

Rationale-Augmented Convolutional Neural Networks for Text Classification

Ye Zhang,¹ Iain Marshall,² Byron C. Wallace³

¹Department of Computer Science, University of Texas at Austin

²Department of Primary Care and Public Health Sciences, Kings College London

³College of Computer and Information Science, Northeastern University

yezhang@cs.utexas.edu, iain.marshall@kcl.ac.uk

byron@ccs.neu.edu

Abstract

We present a new Convolutional Neural Network (CNN) model for text classification that jointly exploits labels on documents and their constituent sentences. Specifically, we consider scenarios in which annotators explicitly mark sentences (or snippets) that support their overall document categorization, i.e., they provide *rationales*. Our model exploits such supervision via a hierarchical approach in which each document is represented by a linear combination of the vector representations of its component sentences. We propose a sentence-level convolutional model that estimates the probability that a given sentence is a rationale, and we then scale the contribution of each sentence to the aggregate document representation in proportion to these estimates. Experiments on five classification datasets that have document labels and associated rationales demonstrate that our approach consistently outperforms strong baselines. Moreover, our model naturally provides explanations for its predictions.

1 Introduction

Neural models that exploit word embeddings have recently achieved impressive results on text classification tasks (Goldberg, 2015). Feed-forward Convolutional Neural Networks (CNNs), in particular, have emerged as a relatively simple yet powerful class of models for text classification (Kim, 2014).

These neural text classification models have tended to assume a standard supervised learning setting in which instance labels are provided. Here we consider an alternative scenario in which we assume

that we are provided a set of *rationales* (Zaidan et al., 2007; Zaidan and Eisner, 2008; McDonnell et al., 2016) in addition to instance labels, i.e., sentences or snippets that support the corresponding document categorizations. Providing such rationales during manual classification is a natural interaction for annotators, and requires little additional effort (Settles, 2011; McDonnell et al., 2016). Therefore, when training new classification systems, it is natural to acquire supervision at both the document and sentence level, with the aim of inducing a better predictive model, potentially with less effort.

Learning algorithms must be designed to capitalize on these two types of supervision. Past work (Section 2) has introduced such methods, but these have relied on linear models such as Support Vector Machines (SVMs) (Joachims, 1998), operating over sparse representations of text. We propose a novel CNN model for text classification that exploits both document labels and associated rationales.

Specific contributions of this work as follows. (1) This is the first work to incorporate rationales into neural models for text classification. (2) Empirically, we show that the proposed model uniformly outperforms relevant baseline approaches across five datasets, including previously proposed models that capitalize on rationales (Zaidan et al., 2007; Marshall et al., 2016) and multiple baseline CNN variants, including a CNN equipped with an attention mechanism. We also report state-of-the-art results on the important task of automatically assessing the risks of bias in the studies described in full-text biomedical articles (Marshall et al., 2016). (3) Our model naturally provides *explanations* for its predic-

tions, providing interpretability.

We have made available online both a Theano¹ and a Keras implementation² of our model.

2 Related Work

2.1 Neural models for text classification

Kim (2014) proposed the basic CNN model we describe below and then build upon in this work. Properties of this model were explored empirically in (Zhang and Wallace, 2015). We also note that Zhang et al. (2016) extended this model to jointly accommodate multiple sets of pre-trained word embeddings. Roughly concurrently to Kim, Johnson and Zhang (2014) proposed a similar CNN architecture, although they swapped in one-hot vectors in place of (pre-trained) word embeddings. They later developed a semi-supervised variant of this approach (Johnson and Zhang, 2015).

In related recent work on Recurrent Neural Network (RNN) models for text, Tang et al. (2015) proposed using a Long Short Term Memory (LSTM) layer to represent each sentence and then passing another RNN variant over these. And Yang et al. (2016) proposed a hierarchical network with two levels of attention mechanisms for document classification. We discuss this model specifically as well as attention more generally and its relationship to our proposed approach in Section 4.3.

2.2 Exploiting *rationales*

In long documents the importance of sentences varies; some are more central than others. Prior work has investigated methods to measure the relative importance sentences (Ko et al., 2002; Murata et al., 2000). In this work we adopt a particular view of sentence importance in the context of document classification. In particular, we assume that documents comprise sentences that directly support their categorization. We call such sentences *rationales*.

The notion of rationales was first introduced by Zaidan et al. (2007). To harness these for classification, they proposed modifying the Support Vector Machine (SVM) objective function to encode a preference for parameter values that result in instances containing manually annotated rationales

¹<https://github.com/yezhang-xiaofan/Rationale-CNN>

²<https://github.com/bwallace/rationale-CNN>

being more confidently classified than ‘pseudo’-instances from which these rationales had been stripped. This approach dramatically outperformed baseline SVM variants that do not exploit such rationales. Yessenalina et al. (2010) later developed an approach to *generate* rationales.

Another line of related work concerns models that capitalize on *dual supervision*, i.e., labels on individual *features*. This work has largely involved inserting constraints into the learning process that favor parameter values that align with *a priori* feature-label affinities or rankings (Druck et al., 2008; Mann and McCallum, 2010; Small et al., 2011; Settles, 2011). We do not discuss this line of work further here, as our focus is on exploiting provided rationales, rather than individual labeled features.

3 Preliminaries: CNNs for text classification

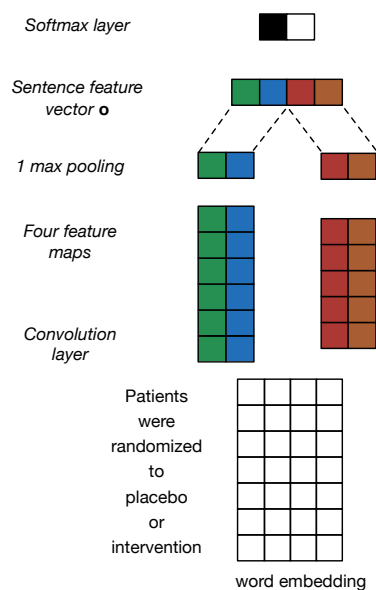


Figure 1: A toy example of a CNN for sentence classification. Here there are four filters, two with heights 2 and two with heights 3, resulting in feature maps with lengths 6 and 5 respectively.

We first review the simple one-layer CNN for sentence modeling proposed by Kim (2014). Given a sentence or document comprising n words w_1, w_2, \dots, w_n , we replace each word with its d -dimensional pretrained embedding, and stack them row-wise, generating an instance matrix $A \in \mathbb{R}^{n \times d}$.

We then apply convolution operations on this matrix using multiple linear filters, these will have the same width d but may vary in *height*. Each filter thus effectively considers distinct n -gram features, where n corresponds to the filter height. In practice, we introduce multiple, redundant features of each height; thus each filter height might have hundreds of corresponding instantiated filters. Applying filter i parameterized by $W_i \in \mathbb{R}^{h_i \times d}$ to the instance matrix induces a *feature map* $f_i \in \mathbb{R}^{n-h_i+1}$. This process is performed by sliding the filter from the top of the matrix (the start of the document or sentence) to the bottom. At each location, we apply element-wise multiplication between filter i and sub-matrix $A[j : j + h_i - 1]$, and then sum up the resultant matrix elements. In this way, we induce a vector (feature map) for each filter.

We next run the feature map through an element-wise non-linear transformation. Specifically, we use the *Rectified Linear Unit*, or ReLU (Krizhevsky et al., 2012). We extract the maximum value o_i from each feature map i (*l-max pooling*).

Finally, we concatenate all of the features o_i to form a vector representation $\mathbf{o} \in \mathbb{R}^{|F|}$ for this instance, where $|F|$ denotes the total number of filters. Classification is then performed on top of \mathbf{o} , via a softmax function. Dropout (Srivastava et al., 2014) is often applied at this layer as a means of regularization. We provide an illustrative schematic of the basic CNN architecture just described in Figure 1. For more details, see (Zhang and Wallace, 2015).

This model was originally proposed for sentence classification (Kim, 2014), but we can adapt it for document classification by simply treating the document as one long sentence. We will refer to this basic CNN variant as *CNN* in the rest of the paper. Below we consider extensions that account for document structure.

4 Rationale-Augmented CNN for Document Classification

We now move to the main contribution of this work: a rationale-augmented CNN for text classification. We first introduce a simple variant of the above CNN that models document structure (Section 4.1) and then introduce a means of incorporating rationale-level supervision into this model (Section

4.2). In Section 4.3 we discuss connections to attention mechanisms and describe a baseline equipped with one, inspired by Yang et al. (2016).

4.1 Modeling Document Structure

Recall that rationales are snippets of text marked as having supported document-level categorizations. We aim to develop a model that can exploit these annotations during training to improve classification. Here we achieve this by developing a hierarchical model that estimates the probabilities of individual sentences being rationales and uses these estimates to inform the document level classification.

As a first step, we extend the CNN model above to explicitly account for document structure. Specifically, we apply a CNN to each individual sentence in a document to obtain sentence vectors independently. We then sum the respective sentence vectors to create a document vector.³ As before, we add a softmax layer on top of the document-level vector to perform classification. We perform regularization by applying dropout both on the individual sentence vectors and the final document vector. We will refer to this model as *Doc-CNN*. Doc-CNN forms the basis for our novel approach, described below.

4.2 RA-CNN

In this section we present the Rationale-Augmented CNN (*RA-CNN*). Briefly, RA-CNN induces a document-level vector representation by taking a *weighted* sum of its constituent sentence vectors. Each sentence weight is set to reflect the estimated probability that it is a rationale in support of the most likely class. We provide a schematic of this model in Figure 2.

RA-CNN capitalizes on both sentence- and document-level supervision. There are thus two steps in the training phase: *sentence level* training and *document level* training. For the former, we apply a CNN to each sentence j in document i to obtain sentence vectors $\mathbf{x}_{\text{sen}}^{ij}$. We then add a softmax layer parametrized by \mathbf{W}_{sen} ; this takes as input sentence vectors. We fit this model to maximize the probabilities of the observed rationales:

³We also experimented with taking the average of sentence vectors, but summing performed better in informal testing.

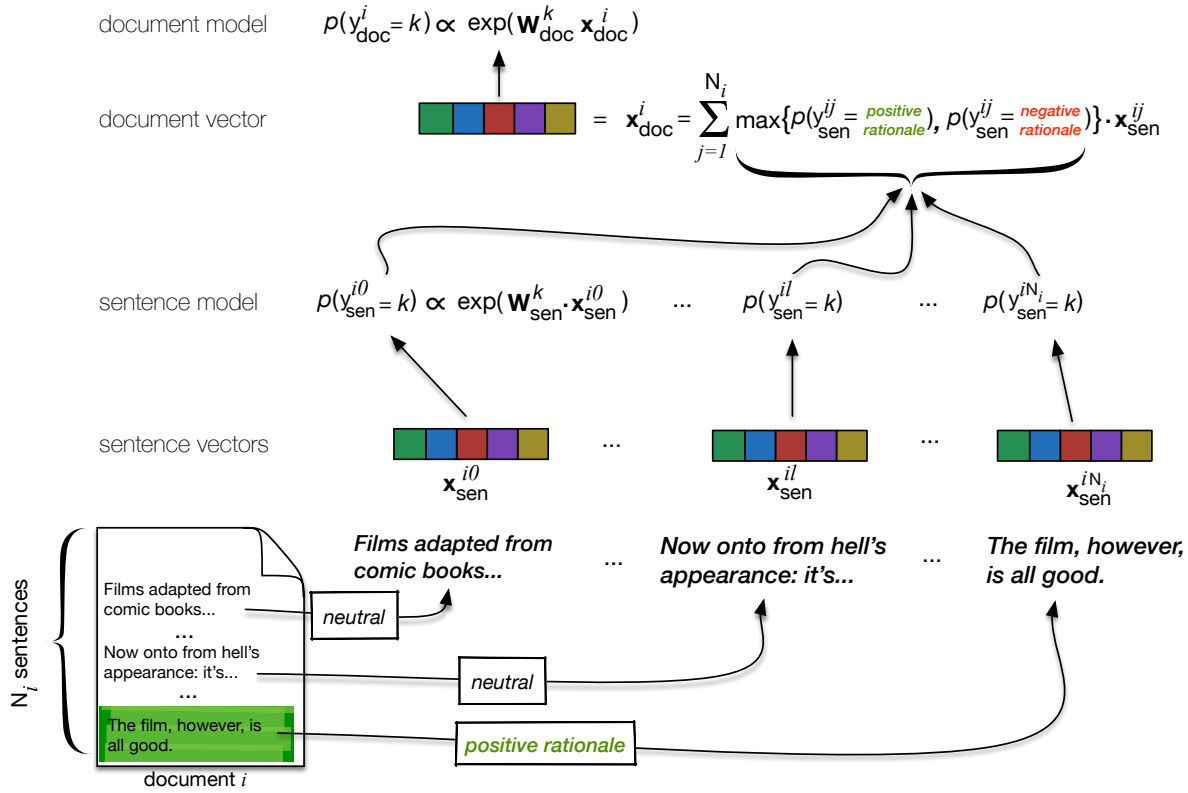


Figure 2: A schematic of our proposed Rationale-Augmented Convolution Neural Network (RA-CNN). The sentences comprising a text are passed through a sentence model that outputs probabilities encoding the likelihood that sentences are neutral or a (positive or negative) rationale. Sentences likely to be rationales are given higher weights in the global document vector, which is the input to the document model.

$$p(y_{\text{sen}}^{ij} = k; \mathbf{E}, \mathbf{C}, \mathbf{W}_{\text{sen}}) = \frac{\exp(\mathbf{W}_{\text{sen}}^{(k)T} \mathbf{x}_{\text{sen}}^{ij})}{\sum_{k=1}^{K_{\text{sen}}} \exp(\mathbf{W}_{\text{sen}}^{(k)T} \mathbf{x}_{\text{sen}}^{ij})} \quad (1)$$

Where y_{sen}^{ij} denotes the rationale label for sentence j in document i , K_{sen} denotes the number of possible classes for sentences, \mathbf{E} denotes the word embedding matrix, \mathbf{C} denotes the convolution layer parameters, and \mathbf{W}_{sen} is a matrix of weights (comprising one weight vector per sentence class).

In our setting, each sentence has three possible labels ($K_{\text{sen}} = 3$). When a rationale sentence appears in a positive document,⁴ it is a *positive rationale*; when a rationale sentence appears in a negative document, it is a *negative rationale*. All other sen-

⁴All of the document classification tasks we consider here are binary, although extension of our model to multi-class scenarios is straight-forward.

tences belong to a third, neutral class: these are *non-rationales*. We also experimented with having only two sentence classes: *rationales* and *non-rationales*, but this did not perform as well as explicitly maintaining separate classes for rationales of different polarities.

We train an estimator using the provided rationale annotations, optimizing over $\{\mathbf{E}, \mathbf{C}, \mathbf{W}_{\text{sen}}\}$ to minimize the categorical cross-entropy of sentence labels. Once trained, this sub-model can provide conditional probability estimates regarding whether a given sentence is a positive or a negative rationale, which we will denote by p_{pos} and p_{neg} , respectively.

We next train the document-level classification model. The inputs to this are vector representations of documents, induced by summing over constituent sentence vectors, as in Doc-CNN. However, in the RA-CNN model this is a weighted sum. Specifically, weights are set to the estimated probabilities

that corresponding sentences are rationales in the most likely direction. More precisely:

$$\mathbf{x}_{\text{doc}}^i = \sum_{j=1}^{N_i} \mathbf{x}_{\text{sen}}^{ij} \cdot \max\{p_{\text{pos}}^{ij}, p_{\text{neg}}^{ij}\} \quad (2)$$

Where N_i is the number of sentences in the i th document. The intuition is that sentences likely to be rationales will have greater influence on the resultant document vector representation, while the contribution of neutral sentences (which are less relevant to the classification task) will be minimized.

The final classification is performed by a softmax layer parameterized by \mathbf{W}_{doc} ; the inputs to this layer are the document vectors. The \mathbf{W}_{doc} parameters are trained using the document-level labels, y_{doc}^i :

$$p(y_{\text{doc}}^i = k; \mathbf{E}, \mathbf{C}, \mathbf{W}_{\text{doc}}) = \frac{\exp(\mathbf{W}_{\text{doc}}^{(k)T} \mathbf{x}_{\text{doc}}^i)}{\sum_{k=1}^{K_{\text{doc}}} \exp(\mathbf{W}_{\text{doc}}^{(k)T} \mathbf{x}_{\text{doc}}^i)} \quad (3)$$

where K_{doc} is the cardinality of the document label set. We optimize over parameters to minimize cross-entropy loss (w.r.t. the document labels).

We note that the sentence- and document-level models share word embeddings \mathbf{E} and convolution layer parameters \mathbf{C} , but the document-level model has its own softmax parameters \mathbf{W}_{doc} . When training the document-level model, \mathbf{E} , \mathbf{C} and \mathbf{W}_{doc} are fit, but we hold \mathbf{W}_{sen} fixed.

The above two-step strategy can be equivalently described as follows. We first estimate \mathbf{E} , \mathbf{C} and \mathbf{W}_{sen} , which parameterize our model for identifying rationales in documents. We then move to fitting our document classification model. For this we initialize the word embedding and convolution parameters to the \mathbf{E} and \mathbf{C} estimates from the preceding step. We then directly minimize the document level classification objective, tuning \mathbf{E} and \mathbf{C} and simultaneously fitting \mathbf{W}_{doc} .

Note that this sequential training strategy differs from the *alternating* training approach commonly used in multi-task learning (Collobert and Weston, 2008). We found that the latter approach does not work well here, leading us to instead adopt the cascade-like feature learning approach (Collobert and Weston, 2008) just described.

One nice property of our model is that it naturally provides explanations for its predictions: the model identifies rationales and then categorizes documents informed by these. Thus if the model classifies a test instance as *positive*, then by construction the sentences associated with the highest p_{pos}^{ij} estimates are those that the model relied on most in coming to this disposition. These sentences can of course be output in conjunction with the prediction. We provide concrete examples of this in Section 7.2.

4.3 Rationales as ‘Supervised Attention’

One may view RA-CNN as a supervised variant of a model equipped with an *attention mechanism* (Bahdanau et al., 2014). On this view, it is apparent that rather than capitalizing on rationales directly, we could attempt to let the model learn which sentences are important, using *only* the document labels. We therefore construct an additional baseline that does just this, thereby allowing us to assess the impact of learning directly from rationale-level supervision.

Following the recent work of Yang et al. (2016), we first posit for each sentence vector a hidden representation $\mathbf{u}_{\text{sen}}^{ij}$. We then define a sentence-level *context vector* \mathbf{u}_s , which we multiply with each $\mathbf{u}_{\text{sen}}^{ij}$ to induce a weight α_{ij} . Finally, the document vector is taken as a weighted sum over sentence vectors, where weights reflect α ’s. We have:

$$\mathbf{u}_{\text{sen}}^{ij} = \tanh(\mathbf{W}_s \mathbf{x}_{\text{sen}}^{ij} + b_s) \quad (4)$$

$$\alpha_{ij} = \frac{\exp(\mathbf{u}_s^T \mathbf{u}_{\text{sen}}^{ij})}{\sum_j^{N_i} \exp(\mathbf{u}_s^T \mathbf{u}_{\text{sen}}^{ij})} \quad (5)$$

$$\mathbf{x}_{\text{doc}}^i = \sum_j^{N_i} \alpha_{ij} \mathbf{x}_{\text{sen}}^{ij} \quad (6)$$

where $\mathbf{x}_{\text{doc}}^i$ again denotes the document vector fed into a softmax layer, and \mathbf{W}_s , \mathbf{u}_s and b_s are learned during training. We will refer to this attention-based method as *AT-CNN*.

5 Datasets

We used five text classification datasets to evaluate our approach in total. Four of these are biomedical text classification datasets (5.1) and the last is a collection of movie reviews (5.2). These datasets share the property of having recorded rationales associated

with each document categorization. We summarize attributes of all datasets used in this work in Table 1.

5.1 Risk of Bias (RoB) Datasets

We used a collection *Risk of Bias* (RoB) text classification datasets, described at length elsewhere (Marshall et al., 2016). Briefly, the task concerns assessing the reliability of the evidence presented in full-text biomedical journal articles that describe the conduct and results of randomized controlled trials (RCTs). This involves, e.g., assessing whether or not patients were properly blinded as to whether they were receiving an active treatment or a comparator (such as a placebo). If such blinding is not done correctly, it compromises the study by introducing statistical bias into the treatment efficacy estimate(s) derived from the trial.

A formal system for making bias assessments is codified by the Cochrane Risk of Bias Tool (Higgins et al., 2011). This tool defines multiple *domains*; the risk of bias may be assessed in each of these. We consider four domains here. (1) Random sequence generation (RSG): were patients were assigned to treatments in a truly random fashion? (2) Allocation concealment (AC): were group assignments revealed to the person assigning patients to groups (so that she may have knowingly or unknowingly) influenced these assignments? (3) Blinding of Participants and Personnel (BPP): were all trial participants and individuals involved in running the trial blinded as to who was receiving which treatment? (4) Blinding of outcome assessment (BOA): were the parties who measured the outcome(s) of interest blinded to the intervention group assignments? These assessments are somewhat subjective. To increase transparency, researchers performing RoB assessment therefore record rationales (sentences from articles) supporting their assessments.

5.2 Movie Review Dataset

We also ran experiments on a movie review (MR) dataset with accompanying rationales. Pang and Lee (2004) developed and published the original version of this dataset, which comprises 1000 positive and 1000 negative movie reviews from the Internet Movie Database (IMDB).⁵ Zaidan et al. (2007) then

⁵<http://www.imdb.com/>

	<i>N</i>	<i>#sen</i>	<i>#token</i>	<i>#rat</i>
RSG	8399	300	9.92	0.31
AC	11512	297	9.87	0.15
BPP	7997	296	9.95	0.21
BOA	2706	309	9.92	0.2
MR	1800	32.6	21.2	8.0

Table 1: Dataset characteristics. *N* is the number of instances, *#sen* is the average sentence count, *#token* is the average token per-sentence count and *#rat* is the average number of rationales per document.

augmented this dataset by adding rationales corresponding to the binary classifications for 1800 documents, leaving the remaining 200 for testing. Because 200 documents is a modest test sample size, we ran 9-fold cross validation on the 1800 annotated documents (each fold comprising 200 documents). The rationales, as originally marked in this dataset, were sub-sentential snippets; for the purposes of our model, we considered the entire sentences containing the marked snippets as rationales.

6 Experimental Setup

6.1 Baselines

We compare against several baselines to assess the advantages of directly incorporating rationale-level supervision into the proposed CNN architecture. We describe these below.

SVMs. We evaluated a few variants of linear Support Vector Machines (SVMs). These rely on sparse representations of text. We consider variants that exploit uni- and bi-grams; we refer to these as *uni-SVM* and *bi-SVM*, respectively. We also re-implemented the rationale augmented SVM (*RA-SVM*) proposed by Zaidan et al. (2007), described in Section 2.

For the RoB dataset, we also compare to a recently proposed multi-task SVM (MT-SVM) model developed specifically for these RoB datasets (Marshall et al., 2015; Marshall et al., 2016). This model exploits the intuition that the risks of bias across the domains codified in the aforementioned Cochrane RoB tool will likely be correlated. That is, if we know that a study exhibits a high risk of bias for one domain, then it seems reasonable to assume it is at an elevated risk for the remaining domains. Furthermore, Marshall et al. (2016) include rationale-level supervision by first training a (multi-task) *sentence-level* model to identify sentences likely to support

RoB assessments in the respective domains. Special features extracted from these predicted rationales are then activated in the *document-level* model, informing the final classification. This model is the state-of-the-art on this task.

CNNs. We compare against several baseline CNN variants to demonstrate the advantages of our approach. We emphasize that our focus in this work is **not** to explore how to induce generally ‘better’ document vector representations – this question has been addressed at length elsewhere, e.g., (Le and Mikolov, 2014; Jozefowicz et al., 2015; Tang et al., 2015; Yang et al., 2016).

Rather, the main contribution here is an augmentation of CNNs for text classification to capitalize on rationale-level supervision, thus improving performance and enhancing interpretability. This informed our choice of baseline CNN variants: standard CNN (Kim, 2014), Doc-CNN (described above) and AT-CNN (also described above) that capitalizes on an (unsupervised) attention mechanism at the sentence level, described in Section 4.3.⁶

6.2 Implementation/Hyper-Parameter Details

Sentence splitting. To split the documents from all datasets into sentences for consumption by our Doc-CNN and RA-CNN models, we used the Natural Language Toolkit (NLTK)⁷ sentence splitter.

SVM-based models. We kept the 50,000 most frequently occurring features in each dataset. For estimation we used SGD. We tuned the C hyperparameter using nested development sets. For the RA-SVM, we additionally tuned the μ and $C_{contrast}$ parameters, as per Zaidan et al. (2007).

CNN-based models. For all models and datasets we initialized word embeddings to pre-trained vectors fit via Word2Vec. For the movie reviews dataset these were 300-dimensional and trained on Google News.⁸ For the RoB datasets, these were 200-dimensional and trained on biomedical texts in PubMed/PubMed Central (Pyysalo et al., 2013).⁹

⁶We also experimented briefly with LSTM and GRU (Gated Recurrent Unit) models, but found that simple CNN performed better than these. Moreover, CNNs are relatively robust and less sensitive to hyper-parameter selection.

⁷<http://www.nltk.org/api/nltk.tokenize.html>

⁸<https://code.google.com/archive/p/word2vec/>

⁹<http://bio.nlplab.org/>

Training proceeded as follows. We first extracted all sentences from all documents in the training data. The distribution of sentence types is highly imbalanced (nearly all are neutral). Therefore, we downsampled sentences before each epoch, so that sentence classes were equally represented. After training on sentence-level supervision, we moved to document-level model fitting. For this we initialized embedding and convolution layer parameters to the estimates from the preceding sentence-level training step (though these were further tuned to optimize the document-level objective).

For RA-CNN, we tuned the dropout rate (range: 0-.9) applied at the sentence vector level on each training fold (using a subset of the training data as a validation set) during the document level training phase. Anecdotally, we found this has a greater effect than the other model hyperparameters, which we thus set after a small informal process of experimentation on a subset of the data. Specifically, we fixed the dropout rate at the document level to 0.5, and we used 3 different filter heights: 3, 4 and 5, following (Zhang and Wallace, 2015). For each filter height, we used 100 feature maps for the baseline CNN, and 20 for all the other CNN variants.

For parameter estimation we used ADADELTA (Zeiler, 2012), mini-batches of size 50, and an early stopping strategy (using a validation set).

7 Results and Discussion

7.1 Quantitative Results

For all CNN models, we replicated experiments 5 times, where each replication constituted 5-fold and 9-fold CV respectively the RoB and the movies datasets, respectively. We report the mean and observed ranges in accuracy across these 5 replications for these models, because attributes of the model (notably, dropout) and the estimation procedure render model fitting stochastic (Zhang and Wallace, 2015). We do not report ranges for SVM-based models because the variance inherent in the estimation procedure is much lower for these simpler, linear models.

Results on the RoB datasets and the movies dataset are shown in Tables 2 and Table 3, respectively. RA-CNN consistently outperforms all of the baseline models, across all five datasets. We also

Method	RSG	AC	BPP	BOA
<i>Uni-SVM</i>	72.16	72.81	72.80	65.85
<i>Bi-SVM</i>	74.82	73.62	75.13	67.29
<i>RA-SVM</i>	72.54	74.11	75.15	66.29
<i>MT-SVM</i>	76.15	74.03	76.33	67.50
<i>CNN</i>	72.50 (72.22, 72.65)	72.16 (71.49, 72.93)	75.03 (74.16, 75.44)	63.76 (63.12, 64.15)
<i>Doc-CNN</i>	72.60 (72.43, 72.90)	72.92 (72.19, 73.48)	74.24 (74.03, 74.38)	63.64 (63.23, 64.37)
<i>AT-CNN</i>	74.14 (73.40, 74.58)	73.66 (73.12, 73.92)	74.29 (74.09, 74.74)	63.34 (63.21, 63.49)
<i>RA-CNN</i>	77.42 (77.33, 77.59)	76.14 (75.89, 76.29)	76.47 (76.15, 76.75)	69.67 (69.33, 69.93)
<i>Human</i>	85.00	80.00	78.10	83.20

Table 2: Accuracies on the four RoB datasets. Uni-SVM: unigram SVM, Bi-SVM: Bigram SVM, RA-SVM: Rationale-augmented SVM (Zaidan et al., 2007), MT-SVM: a multi-task SVM model specifically designed for the RoB task, which also exploits the available sentence supervision (Marshall et al., 2016). We also report an estimate of human-level performance, as calculated using subsets of the data for each domain that were assessed by two experts (one was arbitrarily assumed to be correct). We report these numbers for reference; they are not directly comparable to the cross-fold estimates reported for the models.

observe that CNN/Doc-CNN do not necessarily improve over the results achieved by SVM-based models, which prove to be strong baselines for longer document classification. This differs from previous comparisons in the context of classifying shorter texts. In particular, in previous work (Zhang and Wallace, 2015) we observed that CNN outperforms SVM uniformly on sentence classification tasks (the average sentence-length in these datasets was about 10). In contrast, in the datasets we consider in the present paper, documents often comprise hundreds of sentences, each in turn containing multiple words. We believe that it is in these cases that explicitly modeling which sentences are most important will result in the greatest performance gains, and this aligns with our empirical results.

Another observation is that AT-CNN does often improve performance over vanilla variants of CNN (i.e., without attention), especially on the RoB datasets, probably because these comprise longer documents. However, as one might expect, RA-CNN clearly outperforms AT-CNN by exploiting rationale-level supervision directly. And by exploiting rationale information directly, RA-CNN is able to consistently perform better than baseline CNN and SVM model variants. Indeed, we find that RA-CNN outperformed MT-SVM on all of the RoB datasets, and this was accomplished without exploiting cross-domain correlations (i.e., without multi-task learning).

Method	Accuracy
<i>Uni-SVM</i>	86.44
<i>Bi-SVM</i>	86.94
<i>RA-SVM</i>	88.89
<i>CNN</i>	85.59 (85.27, 86.17)
<i>Doc-CNN</i>	87.14 (86.70, 87.60)
<i>AT-CNN</i>	86.69 (86.28, 87.17)
<i>RA-CNN</i>	90.43 (90.11, 91.00)

Table 3: Accuracies on the movie review dataset.

7.2 Qualitative Results: Illustrative Rationales

In addition to realizing superior classification performance, RA-CNN also provides *explainable* categorizations. The model can provide the highest scoring rationales (ranked by $\max\{p_{\text{pos}}, p_{\text{neg}}\}$) for any given target instance, which in turn – by construction – are those that most influenced the final document classification.

For example, a sample positive rationale supporting a correct designation of a study as being at low risk of bias with respect to blinding of outcomes assessment reads simply *The study was performed double blind*. An example rationale extracted for a study (correctly) deemed at high risk of bias, meanwhile, reads *as the present study is retrospective, there is a risk that the woman did not properly recall how and what they experienced*

Turning to the movie reviews dataset, an example rationale extracted from a glowing review of ‘Goodfellas’ (correctly classified as positive) reads *this cinematic gem deserves its rightful place among the best films of 1990s*. While a rationale extracted

from an unfavorable review of ‘The English Patient’ asserts that *the only redeeming qualities about this film are the fine acting of Fiennes and Dafoe and the beautiful desert cinematography*.

In each of these cases, the extracted rationales directly support the respective classifications. This provides direct, meaningful insight into the automated classifications, an important benefit for neural models, which are often seen as opaque.

8 Conclusions

We developed a new model (RA-CNN) for text classification that extends the CNN architecture to directly exploit *rationales* when available. We showed that this model outperforms several strong, relevant baselines across five datasets, including vanilla and hierarchical CNN variants, and a CNN model equipped with an attention mechanism. Moreover, RA-CNN automatically provides explanations for classifications made at test time, thus providing interpretability.

Moving forward, we plan to explore additional mechanisms for exploiting supervision at lower levels in neural architectures. Furthermore, we believe an alternative approach may be a hybrid of the AT-CNN and RA-CNN models, wherein an auxiliary loss might be incurred when the attention mechanism output disagrees with the available direct supervision on sentences.

Acknowledgments

Research reported in this article was supported by the National Library of Medicine (NLM) of the National Institutes of Health (NIH) under award number R01LM012086. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health. This work was also made possible by the support of the Texas Advanced Computer Center (TACC) at UT Austin.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM.

Gregory Druck, Gideon Mann, and Andrew McCallum. 2008. Learning from labeled features using generalized expectation criteria. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 595–602. ACM.

Yoav Goldberg. 2015. A primer on neural network models for natural language processing. *arXiv preprint arXiv:1510.00726*.

Julian PT Higgins, Douglas G Altman, Peter C Gøtzsche, Peter Jüni, David Moher, Andrew D Oxman, Jelena Savović, Kenneth F Schulz, Laura Weeks, and Jonathan AC Sterne. 2011. The cochrane collaborations tool for assessing risk of bias in randomised trials. *Bmj*, 343:d5928.

Thorsten Joachims. 1998. *Text categorization with support vector machines: Learning with many relevant features*. Springer.

Rie Johnson and Tong Zhang. 2014. Effective use of word order for text categorization with convolutional neural networks. *arXiv preprint arXiv:1412.1058*.

Rie Johnson and Tong Zhang. 2015. Semi-supervised convolutional neural networks for text categorization via region embedding. In *Advances in Neural Information Processing Systems (NIPS)*, pages 919–927.

Rafal Jozefowicz, Wojciech Zaremba, and Ilya Sutskever. 2015. An empirical exploration of recurrent network architectures. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 2342–2350.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Youngjoong Ko, Jinwoo Park, and Jungyun Seo. 2002. Automatic text categorization using the importance of sentences. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pages 1–7. Association for Computational Linguistics.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Quoc V Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. *arXiv preprint arXiv:1405.4053*.

Gideon S Mann and Andrew McCallum. 2010. Generalized expectation criteria for semi-supervised learning with weakly labeled data. *The Journal of Machine Learning Research*, 11:955–984.

- Iain J Marshall, Joël Kuiper, and Byron C Wallace. 2015. Automating risk of bias assessment for clinical trials. *Biomedical and Health Informatics, IEEE Journal of*, 19(4):1406–1412.
- Iain J Marshall, Joël Kuiper, and Byron C Wallace. 2016. Robotreviewer: evaluation of a system for automatically assessing bias in clinical trials. *Journal of the American Medical Informatics Association*, 23(1):193–201.
- Tyler McDonnell, Matthew Lease, Tamer Elsayad, and Mucahid Kutlu. 2016. Why Is That Relevant? Collecting Annotator Rationales for Relevance Judgments. In *Proceedings of the 4th AAAI Conference on Human Computation and Crowdsourcing (HCOMP)*. 10 pages.
- Masaki Murata, Qing Ma, Kiyotaka Uchimoto, Hiromi Ozaku, Masao Utiyama, and Hitoshi Isahara. 2000. Japanese probabilistic information retrieval using location and category information. In *Proceedings of the fifth international workshop on on Information retrieval with Asian languages*, pages 81–88. ACM.
- Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd annual meeting on Association for Computational Linguistics*, page 271. Association for Computational Linguistics.
- Sampo Pyysalo, Filip Ginter, Hans Moen, Tapio Salakoski, and Sophia Ananiadou. 2013. Distributional semantics resources for biomedical text processing. *Proceedings of Languages in Biology and Medicine*.
- Burr Settles. 2011. Closing the loop: Fast, interactive semi-supervised annotation with queries on features and instances. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1467–1478. Association for Computational Linguistics.
- Kevin Small, Byron Wallace, Thomas Trikalinos, and Carla E Brodley. 2011. The constrained weight space svm: learning with ranked features. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 865–872.
- Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.
- Duyu Tang, Bing Qin, and Ting Liu. 2015. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1422–1432.
- 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*.
- Ainur Yessenalina, Yejin Choi, and Claire Cardie. 2010. Automatically generating annotator rationales to improve sentiment classification. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 336–341. Association for Computational Linguistics.
- Omar F Zaidan and Jason Eisner. 2008. Modeling annotators: A generative approach to learning from annotator rationales. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 31–40. Association for Computational Linguistics.
- Omar Zaidan, Jason Eisner, and Christine D Piatko. 2007. Using” annotator rationales” to improve machine learning for text categorization. In *HLT-NAACL*, pages 260–267. Citeseer.
- Matthew D Zeiler. 2012. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.
- Ye Zhang and Byron C. Wallace. 2015. A sensitivity analysis of (and practitioners’ guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820*.
- Ye Zhang, Stephen Roller, and Byron C. Wallace. 2016. Mgn-cnn: A simple approach to exploiting multiple word embeddings for sentence classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1522–1527, San Diego, California, June. Association for Computational Linguistics.