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, Victoria, Australia
ailis@student.unimelb.edu.au, salehi.b@unimelb.edu.au, tb@ldwin.net, jianzhong.qi@unimelb.edu.au

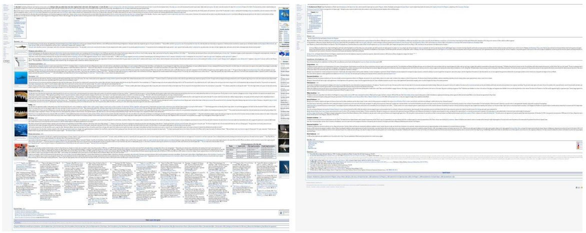
Abstract

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 and academic papers. 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. Experimental results over two datasets reveal that textual and visual features are complementary, achieving state-of-the-art results.

Introduction

The task of document quality assessment is to automatically assess a document according to some predefined inventory of quality labels. This can take many forms, including essay scoring (quality = language quality, coherence, and relevance to a topic), job application filtering (quality = suitability for role + visual/presentational quality of the application), or answer selection in community question answering (quality = actionability + relevance of the answer to the question). In the case of this paper, we focus on document quality assessment in two contexts: Wikipedia document quality classification, and whether a paper submitted to a conference was accepted or not.

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 in the case of Wikipedia, it is particularly pertinent, as any edit potentially has implications for the quality label of that document (and around 10 English Wikipedia documents are edited per second[1]). Furthermore, when the quality assessment task is decentralized (as in the case of Wikipedia and academic paper assessment), 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.



(a) Featured article (b) Lower quality article

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

Current studies on document quality assessment mainly focus on textual features. For example, Warncke-Wang et al. (2015) examine features such as the article length and the number of headings to predict the quality class of a Wikipedia article. In contrast to these studies, in this paper, we propose to combine text 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 Figure 1a has higher quality than Figure 1b, 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 (Szegedy et al. 2016) over visual renderings of documents, while our textual model is based on a hierarchical biLSTM. We further combine the two into a joint model. We perform experiments on two datasets: a Wikipedia dataset novel to this paper, and an arXiv dataset provided by Kang et al. (2018) split into three sub-parts based on subject category. 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 3/4 of our datasets.

This paper makes the following contributions:

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

(i) this is the first study to use visual renderings of documents to capture implicit quality indicators not present in the document text, such as document visual layout; experimental results show that we can obtain a 2.9% higher accuracy using only visual renderings of documents compared with using only textual features over a Wikipedia dataset, and we can obtain competitive results over an arXiv dataset.

(ii) we further propose a joint model to predict document quality combining visual and textual features; we observe further improvements on the Wikipedia dataset and on two of the three arXiv subsets, indicating that visual and textual features are complementary.

(iii) we construct a large-scale Wikipedia dataset with full textual data, visual renderings, and quality class labels; we also supplement the existing arXiv datasets with visual renderings of each document.

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

Related Work

A variety of approaches have been proposed for document quality assessment across different domains: Wikipedia article quality assessment, academic paper rating, content quality assessment in community question answering (cQA), and essay scoring. Among these approaches, some use hand-crafted features while others use neural networks to learn features from documents. For each domain, we first briefly describe feature-based approaches and then review neural network-based approaches.

Wikipedia article quality assessment: 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. 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 (2010) use writing styles represented by binarized character trigram features to identify featured articles. Warncke-Wang, Cosley, and Riedl (2013) and Warncke-Wang et al. (2015) explore the number of headings, images, and references in the article. Dang and Ignat (2016a) use nine readability scores, such as the percentage of difficult words in the document, to measure the quality of the article. Meta-data features, which are indirect indicators of article quality, are usually extracted from revision history, and the interaction between editors and articles. For example, one heuristic that has been proposed is that higher-quality articles have more edits (Dalip et al. 2017; Dalip et al. 2014). Wang and Iwaihara (2011) 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 (2007) 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 (2016b) use a version of doc2vec (Le and Mikolov 2014) to represent articles, and feed the document embeddings into a four hidden layer neural network. Shen, Qi, and Baldwin (2017) first obtain sentence representations by averaging words within a sentence, and then apply a biLSTM (Hochreiter and Schmidhuber 1997) to learn a document-level representation, which is combined with hand-crafted features as side information. Dang and Ignat (2017) exploit two stacked biLSTMs to learn document representations.

Academic paper rating: Academic paper rating is a relatively new task in NLP/AI, with the basic formulation being to automatically predict whether to accept or reject a paper. Kang et al. (2018) explore hand-crafted features, such as the length of the title, whether specific words (such as *outperform*, *state-of-the-art*, and *novel*) appear in the abstract, and an embedded representation of the abstract as input to different downstream learners, such as logistic regression, decision tree, and random forest. Yang et al. (2018) exploit a modularized hierarchical convolutional neural network (CNN), where each paper section is treated as a module. For each paper section, they train an attention-based CNN, and an attentive pooling layer is applied to the concatenated representation of each section, which is then fed into a softmax layer.

Content quality assessment in cQA: Automatic quality assessment in cQA is the task of determining whether an answer is of high quality, selected as the best answer, or ranked higher than other answers. To measure answer content quality in cQA, researchers have exploited various features from different sources, such as the answer content itself, the answerer's profile, interactions among users, and usage of the content. The most common feature used is the answer length (Jeon et al. 2006; Suryanto et al. 2009), with other features including: syntactic and semantic features, such as readability scores. (Agichtein et al. 2008); similarity between the question and the answer at lexical, syntactic, and semantic levels (Agichtein et al. 2008; Belinkov et al. 2015; Hou et al. 2015); or user data (e.g., a user's status points or the number of answers written by the user). There have also been approaches using neural networks. For example, Suggu et al. (2016) combine CNN-learned representations with hand-crafted features to predict answer quality. Zhou et al. (2015) use a 2-dimensional CNN to learn the semantic relevance of an answer to the question, and apply an LSTM to the answer sequence to model thread context. Guzmán, Màrquez, and Nakov (2016) and Guzmán, Nakov, and Màrquez (2016) model the problem similarly to machine translation quality estimation, treating answers as competing translation hypotheses and the question as the reference translation, and apply neural machine translation to the problem.

Essay scoring: Automated essay scoring is the task of assigning a score to an essay, usually in the context of assessing the language ability of a language learner. The quality of an essay is affected by the following four primary dimensions: topic relevance, organization and coherence, word usage and sentence complexity, and grammar and mechanics. To measure whether an essay is relevant to its “prompt” (the description of the essay topic), lexical and semantic overlap is commonly used (Persing and Ng 2014; Phandi, Chai, and Ng 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. Cummins, Zhang, and Briscoe (2016) use sentence structure features to measure sentence variety. The effects of grammatical and mechanic errors on the quality of an essay are measured via word and part-of-speech n-gram features and “mechanics” features (Persing and Ng 2013) (e.g., spelling, capitalization, and punctuation), respectively. Taghipour and Ng (2016), Alikaniotis, Yannakoudakis, and Rei (2016), and Tay et al. (2018) use an LSTM to obtain an essay representation, which is used as the basis for classification. Similarly, Dong, Zhang, and Yang (2017) utilize a CNN to obtain sentence representation and an LSTM to obtain essay representation, with an attention layer at both the sentence and essay levels.

The Proposed Joint Model

We treat document quality assessment as a classification problem, i.e., given a document, we predict its quality class (e.g., whether an academic paper should be accepted or rejected). The proposed model is a joint model that integrates visual features learned through Inception V3 with textual features learned through a biLSTM. In this section, we present the details of the visual and textual embeddings, and finally describe how we combine the two. We return to discuss hyper-parameter settings and the experimental configuration in the Experiments section.

Visual Embedding Learning

A wide range of models have been proposed to tackle the image classification task, such as VGG (Simonyan and Zisserman 2014), ResNet (He et al. 2016), Inception V3 (Szegedy et al. 2016), and Xception (Chollet 2017). However, to the best of our knowledge, there is no existing work that has proposed to use visual renderings of documents to assess document quality. 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.

Based on the observation that it is difficult to decide what types of convolution to apply to each layer (such as 3×3 or 5×5), the basic Inception model applies multiple convolution filters in parallel and concatenates the resulting features, which are fed into the next layer. This has the benefit of capturing both local features through smaller convolutions and abstracted features through larger convolutions. INCEPTION is a hybrid of multiple Inception models of different architectures. To reduce computational cost, INCEPTION also modifies the basic model by applying a 1×1 convolution to the input and factorizing larger convolutions into smaller ones.

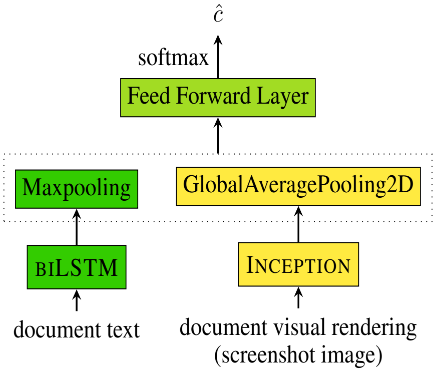


Figure 2: Overview of the proposed model.

Textual Embedding Learning

We adopt a bi-directional LSTM model to generate textual embeddings for document quality assessment, following the method of Shen, Qi, and Baldwin (2017) (“BILSTM” hereafter). The input to BILSTM is a textual document, and the output is a textual embedding, which will later integrate with the visual embedding.

For BILSTM, each word is represented as a word embedding (Bengio et al. 2003), and an average-pooling layer is applied to the word embeddings to obtain the sentence embedding, which is fed into a bi-directional LSTM to generate the document embedding from the sentence embeddings. Then a max-pooling layer is applied to select the most salient features from the component sentences.

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, as shown in Figure 2. We optimize our model based on cross-entropy loss.

Experiments

In this section, we first describe the two datasets used in our experiments: (1) Wikipedia, and (2) arXiv. Then, we report the experimental details and results.

Datasets

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

[2] http://www.image-net.org/

Class	Train	Dev	Test	Total
FA	4000	500	500	5000
GA	4000	500	500	5000
B	4000	500	455	4955
C	4000	500	467	4967
Start	4000	500	451	4951
Stub	4000	500	421	4921
Total	24000	3000	2794	29794

Table 1: Wikipedia dataset.

in the Wikipedia grading scheme page.[3] The quality class of a Wikipedia article is assigned by Wikipedia reviewers or any registered user, who can discuss through the article's talk page[4] to reach consensus. We constructed the dataset by first crawling all articles from each quality class repository, e.g., we get FA articles by crawling pages from the FA repository: `https://en.wikipedia.org/wiki/Category:Featured_articles`. This resulted in around 5K FA, 28K GA, 212K B, 533K C, 2.6M Start, and 3.2M Stub articles.

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 contains markup relating to the document category such as {*Featured Article*} or {*geo-stub*}, which reveals the label, we remove such information. We additionally randomly partitioned this dataset into training, development, and test splits based on a ratio of 8:1:1. Details of the dataset are summarized in Table 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.

arXiv dataset The arXiv dataset (Kang et al. 2018) consists of three subsets of academic articles under the arXiv repository of Computer Science (cs), from the three subject areas of: Artificial Intelligence (cs.ai), Computation and Language (cs.cl), and Machine Learning (cs.lg). In line with the original dataset formulation (Kang et al. 2018), a paper is considered to have been accepted (i.e. is positively labeled) if it matches a paper in the DBLP database or is otherwise accepted by any of the following conferences: ACL, EMNLP, NAACL, EACL, TACL, NIPS, ICML, ICLR, or AAAI. Failing this, it is considered to be rejected (noting that some of the papers may not have been submitted to one of these conferences). The median numbers of pages for papers in cs.ai, cs.cl, and cs.lg are 11, 10, and 12, respectively. To make sure each page in the PDF file has the same size in the screenshot, we crop the PDF file of a paper to the first 12; we pad the PDF file with blank pages if a PDF file has less than 12 pages, using the PyPDF2 Python package.[6] We then use ImageMagick[7] to convert the 12-page PDF file to a single 1,000×2,000 pixel screenshot. Table 2 details this dataset, where the "Accepted" column denotes the percentage of positive instances (accepted papers) in each subset.

Subject	Accepted	Train	Dev	Test	Total
cs.ai	10%	3682	205	205	4092
cs.cl	30%	2374	132	132	2638
cs.lg	32%	4543	252	253	5048

Table 2: arXiv dataset. "Accepted" indicates the proportion of accepted papers in the given subject.

Experimental Setting

As discussed above, our model has two main components — biLSTM and Inception— which generate textual and visual representations, respectively. For the biLSTM component, the documents are preprocessed as described in Shen, Qi, and Baldwin (2017), where an article is divided into sentences and tokenized using NLTK (Bird 2006). 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 pre-trained GloVe (Pennington, Socher, and Manning 2014) 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 have the size of 1,000×2,000 pixels, they are resized to 500×500 to feed into Inception, where the input shape is (500, 500, 3). A dropout layer is applied with a probability of 0.5. Then, a GlobalAveragePooling2D 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

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

[4] `https://en.wikipedia.org/wiki/Help:Talk_pages`

[5] `https://github.com/ariya/phantomjs/blob/master/examples/rasterize.js`

[6] `https://pypi.org/project/PyPDF2/`

[7] `https://www.imagemagick.org/script/index.php`

0.001. For both INCEPTION and joint model, 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 (Kingma and Ba 2014).

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. However, to prevent overfitting, we adopt early stopping, where we stop training the model if the performance on the development set does not improve for 20 epochs. For evaluation, we use (micro-)accuracy, following previous studies (Dang and Ignat 2016a; Kang et al. 2018).

Baseline Approaches

We compare our models against the following five baselines:

- MAJORITY: the model labels all test samples with the majority class of the training data.
- BENCHMARK: a benchmark method from the literature. In the case of Wikipedia, this is Dang and Ignat (2016a), who use structural features and readability scores as features to build a random forest classifier; for arXiv, this is Kang et al. (2018), who use hand-crafted features, such as the number of references and TF-IDF weighted bag-of-words in abstract, to build a classifier based on the best of logistic regression, multi-layer perception, and AdaBoost.
- DOC2VEC: doc2vec (Le and Mikolov 2014) to learn document embeddings with a dimension of 500, and a 4-layer feed-forward classification model on top of this, with 2000, 1000, 500, and 200 dimensions, respectively.
- BILSTM: first derive a sentence representation by averaging across words in a sentence, then feed the sentence representation into a biLSTM and a maxpooling layer over output sequence to learn a document level representation with a dimension of 512, which is used to predict document quality.
- INCEPTION$_{\text{FIXED}}$: the frozen INCEPTION model, where only parameters in the last layer are fine-tuned during training.

The hyper-parameters of BENCHMARK, DOC2VEC, and BILSTM are based on the corresponding papers except that: (1) we fine-tune the feed forward layer of DOC2VEC on the development set and train the model 300 epochs on Wikipedia and 50 epochs on arXiv; (2) we do not use hand-crafted features for BILSTM as we want the baselines to be comparable to our models, and the main focus of this paper is not to explore the effects of hand-crafted features (e.g., see Shen, Qi, and Baldwin (2017)).

Experimental Results

Table 3 shows the performance of the different models over our two datasets, in the form of the average accuracy on the test set (along with the standard deviation) over 10 runs, with different random initializations.

On Wikipedia, we observe that the performance of BILSTM, INCEPTION, and JOINT is much better than that of 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.

For arXiv, baseline methods MAJORITY, BENCHMARK, and INCEPTION$_{\text{FIXED}}$ outperform BILSTM over cs.ai, in large part because of the class imbalance in this dataset (90% of papers are rejected). Surprisingly, INCEPTION$_{\text{FIXED}}$ is better than MAJORITY and BENCHMARK over the arXiv cs.lg subset, which verifies the usefulness of visual features, even when only the last layer is fine-tuned. Table 3 also shows that INCEPTION and BILSTM achieve similar performance on arXiv, 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. Once again, the JOINT model achieves the highest accuracy on cs.ai and cs.cl by combining textual and visual representations (at a level of 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. On arXiv cs.lg, JOINT achieves a 0.6% higher accuracy than INCEPTION by combining visual features and textual features, but BILSTM achieves the highest accuracy. One characteristic of cs.lg documents is that they tend to contain more equations than the other two arXiv datasets, and preliminary analysis suggests that the BILSTM is picking up on a correlation between the volume/style of mathematical presentation and the quality of the document.

Analysis

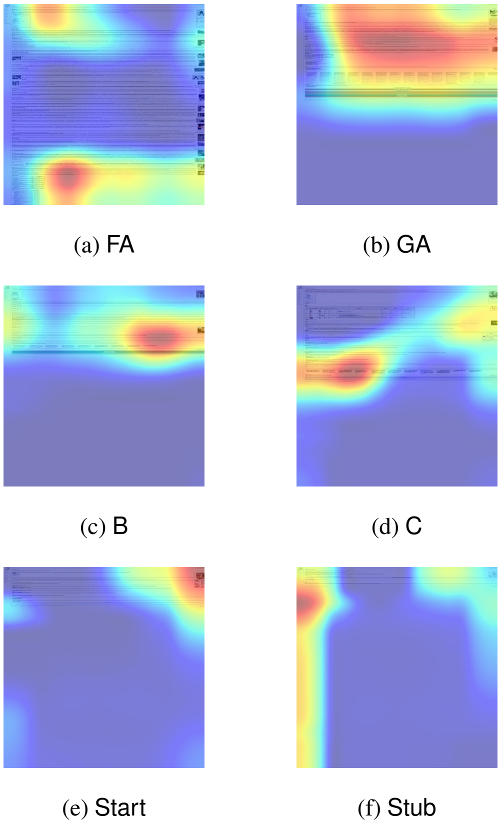
In this section, we first analyze the performance of INCEPTION and JOINT. We also analyze the performance of different models on different quality classes. The high-level representations learned by different models are also visualized and discussed. As the Wikipedia test set is larger and more balanced than that of arXiv, our analysis will focus on Wikipedia.

INCEPTION

To better understand the performance of INCEPTION, we generated the gradient-based class activation map (Selvaraju et al. 2017), by maximizing the outputs of each class in the penultimate layer, as shown in Figure 3. From Figure 3a and Figure 3b, 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, and a region in the upper half of the image that contributes to the GA class prediction (capturing the length of the article body). From Figure 3c and Figure 3d, we can see that the most important regions in

		Majority	Benchmark	Doc2Vec	Inception$_{\text{FIXED}}$	biLSTM	Inception	Joint
Wikipedia		16.7%	46.7±0.34%	23.2±1.41%	43.7±0.51	54.1±0.47%	57.0±0.63%	**59.4±0.47%**†
arXiv	cs.ai	92.2%	92.6%	73.3±9.81%	92.3±0.29	91.5±1.03%	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%	76.2±2.92%	**77.1±3.10%**
	cs.lg	67.9%	70.7%	64.7±9.08%	73.9±1.23	**81.1±0.83%**	79.3±2.94%	79.9±2.54%

Table 3: 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 arXiv dataset are from the original paper, where the standard deviation values were not reported. All neural models except for Inception$_{\text{FIXED}}$ have larger standard deviation values on arXiv than Wikipedia, which can be explained by the small size of the arXiv test set.



(a) FA

(b) GA

(c) B

(d) C

(e) Start

(f) Stub

Figure 3: Heatmap overlapped onto screenshots of each Wikipedia quality class. Best viewed in color.

terms of B and C class prediction capture images (down the left and right of the page, in the case of B and C), and document length/references. From Figure 3e and Figure 3f, we can see that Inception finds that images in the top right corner are the strongest predictor of Start class prediction, and (the lack of) images/the link bar down the left side of the document are the most important for Stub class prediction.

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 4: Confusion matrix of the Joint model on Wikipedia. Rows are the actual quality classes and columns are the predicted quality classes. The diagonal (gray cells) indicates correct predictions.

Quality	Metric	biLSTM	Inception	Joint
FA	$\mathcal{P}$	**76.6**	74.8	73.8
	$\mathcal{R}$	72.0	68.2	**79.4**
	$\mathcal{F}_{\beta=1}$	74.2	71.3	**76.5**
GA	$\mathcal{P}$	51.3	57.7	**63.1**
	$\mathcal{R}$	**59.8**	59.0	**59.8**
	$\mathcal{F}_{\beta=1}$	55.2	58.3	**61.4**
B	$\mathcal{P}$	37.6	41.8	**44.2**
	$\mathcal{R}$	42.4	44.0	**55.6**
	$\mathcal{F}_{\beta=1}$	39.9	42.9	**49.2**
C	$\mathcal{P}$	36.3	38.9	**39.6**
	$\mathcal{R}$	27.0	**36.0**	26.6
	$\mathcal{F}_{\beta=1}$	31.0	**37.4**	31.8
Start	$\mathcal{P}$	48.2	49.4	**53.3**
	$\mathcal{R}$	44.8	**57.2**	53.0
	$\mathcal{F}_{\beta=1}$	46.4	53.0	**53.1**
Stub	$\mathcal{P}$	71.9	**83.3**	77.0
	$\mathcal{R}$	78.9	78.2	**81.9**
	$\mathcal{F}_{\beta=1}$	75.2	**80.7**	79.4

Table 5: Precision ("$\mathcal{P}$"), recall ("$\mathcal{R}$"), and F1 ("$\mathcal{F}_{\beta=1}$") of biLSTM, Inception, and Joint on Wikipedia.

Joint

Table 4 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

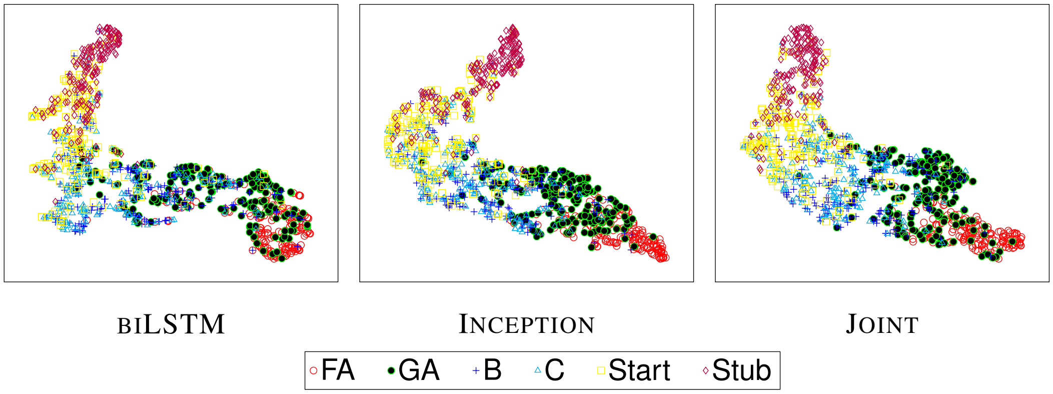


Figure 4: t-SNE scatter plot of Wikipedia article representations (representations from penultimate layer of each model, 200 random samples from each quality class; best viewed in color)

quality classes are more similar.[8]

We also provide a breakdown of precision ("$\mathcal{P}$"), recall ("$\mathcal{R}$"), and F1 score ("$\mathcal{F}_{\beta=1}$") for biLSTM, Inception, and Joint across the quality classes in Table 5. We can see that Joint achieves the highest accuracy in 11 out of 18 cases. It is also worth noting that all models achieve higher scores for FA, GA, and Stub articles than B, C and Start articles. This can be explained in part by the fact that FA and GA articles must pass an official review based on structured criteria, and in part by the fact that Stub articles are usually very short, which is discriminative for Inception, and Joint. All models perform worst on the B and C quality classes. It is difficult to differentiate B articles from C articles even for Wikipedia contributors. As evidence of this, when we crawled a new dataset including talk pages with quality class votes from Wikipedia contributors, we found that among articles with three or more quality labels, over 20% percent of B and C articles have inconsistent votes from Wikipedia contributors, whereas for FA and GA articles the number is only 0.7%.

We further visualize the learned document representations of biLSTM, Inception, and Joint in the form of a t-SNE plot (van der Maaten and Hinton 2008) in Figure 4. The degree of separation between Start and Stub achieved by Inception is much greater than for biLSTM, with the separation between Start and Stub achieved by Joint being the clearest among the three models. Inception and Joint are better than biLSTM at separating Start and C. Joint achieves slightly better performance than Inception in separating GA and FA. We can also see that it is difficult for all models to separate B and C, which is consistent with the findings of Tables 4 and 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 Wikipedia, and performance competitive with or surpassing state-of-the-art approaches over arXiv. 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 in all cases, and the text-only model on Wikipedia and two subsets of arXiv. These results underline the feasibility of assessing document quality via visual features, and the complementarity of visual and textual document representations for quality assessment.

References

[Agichtein et al. 2008] Agichtein, E.; Castillo, C.; Donato, D.; Gionis, A.; and Mishne, G. 2008. Finding high-quality content in social media. In *WSDM*, 183–194.

[Alikaniotis, Yannakoudakis, and Rei 2016] Alikaniotis, D.; Yannakoudakis, H.; and Rei, M. 2016. Automatic text scoring using neural networks. In *ACL*, 715–725.

[Attali and Burstein 2004] Attali, Y., and Burstein, J. 2004. Automated essay scoring with e-rater® v. 2.0. *ETS Research Report Series* 2004(2).

[Belinkov et al. 2015] Belinkov, Y.; Mohtarami, M.; Cyphers, S.; and Glass, J. R. 2015. VectorSLU: A continuous word vector approach to answer selection in community question answering systems. In *SemEval@NAACL-HLT*, 282–287.

[Bengio et al. 2003] Bengio, Y.; Ducharme, R.; Vincent, P.; and Jauvin, C. 2003. A neural probabilistic language model. *Journal of Machine Learning Research* 3:1137–1155.

[Bird 2006] Bird, S. 2006. NLTK: the natural language toolkit. In *ACL*, 69–72.

[Chollet 2017] Chollet, F. 2017. Xception: Deep learning with depthwise separable convolutions. In *CVPR*, 1800–1807.

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

[Cummins, Zhang, and Briscoe 2016] Cummins, R.; Zhang, M.; and Briscoe, T. 2016. Constrained multi-task learning for automated essay scoring. In *ACL*, 789–799.

[Dalip et al. 2014] Dalip, D. H.; Lima, H.; Gonçalves, M. A.; Cristo, M.; and Calado, P. 2014. Quality assessment of collaborative content with minimal information. In *JCDL*, 201–210.

[Dalip et al. 2017] Dalip, D. H.; Gonçalves, M. A.; Cristo, M.; and Calado, P. 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):286–308.

[Dang and Ignat 2016a] Dang, Q.-V., and Ignat, C.-L. 2016a. Measuring quality of collaboratively edited documents: the case of Wikipedia. In *The 2nd IEEE International Conference on Collaboration and Internet Computing*, 266–275.

[Dang and Ignat 2016b] Dang, Q.-V., and Ignat, C.-L. 2016b. Quality assessment of Wikipedia articles without feature engineering. In *JCDL*, 27–30.

[Dang and Ignat 2017] Dang, Q. V., and Ignat, C. 2017. An end-to-end learning solution for assessing the quality of Wikipedia articles. In *Proceedings of the 13th International Symposium on Open Collaboration*, 4:1–4:10.

[Dong, Zhang, and Yang 2017] Dong, F.; Zhang, Y.; and Yang, J. 2017. Attention-based recurrent convolutional neural network for automatic essay scoring. In *CoNLL*, 153–162.

[Guzmán, Màrquez, and Nakov 2016] Guzmán, F.; Màrquez, L.; and Nakov, P. 2016. Machine translation evaluation meets community question answering. In *ACL*, 460–466.

[Guzmán, Nakov, and Màrquez 2016] Guzmán, F.; Nakov, P.; and Màrquez, L. 2016. MTE-NN at SemEval-2016 task 3: Can machine translation evaluation help community question answering? In *SemEval@NAACL-HLT*, 887–895.

[He et al. 2016] He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*, 770–778.

[Hochreiter and Schmidhuber 1997] Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural Computation* 9(8):1735–1780.

[Hou et al. 2015] Hou, Y.; Tan, C.; Wang, X.; Zhang, Y.; Xu, J.; and Chen, Q. 2015. HITSZ-ICRC: exploiting classification approach for answer selection in community question answering. In *SemEval@NAACL-HLT*, 196–202.

[Jeon et al. 2006] Jeon, J.; Croft, W. B.; Lee, J. H.; and Park, S. 2006. A framework to predict the quality of answers with non-textual features. In *SIGIR*, 228–235.

[Kang et al. 2018] Kang, D.; Ammar, W.; Dalvi, B.; van Zuylen, M.; Kohlmeier, S.; Hovy, E. H.; and Schwartz, R. 2018. A dataset of peer reviews (peerread): Collection, insights and NLP applications. In *NAACL-HLT*, 1647–1661.

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

[Le and Mikolov 2014] Le, Q. V., and Mikolov, T. 2014. Distributed representations of sentences and documents. In *ICML*, 1188–1196.

[Lipka and Stein 2010] Lipka, N., and Stein, B. 2010. Identifying featured articles in Wikipedia: writing style matters. In *WWW*, 1147–1148.

[Pennington, Socher, and Manning 2014] Pennington, J.; Socher, R.; and Manning, C. D. 2014. GloVe: Global vectors for word representation. In *EMNLP*, 1532–1543.

[Persing and Ng 2013] Persing, I., and Ng, V. 2013. Modeling thesis clarity in student essays. In *ACL*, 260–269.

[Persing and Ng 2014] Persing, I., and Ng, V. 2014. Modeling prompt adherence in student essays. In *ACL*, 1534–1543.

[Phandi, Chai, and Ng 2015] Phandi, P.; Chai, K. M. A.; and Ng, H. T. 2015. Flexible domain adaptation for automated essay scoring using correlated linear regression. In *EMNLP*, 431–439.

[Selvaraju et al. 2017] Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; and Batra, D. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, 618–626.

[Shen, Qi, and Baldwin 2017] Shen, A.; Qi, J.; and Baldwin, T. 2017. A hybrid model for quality assessment of Wikipedia articles. In *Proceedings of the Australasian Language Technology Association Workshop*, 43–52.

[Simonyan and Zisserman 2014] Simonyan, K., and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* abs/1409.1556.

[Stein and Hess 2007] Stein, K., and Hess, C. 2007. Does it matter who contributes: a study on featured articles in the german wikipedia. In *HYPERTEXT*, 171–174.

[Suggu et al. 2016] Suggu, S. P.; Goutham, K. N.; Chinnakotla, M. K.; and Shrivastava, M. 2016. Hand in glove: Deep feature fusion network architectures for answer quality prediction in community question answering. In *COLING*, 1429–1440.

[Suryanto et al. 2009] Suryanto, M. A.; Lim, E.; Sun, A.; and Chiang, R. H. L. 2009. Quality-aware collaborative question answering: methods and evaluation. In *WSDM*, 142–151.

[Szegedy et al. 2016] Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016. Rethinking the Inception architecture for computer vision. In *CVPR*, 2818–2826.

[Taghipour and Ng 2016] Taghipour, K., and Ng, H. T. 2016. A neural approach to automated essay scoring. In *EMNLP*, 1882–1891.

[Tay et al. 2018] Tay, Y.; Phan, M. C.; Tuan, L. A.; and Hui, S. C. 2018. SkipFlow: Incorporating neural coherence features for end-to-end automatic text scoring. In *AAAI*.

[van der Maaten and Hinton 2008] van der Maaten, L., and Hinton, G. 2008. Visualizing data using t-SNE. *JMLR* 9:2579–2605.

[Wang and Iwaihara 2011] Wang, S., and Iwaihara, M. 2011. Quality evaluation of wikipedia articles through edit history and editor groups. In *APWeb*, 188–199.

[Warncke-Wang et al. 2015] Warncke-Wang, M.; Ayukaev, V. R.; Hecht, B.; and Terveen, L. 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, CSCW 2015*, 743–756.

[Warncke-Wang, Cosley, and Riedl 2013] Warncke-Wang, M.; Cosley, D.; and Riedl, J. 2013. Tell me more: an actionable quality model for Wikipedia. In *Proceedings of the 9th International Symposium on Open Collaboration*, 8:1–8:10.

[Yang et al. 2018] Yang, P.; Sun, X.; Li, W.; and Ma, S. 2018. Automatic academic paper rating based on modularized hierarchical convolutional neural network. In *ACL*, 496–502.

[Zhou et al. 2015] Zhou, X.; Hu, B.; Chen, Q.; Tang, B.; and Wang, X. 2015. Answer sequence learning with neural networks for answer selection in community question answering. *arXiv preprint arXiv:1506.06490*.