	**81.1±0.83%**	80.0±2.30%	79.3±2.94%	79.9±2.54%

Table 1: Experimental results. The best result for each dataset is indicated in bold, and marked with "†" if it is significantly higher than the second best result (based on a one-tailed Wilcoxon signed-rank test; $p < 0.05$). The results of Benchmark on the Peer Review dataset are from the original paper, where the standard deviation values were not reported.

Quality	FA	GA	B	C	Start	Stub
FA	397	83	20	0	0	0
GA	112	299	65	22	2	0
B	23	53	253	75	44	7
C	5	33	193	124	100	12
Start	1	6	36	85	239	84
Stub	0	0	6	7	63	345

Table 2: Confusion matrix of the Joint model on Wikipedia. Rows are the actual quality classes and columns are the predicted quality classes. The gray cells are correct predictions.

splits for each of the three subsets and labels of these subsets are accepted and rejected, wherein the accepted ratios are roughly 10%, 30%, and 32%, respectively.

We use a Benchmark model [11] that uses hand-crafted features, such as the number of references and TF-IDF weighted bag-of-words, to build a classifier based on the best of logistic regression, multi-layer perception, and AdaBoost. Table 1 shows that Inception and biLSTM achieve similar performance on Peer Review, showing that textual and visual representations are equally discriminative: Inception and biLSTM are indistinguishable over cs.cl; biLSTM achieves 1.8% higher accuracy over cs.lg, while Inception achieves 1.3% higher accuracy over cs.ai. Joint achieves the highest accuracy on cs.ai and cs.cl by combining textual and visual representations (with statistical significance for cs.ai). This, again, confirms that textual and visual features complement each other, and together they achieve state-of-the-art results.

5 CONCLUSIONS

We proposed to use visual renderings of documents to capture implicit document quality indicators, such as font choices, images, and visual layout, which are not captured in textual content. We applied neural network models to capture visual features given visual renderings of documents. Experimental results show that we achieve a 2.9% higher accuracy than state-of-the-art approaches based on textual features over the Wikipedia dataset. We further proposed a joint model, combining textual and visual representations, to predict the quality of a document. Experimental results show that our joint model outperforms the visual-only model and the text only model on the Wikipedia dataset, which underlines the feasibility of assessing document quality via visual features, and the complementarity of visual and textual document representations for quality assessment. Experimental results over the Peer Review dataset further verifies the general applicability of our proposed model.

REFERENCES

[1] Steven Bird. 2006. NLTK: The Natural Language Toolkit. In *ACL*. 69–72.
[2] François Chollet. 2017. Xception: deep learning with depthwise separable convolutions. In *CVPR*. 1800–1807.
[3] Daniel Hasan Dalip, Marcos André Gonçalves, Marco Cristo, and Pável Calado. 2017. A general multiview framework for assessing the quality of collaboratively created content on Web 2.0. *Journal of the Association for Information Science and Technology* 68, 2 (2017), 286–308.
[4] Daniel Hasan Dalip, Harlley Lima, Marcos André Gonçalves, Marco Cristo, and Pável Calado. 2014. Quality assessment of collaborative content with minimal information. In *JCDL*. 201–210.
[5] Quang Vinh Dang and Claudia-Lavinia Ignat. 2017. An end-to-end learning solution for assessing the quality of Wikipedia articles. In *International Symposium on Open Collaboration*. 4:1–4:10.
[6] Quang-Vinh Dang and Claudia-Lavinia Ignat. 2016. Measuring quality of collaboratively edited documents: the case of Wikipedia. In *IEEE International Conference on Collaboration and Internet Computing*. 266–275.
[7] Quang-Vinh Dang and Claudia-Lavinia Ignat. 2016. Quality assessment of Wikipedia articles without feature engineering. In *JCDL*. 27–30.
[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*. 770–778.
[9] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9, 8 (1997), 1735–1780.
[10] Jia-Bin Huang. 2018. Deep Paper Gestalt. *CoRR* abs/1812.08775 (2018).
[11] Dongyeop Kang, Waleed Ammar, Bhavana Dalvi, Madeleine van Zuylen, Sebastian Kohlmeier, Eduard H. Hovy, and Roy Schwartz. 2018. A dataset of peer reviews (PeerRead): collection, insights and NLP applications. In *NAACL-HLT*. 1647–1661.
[12] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
[13] Quoc V. Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *ICML*. 1188–1196.
[14] Nedim Lipka and Benno Stein. 2010. Identifying featured articles in Wikipedia: writing style matters. In *WWW*. 1147–1148.
[15] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: global vectors for word representation. In *EMNLP*. 1532–1543.
[16] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In *ICCV*. 618–626.
[17] Aili Shen, Jianzhong Qi, and Timothy Baldwin. 2017. A hybrid model for quality assessment of Wikipedia articles. In *Australasian Language Technology Association Workshop*. 43–52.
[18] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* abs/1409.1556 (2014).
[19] Klaus Stein and Claudia Hess. 2007. Does it matter who contributes: a study on featured articles in the German Wikipedia. In *HYPERTEXT*. 171–174.
[20] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2016. Rethinking the Inception architecture for computer vision. In *CVPR*. 2818–2826.
[21] Se Wang and Mizuho Iwaihara. 2011. Quality evaluation of Wikipedia articles through edit history and editor groups. In *APWeb*. 188–199.
[22] 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 *ACM Conference on Computer Supported Cooperative Work & Social Computing*. 743–756.
[23] Morten Warncke-Wang, Dan Cosley, and John Riedl. 2013. Tell me more: an actionable quality model for Wikipedia. In *International Symposium on Open Collaboration*. 8:1–8:10.
[24] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alexander J. Smola, and Eduard H. Hovy. 2016. Hierarchical attention networks for document classification. In *HLT-NAACL*. 1480–1489.