University of Groningen

Structure-Tags Improve Text Classification for Scholarly Document Quality Prediction

Maillette de Buy Wenniger, Gideon; van Dongen, Thomas; Aedmaa, Eleri; Teun Kruitbosch, Herbert; Valentijn, Edwin A.; Schomaker, Lambert

Structure-Tags Improve Text Classification for Scholarly Document Quality Prediction

Gideon Maillette de Buy Wenniger[†], **Thomas van Dongen**[†], **Eleri Aedmaa**[‡],
Herbert Teun Kruitbosch[‡], **Edwin A. Valentijn**[§], **and Lambert Schomaker**[†]

[†]Bernoulli Institute for Mathematics,Computer Science and Artificial Intelligence
University of Groningen, Groningen, The Netherlands
[‡]Center for Information Technology, University of Groningen, Groningen, The Netherlands
[§]Kapteyn Astronomical Institute, University of Groningen, Groningen, The Netherlands

Abstract

Training recurrent neural networks on long texts, in particular scholarly documents, causes problems for learning. While hierarchical attention networks (HANs) are effective in solving these problems, they still lose important information about the structure of the text. To tackle these problems, we propose the use of HANs combined with *structure-tags* which mark the role of sentences in the document. Adding tags to sentences, marking them as corresponding to title, abstract or main body text, yields improvements over the state-of-the-art for scholarly document quality prediction: substantial gains on average against other models and consistent improvements over HANs without structure-tags. The proposed system is applied to the task of accept/reject prediction on the PeerRead dataset and compared against a recent BiLSTM-based model and joint textual+visual model. It gains 4.7% accuracy over the best of both models on the computation and language domain and loses 2.4% against the best of both on the machine learning domain. Compared to plain HANs, accuracy increases on both domains, with 1.5% and 2% respectively. We also obtain improvements when introducing the tags for prediction of the number of citations for 88k scientific publications that we compiled from the Allen AI S2ORC dataset. For our HAN-system with structure-tags we reach 28.5% explained variance, an improvement of 1.0% over HANs without structure-tags.

1 Introduction

Automatic prediction of the quality of scientific and other texts is a new topic within the field of deep learning. Deep learning has been successfully applied to many natural language processing (NLP) problems including text classification, as well as many computer vision applications including document structure analysis. These successes suggest automatic quality assessment of scientific documents, while still highly ambitious, are feasible for scientific study.

Sequential deep learning models, particularly recurrent neural networks (RNNs), long short-term memories (LSTMs) and their variants, have been particularly successful for applications that require the encoding and/or generation of relatively short sequences of text, typically at most a few sentences, see for example (Rao and Spasojevic, 2016; Rocktĩd'schel et al., 2015). For such applications, including (neural) machine translation (Bahdanau et al., 2014; Luong et al., 2015) and parsing, earlier models not relying on deep learning have typically been beaten by a large margin by their newer deep learning based competitors. This trend has only increased by the newer attention-based models, particularly the transformer model (Vaswani et al., 2017), which are even more apt at using all of the possible context when building encodings of sentences. Transformers are also used as the basis to build general sentence embeddings with the BERT model (Devlin et al., 2018), which are used as a versatile basis on top of which many other applications can be performed with high accuracy.

In comparison to these major successes, the accurate classification of full documents remains more challenging. To be effective, a deep learning model for longer text should fulfill the following three criteria:

1. *Trainability*: The model should be trainable on long texts.

2. *Computational efficiency*: The model should be computationally efficient as well as parallelizable, in order to make efficient use of GPUs.

3. *Rich context*: The model should have access to rich context at sentence and document level

while encoding the text. Therefore it should avoid: 1) the assumption that sentences at different locations are independent, 2) the even even more crippling assumption that words in the document can be modeled as being statistically independent.

Plain sequential models such as RNNs and LSTMs model text as unstructured word sequences . This cause problems on longer texts because of the vanishing gradient and exploding gradient problem (Pascanu et al., 2013), which hampers *trainability*, the first criterion for effectiveness. Gradient bounding methods including gradient clipping (Hochreiter, 1998), can help to reduce these problems, but provide no solution for documents with thousands of words. A general approach towards increasing the trainability of very deep neural networks is adding residual connections, which skip over one or multiple layers (He et al., 2015; Srivastava et al., 2015). But with sequential models for text, this approach does not solve the other important problem of non-parallizability across the sequence direction because of the sequential dependencies. Thus this approach does not fulfill the second requirement of *computational efficiency*. Transformers and BERT are not a good match for long texts either. While not suffering from exploding and vanishing gradients, these models have a computational cost that grows quadratically with sentence length, consequently state-of-the-art BERT implementations limit their input to 512 tokens. Arguably, bag-of-word models, including models performing average pooling over word embeddings form a way to deal effectively with large texts by fulfilling the first two criteria of *trainability* and *computational efficiency*. However, their computational cheapness is achieved at the price of making very strong statistical independence assumptions that hamper the quality of predictions. Thus, these models fail on the third criterion of allowing use of *rich context* while creating an encoding the text.

There is however a group of models that does fulfill all three criteria: hierarchical versions of sequential models, in particular hierarchical attention networks (HANs). HANs use a hierarchical stacking of LSTM models with attention for the sentence and text level. This massively increases parallelization while simultaneously reducing the number of steps the gradient signal needs to be back-propagated during training, increasing learnability. The hierarchical text encodings produced by these models can still take much context into account at every level in the representation, thanks to the use of LSTMs.

HANs are thus highly effective in forming adequate representations of longer texts to be used for text classification and other tasks. However, these hierarchical encoding models of text are still deficient in the use of *structure information* inherent in the text. The reason is simple: these models have only one encoding sub-model per level in the hierarchy, a LSTM in case of HAN. This sub-model is used to encode all the inputs at that level, without access to relevant structure context. For example, the LSTM at the first level encodes all the sentences in the input, with no information about what part of the text these sentences belong to, or about their relative position in the text. In this work we observe that this deficiency can be effectively solved by adding XML-like structure-tags at the beginning and end of each sentence in the input. The effectiveness of our approach is demonstrated on two tasks:

A Paper accept/reject prediction on the Peer-Read dataset (Kang et al., 2018).

B Number of citations prediction for scholarly documents, on a new dataset with 88K articles compiled from the Allen AI S2ORC dataset, more than 23 times larger than datasets used earlier in the literature.

The experiments for both tasks show that using just three tags to mark abstract, title and body text, already provides substantial improvements over a baseline where such information is not provided. Notably, this type of structure-tags used in this first exploration is still restricted, and larger gains can likely be made by further enriching the tag-set. In particular, it would be straightforward enough to add more and more fine-grained tags, as well as tags encoding positional information, somewhat similar in spirit to positional encodings in the transformer model. This does not take away however from our main contribution, a proof of concept that shows that structure-tags can yield substantial improvements to text classification and text-based regression. This is particularly useful in the domain of scholarly document understanding, since while these document are typically long, they are also highly structured.

The rest of the paper is structured as follows. In section 2 we discuss the various existing and alter-

native NLP models for the aforementioned tasks of quality prediction. Section 3 describes the proposed HAN model combined with structure-tags. Section 4 and 5 respectively discuss their use for accept/reject and number of citations prediction.

2 Related Work

Multiple methods have been proposed to estimate the quality of scientific papers. The most common approach is to use the citation counts as a measure of quality, to be predicted by models. Fu and Aliferis (2008) proposed one of the first models which used both the papers content in the form of the paper title, abstract and keywords as well as bibliometric information such as the number of articles for the first author, publication type and quality of first authorâĂŹs institution. Notably they used automated scripts to query web of science for retrieving bibliometric information, even so their final corpus is still relatively modest in size, containing 3788 papers. While Brody et al. (2006) use information that becomes available after publication, like citation count, Fu and Aliferis use only information available before publication by using term-vectors as input to a SVM. IbÃąÃśez et al. (2009) expands upon this research by using several different classification methods. Both the naive Bayes as well as the logistic regression model outperform the model proposed by Fu and Aliferis.

More recent papers use deep learning techniques to predict the citations of papers. Abrishami and Aliakbary (2019) use recurrent neural networks to predict future citations, outperforming all other state-of-the-art methods. However, like the model proposed by Brody et al., this method is only applicable for predicting future citations when some citations are already available.

Limited recent research is available on the subject of predicting the quality of papers using the textual content. One recent method which does use the textual content is proposed by Shen et al. (2019). In this paper, visual and textual content are combined using a CNN and LSTM respectively. The authors make use of the Wikipedia and the arXiv datasets. The authors propose a joint model that classifies the quality of papers. To generate textual embeddings, the authors use a bi-directional LSTM model similar to the one proposed by the same authors in (Shen et al., 2017). The input to the model is the word embeddings of a paper which are obtained using GloVe, and the output is a textual

embedding.

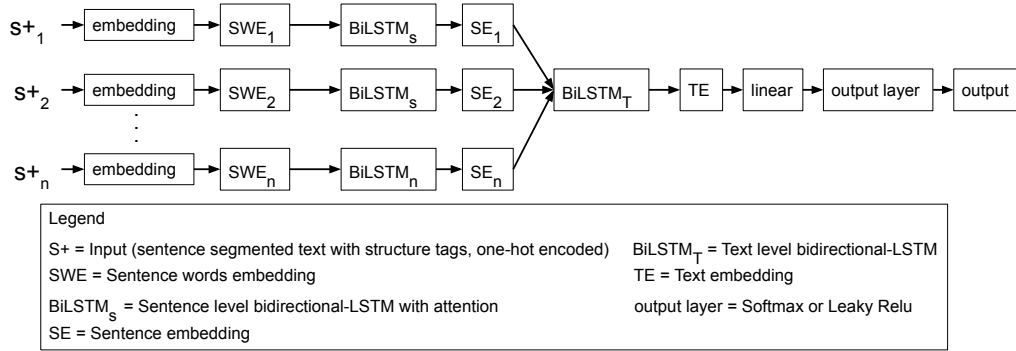Some recent work focuses on predicting the number of citations from both the paper contents text augmented with review text. To do so, Li et al. (2019) create a datset of abstracts and reviews from the ICLR and NIPS conferences. For the ICLR conferences they collected a total of 1739 abstracts with in total of 7171 reviews and for the NIPS conference a total of 384 abstract with in total 1119 reviews. Plank and van Dale (2019) collect a datset of 3427 papers with 12260 reviews. Both papers show improvement in the results from using the review information.
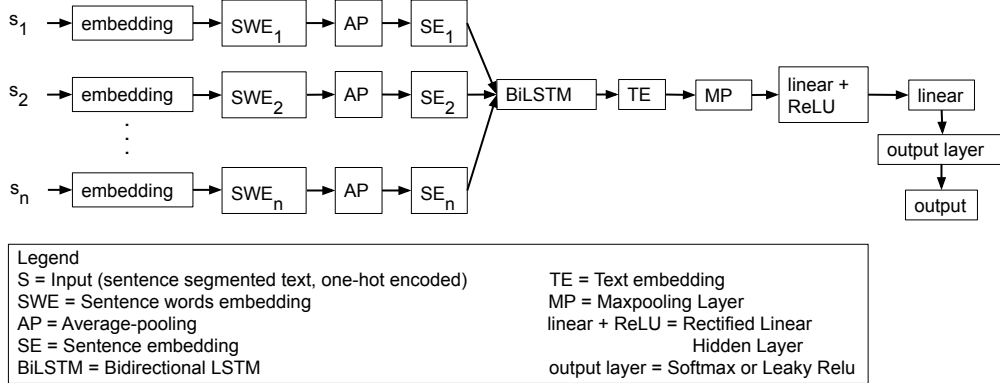
Hierarchical sequential models
Hierarchical versions of sequential models have already been pioneered in the literature for a long time in the form of hierarchical recurrent neural networks (Hihi and Bengio, 1996). More recently however, use of LSTMs instead of RNNs and use of attention resulted in the now popular HAN model (Yang et al., 2016), which was successfully applied to sentiment analysis and different text classification tasks. Recently, (Qiao et al., 2018) used a hybrid hierarchical network, with a convolutional layer plus attention pooling layer to represent the content of entire article sections and an LSTM with attention to merge the section representations into a final document representation. Their approach is tested on the task of predicting aspect scores on papers from aspect-score labeled portion of the PeerRead dataset. Compared to HAN which uses LSTMs on both layers, using a convolution layer with attention in their section encoding restricts the amount of context that is accesible when constructing this encoding.

Adding structure through additional inputs
Our proposed structure-tag framework is most similar in spirit to the approach that been used for automatic translation of multiple source languages to multiple target languages using a unified model (Johnson et al., 2016), in which a special "command token" is used to indicate which kind of translation is desired. Related also is the idea of using multiple embeddings for different types of information, as introduced in the field of neural machine translation by (Sennrich and Haddow, 2016), which was later also exploited in the popular transformer model (Vaswani et al., 2017). In contrast to the latter approaches though which change the embedding layer, like (Johnson et al., 2016) we leave the

S+ = Input (sentence segmented text with structure tags, one-hot encoded)
SWE = Sentence words embedding
BiLSTM$_s$ = Sentence level bidirectional-LSTM with attention
SE = Sentence embedding
BiLSTM$_T$ = Text level bidirectional-LSTM
TE = Text embedding
output layer = Softmax or Leaky Relu

(a) Our model based on HAN.

S = Input (sentence segmented text, one-hot encoded)
SWE = Sentence words embedding
AP = Average-pooling
SE = Sentence embedding
BiLSTM = Bidirectional LSTM
TE = Text embedding
MP = Maxpooling Layer
linear + ReLU = Rectified Linear Hidden Layer
output layer = Softmax or Leaky Relu

(b) Model proposed by Shen et al. (2019).

Figure 1: Most important models compared in this work.

(HAN) model exactly as is and only change the input, however with LSTMs it may be expected that to a large extent adding information through extra tokens or through additional embeddings can achieve the same effect.

3 Models

In this work we use and refine state-of-the-art text-based deep learning models for text classification and regression tasks: accept/reject prediction and number of citations prediction respectively. Our contributions focus on HANs, which we show for these tasks to be equally or better performing than models that use a flat BiLSTM encoder at their core (Shen et al., 2019). Figure 3a shows a diagram of our HAN model with structure-tags added to the input, and Figure 3b shows a diagram of the (Shen et al., 2019) BiLSTM-based model, which is our baseline for comparison. As can be seen from the diagrams, both models use a BiLSTM at the text level that works on embeddings computed for the sentences of the text. However, while HAN uses the sequential order to compute an embedding, the baseline model averages word vectors, disregarding order similar to bag-of-word representations.

For HAN we furthermore propose to add more text structure information. This is done by adding structure-tags at the sentence-level, implemented as special symbols at the start and end of sentences.

3.1 Sentence type tags for more structure

The hierarchical structure of text characterized by structure elements such as sections, paragraphs and sentences and labeling elements such as document titles and section titles reveals important information. Models without hierarchy such as plain RNN/LSTM models ignore this structure, which motivated HAN (Yang et al., 2016).

HAN uses an LSTM with attention to create encodings of each sentence separately and combines this with a second LSTM with attention on top to transform these into an encoding of the entire text. The hierarchical structure of HAN provides several advantages over flat sequential models, i.e. plain RNNs/LSTMs:

1. *Trainability on long texts*: HAN realizes a much smaller amount of steps for back-propagating gradients during training, allowing it to process much longer texts without running into vanishing/exploding gradient prob-

lems. And it can do so while maintaining high-resolution when forming sentence-level encodings. While HAN and our proposed model uses two levels of LSTMs, more levels can be added to model more levels of structure and possibly deal with even longer texts.

2. *Computational efficiency*: The hierarchical structure makes the computations of HAN much better parallelized, since children at the same hierarchical level, such as sentence-level LSTMs, can process their inputs in parallel.

3. *Interpretability of predictions*: The hierarchical attention of HAN can be visualized, facilitating some qualitative insight into what inputs are most important for making predictions at a sentence and word level.

Despite these large advantages, HAN in its normal application still remains limited in its use of structure. In particular, while HAN encodes sentences in a hierarchical way, it does so while using the same LSTM encoder for every sentence. Unfortunately, in doing so it provides no meta information about the role of these sentences in the text or about other meta-information such as the relative positions of these sentences. In this work we introduce a way to overcome these problems by adding sentence type tags encoding the role of a sentences or other information, which is then directly available to the BiLSTM when encoding the sentences. This is illustrated in Figure 2. First the input is segmented into a list of sentences, just as is done also in preprocessing for regular HAN. Then the role of each sentence is added at the beginning and end of each sentence. In our current experiments the roles are restricted to three options: TITLE, ABSTRACT, BODY_TEXT, however, the idea is general enough to include much more specific tags as well as tags encoding relative or absolute sentence position information. We leave exploring more types of tags for future work.

The tag-base approach has the advantage over other possible solutions, such as using different BiLSTMs for different types of sentences, is that it is much more simpler as well as more scalable. Equally important, it allows the BiLSTM to only specialize its functioning to specific types of sentences where needed, while effectively sharing what can be generalized independent of sentence type.

<TITLE>Cross-Task Knowledge-Constrained Self Training </TITLE>

<ABSTRACT> Abstract </ABSTRACT>
<ABSTRACT> We present an algorithmic framework for learning multiple related tasks. </ABSTRACT>
<ABSTRACT> Our framework exploits a form of prior knowledge that relates the output spaces of these tasks. </ABSTRACT>
… <BODY_TEXT> 1 Introduction </BODY_TEXT>
<BODY_TEXT> When two NLP systems are run on the same data, we expect certain constraints to hold between their outputs. </BODY_TEXT>
<BODY_TEXT> This is a form of prior knowledge. </BODY_TEXT>
<BODY_TEXT> We propose a self-training framework that uses such information to significantly boost the performance of one of the systems. </BODY_TEXT>
<BODY_TEXT> The key idea is to perform self-training only on outputs that obey the constraints.</BODY_TEXT>
…

Figure 2: Example of structure-tags for a paper from in the PeerRead computation and language (CL) arXiv dataset. The input text is segmented into sentences and every sentence is tagged with structure-tags at the start and end of it.

4 Accept/Reject prediction on PeerRead

The first scholarly document quality prediction task we test our methods on is accept/reject prediction on arXiv papers from the PeerRead dataset (Kang et al., 2018). This dataset is chosen because of the large amount of earlier work in the literature reporting results on it, allowing us to compare our models against the state-of-the-art on a well studied task.

The full PeerRead dataset holds 14784 papers in total, each of which contains implicit or explicit accept/reject labels. Furthermore, PeerRead contains different subsets of papers. The largest subset consists of arXiv papers (11778) in three computer-science sub-domains[1]: machine learning (cs.LG), computation and language (cs.CL), artificial intelligence (cs.AI), and has only accept/reject labels; this is the dataset that we use. A part of the papers also include reviews (3006 papers) and a subset of the latter that also contains aspect scores (586 papers). However, of these papers with reviews, the large majority is from NIPS (2420 papers), and those papers are all accepted. This, and the fact that the subset of papers with reviews is also really small for deep learning model training purposes explains the fact that most work has focused on the

---

[1]Based on arXiv categories within computer science, see: https://arxiv.org/archive/cs

	training		validation		testing		total
	num	acc:rej	num	acc:rej	num	acc:rej	
machine learning	4543	36.4% : 63.6%	252	36.5% : 63.5%	253	32.0% : 68.0%	5048
computation & language	2374	24.3% : 75.7%	132	22.0% : 78.0%	132	31.1% : 68.9%	2638
artificial intelligence	3682	10.5% : 89.5%	205	8.3% : 91.7%	205	7.8% : 92.2%	4092

Table 1: Data sizes and division between the ratio of accepted and rejected papers for the arXiv subsets

	PeerRead classification	S2ORC regression
optimizer	Adam	
learning rate	0.005	
maximum input characters	20000	
vocabulary size	10000	
weight initialization		
general	Xavier normal	
lstm	Xavier normal	
bias	zero	
loss function	cross entropy	mean absolute error
dropout probability	0.5	0.2
BiLSTM hidden size	256	100
batch size	4	64
embedding size	50	300

Table 2: Hyperparameters used in the experiments.

larger arXiv subset, and the task of accept/reject prediction for the papers in this set.

Table 1 shows the sizes of the different subsets of the arXiv PeerRead dataset, as well as their respective division in number of accept and reject examples. One observation is that for each of the three domains, this division is imbalanced, with the least imbalance for the machine learning subset and the most – an extreme – imbalance for the artificial intelligence subset, in which around 90% of the examples is rejected. These imbalances in the number of examples for each of the classes make learning harder, but can be partly overcome by using strategies such as re-sampling.

4.1 Experimental Setup

In our experiments we tried to stay close to the experimental setup used by (Shen et al., 2019), while deviating from their settings when necessary. Table 2 gives an overview of the used hyperparameters that are shared across experiments, as well as the hyperparameters that are specific to the accept/reject prediction task. We used Adam (Kingma and Ba, 2014) as optimizer, used Xavier (Glorot) uniform and normal weight initialization (Glorot and Bengio, 2010) to initialize general

and lstm weights respectively, and initialized bias weights to zero.[2] We use a considerably larger learning rate of 0.005, compared to 0.0001 used by (Shen et al., 2019).[3]. We use a small batch size of 4. This is necessary for HAN as it uses relatively much memory, because it builds rich hierarchical BiLSTM-based representations directly from the word embeddings. In comparison, the BiLSTM model of (Shen et al., 2019) uses less memory, since it starts out from sentence embeddings implemented as the average word embeddings of sentences. We furthermore use re-sampling on the computational language (cs.CL) subset, as we find that without re-sampling, due to the imbalance in the labels learning fails. The re-sampling is done for each epoch, by keeping the full subset of examples with the less frequent label, but sub-sampling an equal number of random examples from the more frequent label subset. In early exploratory experiments, we also trained models with re-weighing the loss function, with weight inversely proportional to the relative class frequency

---

[2]We also tried Xavier uniform instead of normal initialization for LSTMs, a short investigation suggested that in our setting this gives not much difference for the results.

[3]We found the learning rate 0.0001 used by (Shen et al., 2019b) to give poor results in our experiment.

arXiv sub-domain dataset	Majority class prediction	Benchmark (Kang et al., 2018)	BiLSTM (Shen et al., 2019)	Joint (Shen et al., 2019)	HAN$_{\text{struct-tag}}$ (Our best model)
computation & language	68.9%	75.7%	76.2 ± 1.30%	77.1 ± 3.10%	**81.8 ± 1.91**%
machine learning	68.0%	70.7%	**81.1 ± 0.83**%	79.9 ± 2.54%	78.7 ± 0.69

Table 3: PeerRead accept/reject prediction accuracy: comparison of our best model against state-of-the-art

arXiv sub-domain dataset	metric	Majority class prediction	Average Word Embeddings	HAN	HAN$_{\text{struct-tag}}$
computation & language	accuracy	68.9%	73.7 ± 0.87%	80.3 ± 2.00%	**81.8 ± 1.91**%
	AUC	50%	74.0 ± 1.00%	71.2 ± 2.90 %	**74.5 ± 1.10**%
machine learning	accuracy	67.9%	72.9 ± 0.60%	76.7 ± 2.77 %	**78.7 ± 0.69** %
	AUC	50%	66.2 ± 0.31%	74.3 ± 1.92 %	**75.8 ± 1.49**%

Table 4: PeerRead accept/reject prediction accuracy and *AUC* (area under ROC curve) scores for our models.

[4]. However, we found that this does not fix the problem that the model is not learning beyond predicting always the majority class, whereas re-sampling does. In our experiments the training of all our models proceeds slower than the number of epochs (60) used by Shen et al. (2019) suggests. This observation holds in spite of the fact that we are using a higher learning rate. We therefore used a higher number of 360 training epochs. In each experiment, we used the highest accuracy score on the validation set to select the best model, using the last epoch that achieves that score in case of ties.[5] In addition to accuracy we also report the AUC (area under the ROC curve) scores for the PeerRead dataset.

4.1.1 Input cutoff

Using the full text as input is in theory preferred over using only a selection of it, for the simple reason of not losing information prematurely. In practice however, this is not feasible with high resolution deep learning models such as HANs, with take input that starts at the word level. To save memory and computation, models may instead start out from the sentence level, using embeddings directly as inputs. But this leads to a substantial loss of information from the input, which may ham-

per performance. In spite of this risk, (Shen et al., 2019) apply this strategy in a basic way by computing the average word embedding for each sentence, and using a BiLSTM model on top of that. Nevertheless, they still use a limit on the input length, by allowing only a maximum of 350 sentences. With HAN, which uses more memory and computation-intense sentence-level encodings, a limitation of the input length is even more crucial.

However, rather than limiting the number of sentences, we instead opted for a cutoff on the maximum allowed number of characters, which we set to 20000. We found that with hierarchical attention networks the latter gives better results than using the 360 sentences cutoff, even though on average it corresponds to less words. Looking for an explanation for this counter-intuitive finding, we looked at the distribution over the number of words per example for each of the two length cutoff policies, see Table 5. We found that fixing the number of sentences instead of the number of characters leads to a large variance in the number of words of examples. A likely cause for this is differences in writing styles across authors. In contrast, fixing the number of characters by definition assures that the length of the input, and hence to a lesser extent the number of words (which is proportional to number of characters) is (more) constant. We assume that a more constant number of words also corresponds to a more constant amount of information in the input. We believe that this more constant amount of input information to predict from makes the learning

---

[4] Pytorch supports this directly in the CrossEntropyLoss code.

[5] In a pre-study, we experimented with using either the first or the last best epoch in case there were multiple that tied for the best score. Generally the last best epoch model seemed to work better in case of ties, so we select that.

easier, more so due to the small size of the training set.

4.1.2 Results

Table 3 shows our best results on the PeerRead dataset, using HANs with structure-tags. The same table also shows the previous literature results of (Shen et al., 2019) and (Kang et al., 2018). Observe that in the computation & language domain, we gain 4.7% accuracy over the best of the these literature models (Joint), while on the machine learning domain dataset we lose 2.4% in comparison to the best performing of these literature models on this domain (BiLSTM).

In Table 4 we show the results for both our HAN models as well as for the average word embeddings baseline. These results show a clear improvement from using structure-tags: 1.5% accuracy for the computation & language domain and 2.1% for the machine learning domain.

In summary, the results show: 1) that our HAN models are competitive with the literature results, 2) that structure-tags help to further improve the performance of HAN.

5 Number of citations prediction

The second scholarly document quality prediction task we test our models in is number of citations prediction. A key advantage of this task over the accept/reject prediction task is that much larger datasets can be obtained relatively easily. Obtaining accept/reject labels in large quantities typically requires having an agreement with publishers, and even then because of legal problems, it is hard to obtain and publish such data.[6] Using number of citations as a label solves these problems to a large extent, since it is information that is publicly available and that can be relatively easily obtained from public resources such as Semantic Scholar Database or services like the google scholar API.

While it is nice that number of citations information can be easily obtained, it is reasonable to wonder how useful it is to predict this information. More specifically: is the number of citations of a paper predictive of its quality? Intuitively one would expect this to be the case at least to some extent. Figure 3 shows histograms of the numbers of citations of articles from the PeerRead datasets

for accepted and rejected papers.[7] While there are some differences between the two domains, the main trend is the same in both cases: for rejected papers, the counts are peaked around zero citations and quickly decrease to one or zero for high citation counts. In contrast, the number of citations for accepted papers is two to three times higher on average, depending on the domain. Accepted papers also have a substantial number of occurrences for high numbers of citations. Finally, we formally computed correlation in the form of the Spearman rank-order correlation coefficient (ρ) and associated p-value for both domains. For both domains, the value of ρ is high and the p-value extremely close to zero, which indicates significant correlation can be concluded at all p-levels of significance for a two-sided test. These histograms and numbers prove that there is indeed a strong correlation between acceptance/rejection and the number of citations. Therefore it makes sense to consider the number of citations as an imperfect but nonetheless useful proxy for the quality of scholarly documents.

5.1 A dataset of of document text, number of citations information pairs

Recent works undertake the task of number of citations prediction based on the scholarly document text, but mostly do so while using relatively small datasets. As discussed in related work, some of the recent work adds review text to the input. However, creating models using reviewer comments limits their practical application to after reviewing and reduces the data available for training.

These observations motivated us to rather aim for a relatively large dataset of ⟨paper, number of citations⟩ pairs, which we compile using the S2ORC data (Lo et al., 2020). We selected a subset of papers from the computer science domain from S2ORC, for which title, abstract and body text information is present. We did this for papers in the year range 2000–2010, and counted the number of citations of citing papers that are published within 8 years after the publication of a paper.[8] Randomly ordering the papers, from this we compiled a dataset with in total about 88K papers, and statistics as shown in

---

[6]Note that while the PeerRead arXiv accept/reject dataset is relatively large, its labels are based on heuristics.

[7]To keep counts comparable across papers with different publication years, we restrict the count of citations by other papers to those citing papers published within two years of each paper's publication.

[8]Since exact publication date is not generally available, only citation year, this is somewhat approximate.

	average words per example	median words per example
20000 characters length cutoff	3909 ± 692	4076
360 sentences length cutoff	**5246 ± 1717**	5514

Table 5: The effect of the length cutoff policy on the number of words distribution.

data subset	num examples	avg num words
training	78894	839.1 ± 473.7
validation	4383	849.1 ± 477.5
testing	4382	856.4 ± 489.0

Table 6: S2ORC dataset size statistics.

Table 6. Note that to the best of our knowledge, the largest number of articles used for citation prediction in earlier work is described in (Plank and van Dale, 2019), we use more than 23 times the number of articles used in their experiments.[9]

While we kept the maximum number of words per example at 20000, the average number of words lies around 840 words per example which is much lower, since the number of words provided in the body_text fields of S2ORC is still limited in practice. We leave creating examples with the full paper text for future work. The the thus created examples consists of the combined title, abstract and body_text. The labels added to these examples consist not exactly of the number of citations but rather a derivative function of this number, as explained in the next subsection.

5.2 The logarithm of number of citations as a proxy for quality

The number of citations follow of scholarly documents follow a Zipfian distribution (Silagadze, 1997). That is, most papers have little citations, but those that obtain more citations tend to get exponentially more. To account for this, we used the log of the number of citations to create a metric that aims to approximates a measure of quality on a linear scale. In practice, we use the function:

$$\text{citation\_score} = log_e(n + 1) \qquad (1)$$

adding one to the number of citations n before taking the log, to make sure the function is well-defined even for papers that have zero citations.

5.2.1 Comparison to alternative citation scores

What alternatives to our log-based metric have been explored in the literature? Li et al. (2019) map citation counts to the [0,1] range, presumably by simply scaling them after the paper with the maximum and minimum number of citations in a dataset have been determined. But this approach transfers poorly to new data, since the number of citations follows the Zipfian distribution, there is a large chance of encountering an even higher number of citations in unseen data. Furthermore, because of the Zipfian nature of the number of citations, this transformation will map the citation score of many papers to a number close to zero, thereby drastically inflating the evaluation scores of predictions for this citation score. A better alternative approach is to discretize the number of citations into a fixed number of ranges. In order to predict the impact of scientific papers, Plank and van Dale (2019) discretize time-normalized citation statistics into low, medium and high impact papers based on a boxplot and outlier analysis. In comparison however, our approach does not require discretization/binning, which has advantages: 1) It does not commit to a fixed resolution, 2) it avoids problems for papers with a number of citations on the border of two bins, 3) it allows the predicted scores to be deterministically transformed back into an actual number of citations.

5.3 Loss function and evaluation metrics

Having motivated our chosen citation score (1), the next important question is what loss function we should optimize when training our networks to predict this score. Whereas *mean-squared-error* is the default choice for regression problems, we found this loss function to perform poorly in combination with our score. In contrast, preliminary experiments showed that *mean absolute error* facilitates effective and relatively stable optimization, so we decided to use this as our loss function in the rest of our experiments.

In addition to the choice of loss function, another important question is what quality metrics we

---

[9]We plan to make our dataset available upon acceptance of this article.
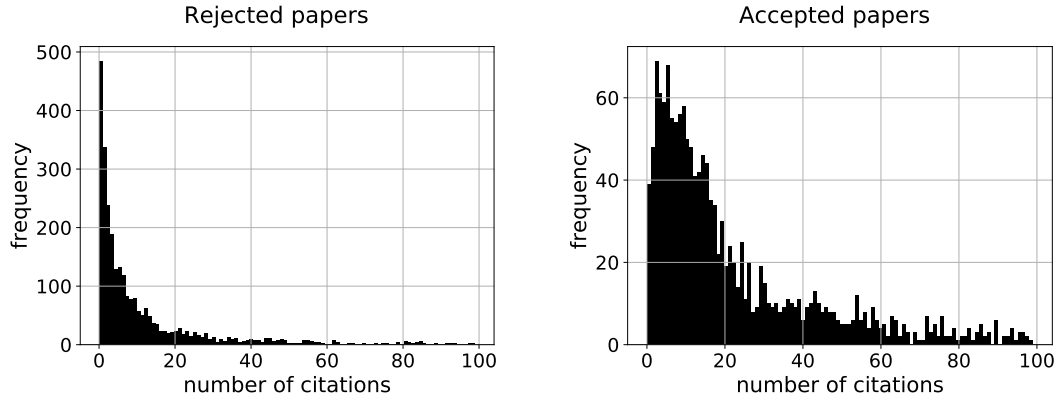
(a) Computation and Language domain.
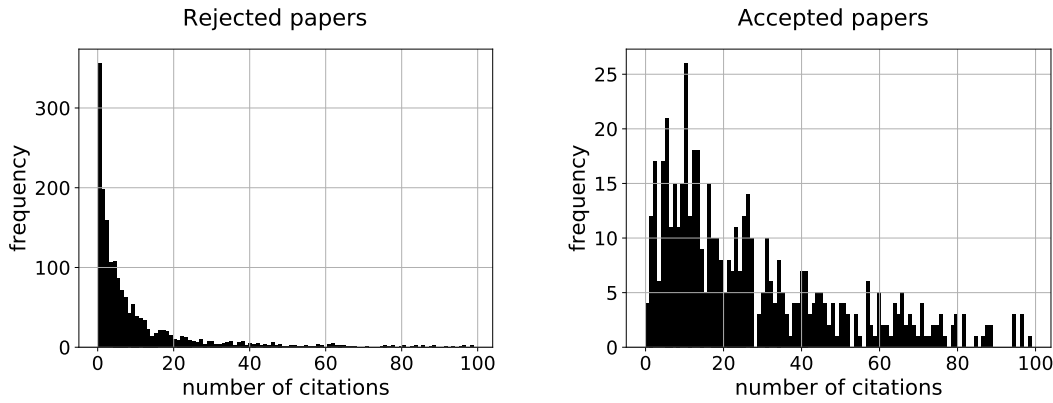
(b) Machine Learning domain

Domain	Average number of citations		Spearman rank-order correlation coefficient (ρ), p-value
	rejected articles	accepted articles	
Computation and Language	14.8 ± 44.3	59.0 ± 105.9	$0.466, 1.6 \times 10^{-128}$
Machine Learning	24.0 ± 127.3	61.0 ± 232.6	$0.375, 5 \times 10^{-153}$

(c) Global statistics and formal correlation measure: average number of citations for rejected and accepted papers and Spearman rank-order correlation in the different domains.

Figure 3: Histograms and global statistics of number of citations for accepted and rejected papers for the sub-domains of PeerRead. Histograms are truncated on the right at 100 citations.

are interested in? Mean-squared-error and mean-absolute-error are standard metrics for regression evaluation, so we report those. However, in addition, we report the R2 score, which denotes the proportion of the variance in the dependent variable that is predictable from the independent variable(s).

The R^2 function is defined as:

$$R^2 = 1 - \text{FVU} = 1 - \frac{MSE(Y, Y')}{var[Y]} \quad (2)$$

With Y' and Y being the predicted and actual labels respectively, MSE being the mean-squared-error and and FVU the fraction of variance unexplained. This explains how the R^2 score normalizes for the relative difficulty for the task, by normalizing by

the variance of the labels in the test set. Another interpretation is that the R^2 score normalizes by the error obtained by always predicting the average of the test labels. Consequently, a R^2 score larger than 0 means performance better than this "average prediction baseline", and below 0 means worse than this baseline. This avoids the need to add scores for this dummy baseline for comparison, making the R^2 score more directly interpretable than mean-squared-error or mean-absolute-error. As such, use of the R^2 score also becomes particularly useful for assuring comparability of model scores across datasets, which will typically differ in test set variance. For these reasons, the R^2 score

	HAN	HAN$_{\text{struct-tag}}$
R^2 score	0.275 ± 0.008	$\mathbf{0.285 \pm 0.002}$
mean squared error	1.201 ± 0.007	$\mathbf{1.184 \pm 0.002}$
mean absolute error	0.833 ± 0.003	$\mathbf{0.831 \pm 0.001}$

Table 7: Test scores for the log number of citations prediction on the S2ORC dataset.

is our preferred metric when comparing the performance of models.

5.4 Number of citations prediction results

Table 7 shows the results of our models trained on our new S2ORC number of citations prediction dataset. We observe that the HAN$_{\text{struct-tag}}$ model outperforms HAN. (Shen et al., 2019).

6 Conclusion

This work showed the usefulness of HAN and rich context tags to the processing of scientific documents, which are significantly longer than the text inputs of usual NLP problems. Substantial improvements in prediction quality were obtained for both accept/reject estimation and number of citations prediction. A strong and significant correlation between accept/reject labels and number of citations was demonstrated, signaling the usefulness of the latter as a measure of scholarly document quality. We derived a new citation prediction dataset from the S2ORC data, more than 23 times larger than alternatives used before in the literature. Our approach demonstrates the feasibility of automatically generating large datasets for number of citations prediction from open resources. This opens new paths for the application of more advanced deep learning models to scholarly document quality prediction, models that are more accurate but also require more data to train effectively.

References

Ali Abrishami and Sadegh Aliakbary. 2019. Predicting citation counts based on deep neural network learning techniques. *Journal of Informetrics*, 13:485–499.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate.

Tim Brody, Stevan Harnad, and Les Carr. 2006. Earlier web usage statistics as predictors of later citation impact. *Journal of the American Association for Information Science and Technology (JASIST)*, 57(8):1060–1072. DOI: 10.1002/asi.20373.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Lawrence Fu and Constantin Aliferis. 2008. Models for predicting and explaining citation count of biomedical articles. *AMIA ... Annual Symposium proceedings / AMIA Symposium. AMIA Symposium*, 6:222–6.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, volume 9 of *JMLR Proceedings*, pages 249–256. JMLR.org.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep residual learning for image recognition. *CoRR*, abs/1512.03385.

Salah El Hihi and Yoshua Bengio. 1996. Hierarchical recurrent neural networks for long-term dependencies. In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, editors, *Advances in Neural Information Processing Systems 8*, pages 493–499. MIT Press.

Sepp Hochreiter. 1998. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(2):107–116.

Alfonso Ibáñez, Pedro Larrañaga, and Concha Bielza. 2009. Predicting citation count of Bioinformatics papers within four years of publication. *Bioinformatics*, 25(24):3303–3309.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda B. Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's multilingual neural machine translation system: Enabling zero-shot translation. *CoRR*, abs/1611.04558.

Dongyeop Kang, Waleed Ammar, Bhavana Dalvi, Madeleine van Zuylen, Sebastian Kohlmeier, Eduard Hovy, and Roy Schwartz. 2018. A dataset of peer reviews (peerread): Collection, insights and nlp applications.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. Cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.

Siqing Li, Wayne Xin Zhao, Eddy Jing Yin, and Ji-Rong Wen. 2019. A neural citation count prediction model based on peer review text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4914–4924, Hong Kong, China. Association for Computational Linguistics.

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel S. Weld. 2020. S2orc: The semantic scholar open research corpus. In *Proceedings of ACL*.

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *ICML (3)*, volume 28 of *JMLR Workshop and Conference Proceedings*, pages 1310–1318. JMLR.org.

Barbara Plank and Reinard van Dale. 2019. Cite-tracked: A longitudinal dataset ofpeer reviews and citations. In *Proceedings of the 4th Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL 2019)*.

Feng Qiao, Lizhen Xu, and Xiaowei Han. 2018. Modularized and attention-based recurrent convolutional neural network for automatic academic paper aspect scoring. In *Web Information Systems and Applications*, pages 68–76, Cham. Springer International Publishing.

Adithya Rao and Nemanja Spasojevic. 2016. Actionable and political text classification using word embeddings and lstm.

Tim RocktÃd'schel, Edward Grefenstette, Karl Moritz Hermann, TomÃąÅą KoÄ■iskÃ¡, and Phil Blunsom. 2015. Reasoning about entailment with neural attention.

Rico Sennrich and Barry Haddow. 2016. Linguistic input features improve neural machine translation. *CoRR*, abs/1606.02892.

Aili Shen, Jianzhong Qi, and Timothy Baldwin. 2017. A hybrid model for quality assessment of wikipedia articles. In *Proceedings of the Australasian Language Technology Association Workshop 2017*, pages 43–52.

Ali Shen, Bahar Salehi, Timothy Baldwin, and Jianzhong Qi. 2019. A joint model for multimodal document quality assessment. In *JCDL '19: Proceedings of the 18th Joint Conference on Digital Libraries*, pages 107–110.

Z. K. Silagadze. 1997. Citations and the zipf-mandelbrot's law. *Complex Systems*, 11(6):487–499.

Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015. Highway networks. *CoRR*, abs/1505.00387.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, San Diego, California. Association for Computational Linguistics.