The Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-20)

CAWA: An Attention-Network for Credit Attribution

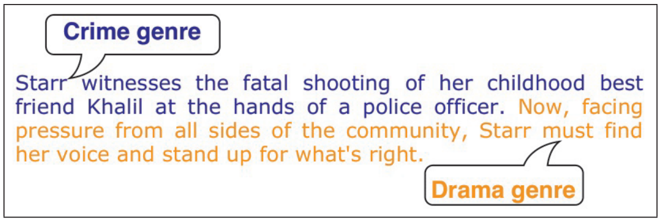
Saurav Manchanda, George Karypis
University of Minnesota, Twin Cities, USA
{manch043, karypis}@umn.edu

Abstract

Credit attribution is the task of associating individual parts in a document with their most appropriate class labels. It is an important task with applications to information retrieval and text summarization. When labeled training data is available, traditional approaches for sequence tagging can be used for credit attribution. However, generating such labeled datasets is expensive and time-consuming. In this paper, we present *Credit Attribution With Attention (CAWA)*, a neural-network-based approach, that instead of using sentence-level labeled data, uses the set of class labels that are associated with an entire document as a source of distant-supervision. CAWA combines an attention mechanism with a multilabel classifier into an end-to-end learning framework to perform credit attribution. CAWA labels the individual sentences from the input document using the resultant attention-weights. CAWA improves upon the state-of-the-art credit attribution approach by not constraining a sentence to belong to just one class, but modeling each sentence as a distribution over all classes, leading to better modeling of semantically-similar classes. Experiments on the credit attribution task on a variety of datasets show that the sentence class labels generated by CAWA outperform the competing approaches. Additionally, on the multilabel text classification task, CAWA performs better than the competing credit attribution approaches[1].

Introduction

A document can be considered as a union of segments (text-pieces), where each segment tends to talk about a single topic (class). In multilabel documents, each of the segments can be associated with one or more of the document's classes. Credit attribution (Ramage et al. 2009) refers to the task of associating these individual segments in a document with their most appropriate class labels. For example, Fig. 1 shows the IMDB plot summary of the movie *The Hate U Give*. The movie belongs to the crime and drama genres. As shown, each sentence in the plot summary can be mapped individually to the crime genre and drama genre. Credit attribution finds its application in many natural language processing and information retrieval tasks (Hearst 1997): it can improve information retrieval (by indexing documents more precisely or by giving the specific part of a document in response to a query); and text summarization (by including information from each of the document's topics).



Starr witnesses the fatal shooting of her childhood best friend Khalil at the hands of a police officer. Now, facing pressure from all sides of the community, Starr must find her voice and stand up for what's right.

Figure 1: Plot summary of the movie *The Hate U Give* from the IMDB, belonging to the crime/drama genre. The blue text depicts crime and the orange text depicts drama genre.

A straightforward way to solve credit attribution is to formulate it as a text-segment-level classification problem and collect the corresponding labeled datasets. However, manually annotating these segments (words, sentences, paragraphs, etc.) with the corresponding class labels is a tedious and expensive task. In order to reduce the need for such labeling, many methods have been developed that work in a distant-supervised fashion, such as Labeled Latent Dirichlet Allocation (LLDA) (Ramage et al. 2009), Partially Labeled Dirichlet Allocation (PLDA) (Ramage, Manning, and Dumais 2011) Multi-Label Topic Model (MLTM) (Soleimani and Miller 2017), SEG-NOISY and SEG-REFINE (Manchanda and Karypis 2018). Among them, the current state-of-the-arts are the dynamic programming based approaches SEG-NOISY and SEG-REFINE, that penalize the number of topic-switches, therefore constraining neighboring sentences to belong to the same topic. However, these approaches cannot model case where sentences can belong to multiple classes; thus, they cannot correctly model semantically similar classes.

To deal with this limitation, we developed *Credit Attribution With Attention (CAWA)*, a neural-network based approach that models multi-topic segments. CAWA uses the class labels of a multilabel document as the source of

[1] Our code and data are available at https://github.com/gurdaspuriya/cawa.

distant-supervision, to assign class labels to the individual sentences of the document. CAWA leverages the attention mechanism to compute weights that establish the relevance of a sentence for each of the classes. The attention weights, which can be interpreted as a probability distribution over the classes, allows CAWA to capture the semantically similar classes by modeling each sentence as a distribution over the classes, instead of mapping to just one class. In addition, CAWA leverages a simple average pooling layer to constrain the neighboring sentences to belong to the same class by smoothing their class distributions. CAWA uses an end-to-end learning framework to combine the attention mechanism with the multilabel classifier.

We evaluate the performance of CAWA on five datasets that were derived from different domains. On the credit attribution task, CAWA performs better than both MLTM and SEG-REFINE with respect to the sentence-labeling accuracy, with an average performance gain of 6.2% and 9.8% compared to MLTM and SEG-REFINE, respectively. On the multilabel classification task, CAWA also performs better than MLTM and SEG-REFINE. Its performance with respect to the F1 score between the predicted and the actual classes is on an average 4.1% and 1.6% better than MLTM and SEG-REFINE, respectively.

Related Work

Various unsupervised, supervised and distant-supervised methods have been developed to deal with the *credit attribution* problem. Popular examples of the unsupervised approaches include TextTiling (Hearst 1997), C99 (Choi 2000) and GraphSeg (Glavaš, Nanni, and Ponzetto 2016). Supervised approaches for text classification include the ones using decision trees (Grosz and Hirschberg 1992), multiple regression analysis (Hajime, Takeo, and Manabu 1998), exponential model (Beeferman, Berger, and Lafferty 1999), probabilistic modeling (Tür et al. 2001) and more recently, deep neural network based approaches (Badjatiya et al. 2018; Koshorek et al. 2018).

The methods proposed in this paper belong to the broad category of distant-supervised methods for text segmentation. Our methods use the set of labels that are associated with a document as a source of supervision, instead of using explicit segment-level ground truth information. Prior approaches proposed for distant-supervised text segmentation include Labeled Latent Dirichlet Allocation (LLDA) (Ramage et al. 2009), Partially Labeled Dirichlet Allocation (PLDA) (Ramage, Manning, and Dumais 2011) Multi-Label Topic Model (MLTM) (Soleimani and Miller 2017), SEG-NOISY and SEG-REFINE (Manchanda and Karypis 2018). We review these prior approaches below.

Labeled Latent Dirichlet Allocation (LLDA) (Ramage et al. 2009) is a probabilistic graphical model for *credit attribution*. It assumes a one-to-one mapping between the class labels and the topics. Like Latent Dirichlet Allocation, LLDA models each document as a mixture of underlying topics and generates each word from one topic. Unlike LDA, LLDA incorporates supervision by simply constraining the topic model to use only those topics that correspond to a document's (observed) label set. LLDA assigns each word in a document to one of the document's labels.

Partially Labeled Dirichlet Allocation (PLDA) (Ramage, Manning, and Dumais 2011) is an extension of the LLDA that allows more than one topic for every class label, and some general topics that are not associated with any class.

Multi-Label Topic Model (MLTM) (Soleimani and Miller 2017) improves upon PLDA by allowing each topic to belong to multiple, one, or even zero classes probabilistically. MLTM also assigns a label to each sentence, based on the assigned topics of the constituent words. The labels of the documents are generated from the labels of its sentences.

A common problem with the above-mentioned approaches is that they model the document as a bag of word/sentences and do not take into consideration the structure within a document, i.e., neighboring sentences tend to talk about the same topic. Recently, we proposed dynamic programming based approaches SEG-NOISY and SEG-REFINE (Manchanda and Karypis 2018) to segment the documents, that penalizes the number of segments, therefore constraining neighboring sentences to belong to the same topic. However, SEG-NOISY and SEG-REFINE approaches model each sentence as belonging to a single class, thus facing the problem of correctly modeling the semantically similar classes, in which case, each sentence can belong to multiple classes.

Another line of work related to the problem addressed in this paper is Rationalizing Neural Predictions (Lei, Barzilay, and Jaakkola 2016). The previous work selects a subset of the words in a document as a rationale for the predictions made by a neural network, where a rationale must be short. As such, this assumption makes sense for some domains, such as sentiment analysis, as a few words are sufficient to describe the sentiment. However, in our work, we do not make this assumption and develop methods for any general multilabel document. Another similar work for distant-supervised sentiment analysis (Angelidis and Lapata 2018) uses attention-mechanism to identifying positive and negative text snippets, only using the overall sentiment rating as supervision. As we explain later, in the case of multilabel documents (such as multi-aspect ratings), the vanilla-attention mechanism can assign high attention-weights to the sentences that provide negative evidence for a class. The proposed approach addresses this limitation and is well-suited for credit-attribution in multilabel documents.

Definitions and Notations

Let C be a set of classes and D be a set of multilabel documents. For each document $d \in D$, let $L_d \subseteq C$ be its set of classes and let $|d|$ be the number of sentences that it contains. The approach developed in this paper assumes that in multilabel documents, each sentence can be labeled with class label(s) from that document. In particular, given a document d we assume that each sentence $d[i]$ can be labeled with a class $y(d, i) \in L_d$. We seek to find these sentence-level class labels, the training data being the multilabel documents and their class labels, i.e., we do not have access to the sentence-level class labels for training.

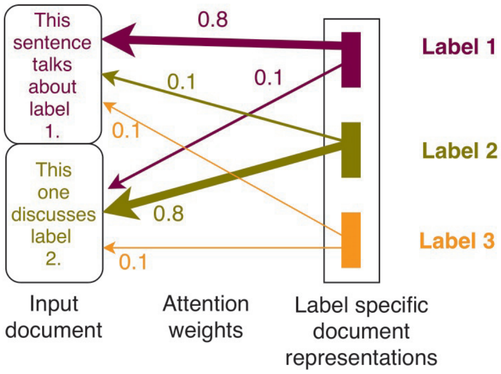

This sentence talks about label 1.
This one discusses label 2.
0.8
0.1
0.1
0.1
0.8
0.1
Label 1
Label 2
Label 3
Input document
Attention weights
Label specific document representations

Figure 2: Example of the attention weights. Each sentence contributes towards the class-specific document representation, the extent of contribution, and hence to relevance for a class, is decided by the attention weights.

Credit Attribution With Attention (CAWA)

As discussed earlier, the existing approaches for credit attribution suffer from the limitations of either not modeling semantically similar classes or not exploiting the local structure within the documents. In order to address these limitations, we present a neural-network based approach *Credit Attribution With Attention (CAWA)*. CAWA addresses these limitations by (i) capturing the semantically similar classes by modeling each sentence as a distribution over the classes, instead of mapping to just one class; and (ii) leveraging a simple average pooling layer to constrain the neighboring sentences to have similar class distribution; thus, leveraging the local structure within the documents.

To this end, CAWA combines an attention mechanism with a multilabel classifier and uses this multilabel classifier to predict the classes of an input document. For each predicted class, the attention mechanism allows CAWA to precisely identify the sentences of the input document which are relevant towards predicting that class. Using these relevant sentences, CAWA estimates a class-specific document representation for each class. Finally, each sentence is assigned the class, for which it is most relevant, i.e., has the highest attention weight. Figure 2 shown an example of sentence-labeling using the attention weights. Additionally, CAWA uses a simple average pooling layer to constrain the neighboring sentences to have similar attention weights (class distribution). We explain CAWA in detail in this section.

Architecture

CAWA consists of three components: (i) a sentence representation generator, which is responsible for generating a representation of the sentences in the input document; (ii) an attention module, which is responsible for generating a class-specific document representation from the sentence representations, and (iii) a multilabel classifier, which is responsible for predicting the classes of the document using the class-specific document representations as input. These three components form an end-to-end learning framework as shown in Figure 3. We explain each of these components in detail in this section.

Sentence-representation generator (SRG): The SRG takes the document as an input and generates two different representations for each sentence in the document. The two representations correspond to the *keys* and the *values* that will be taken as input by the attention mechanism, as explained in next section. For both *keys* and *values*, SRG generates the representation of a sentence as the average of the representations of the constituent words of the sentence, i.e.,

$$\mathbf{k}^d(i) = \frac{1}{|d[i]|}\sum_{x\in d[i]}\mathbf{k}^w(x); \quad \mathbf{v}^d(i) = \frac{1}{|d[i]|}\sum_{x\in d[i]}\mathbf{v}^w(x),$$

where $d[i]$ is the ith sentence of document d, $\mathbf{k}^d(i)$ is the key-representation of $d[i]$, $\mathbf{k}^w(x)$ is the key-representation of word x, $\mathbf{v}^d(i)$ is the value-representation of $d[i]$ and $\mathbf{v}^w(x)$ is the value-representation of word x. These representations for the words are estimated during the training.

Attention module: The attention module takes the sentence representations (*keys* and *values*) as input and outputs the class-specific representation of the document, one document-representation for each class. Since the different sentences have difference relevance for each class, we estimate the class-specific representations as a weighted average of the value-representations of the sentences. We calculate the *attention* weights for this weighted average using a feed-forward network. Specifically, we estimate the class-specific representations as, $\mathbf{r}^d(c) = \sum_{i=1}^{|d|} a(d,i,c) \times \mathbf{v}^d(i)$, where $\mathbf{r}^d(c)$ is the class-specific representation of document d for class c, $a(d,i,c)$ is the attention-weight for of the ith sentence of d for class c. The feed-forward network to calculate the attention weights takes as input the *key* representation of a sentence and outputs the attention weight of the sentence for each class. This feed-forward network plays the role of the sentence classifier and outputs the probability of the input sentence belonging to each of the classes, on its output layer. We implement this feed-forward network with two hidden-layers, and we use softmax on the output layer to calculate the attention-weights. To leverage the local structure within the document, i.e., to constrain the neighboring sentences to have similar class distributions, we apply average pooling before the softmax layer. Average pooling smooths out the neighboring class distributions and cancels the effect due to random variation. Note that, we can also use more flexible sequence modeling approaches, such as Recurrent Neural Networks (RNNs) to leverage the local structure. But, we choose to use a simple average pooling layer, due to its simplicity. We also add a residual connection between the first hidden layer and the output layer, which eases the optimization of the model (He et al. 2016). The architecture for the attention mechanism is shown in Figure 4.

Multilabel classifier: Several architectures and loss-functions have been proposed for multilabel classification, such as Backpropagation for Multi-Label Learning (BP-MLL) (Zhang and Zhou 2007). However, as the focus of this paper is credit attribution and not multilabel classification, we simply implement the multilabel classifier as a sep-

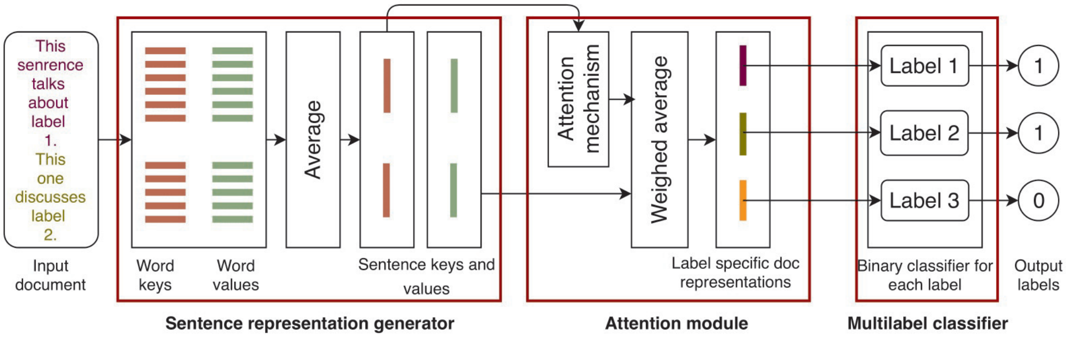


Figure 3: CAWA architecture with an example. The input document consists of two sentences, having class labels 1 and 2 respectively. The *Sentence-representation generator* generates the *key* and *value* representation for these sentences. The attention module generates class-specific document representations using the key and value representations of the sentences. Finally, the multilabel classifier, uses these class-specific representations, to predict the correct classes of the document. Although we don't have direct supervision about the sentence-level class labels, the attention mechanism allows us to find how much each sentence is relevant to a class, that can be used to predict the sentence-level class labels.

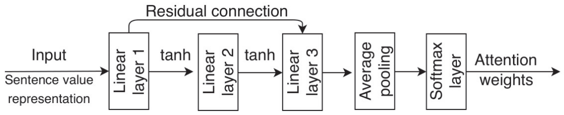


Figure 4: Architecture of the attention mechanism.

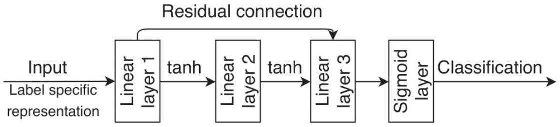


Figure 5: Architecture of the per-class binary classifier.

arate binary classifier for each class. Therefore, each binary classifier predicts whether a particular class is present in the document or not. The input to each of these binary classifiers is the class-specific representation, which is the output of the attention module. We implement each of these binary classifiers as a feed-forward network with two hidden layers and use sigmoid on the output layer to predict the probability of the document belonging to that class. The architecture for the class-specific binary classifiers is shown in Figure 5.

Model estimation

To quantify the loss for predicting the classes of a document, we minimize the weighted binary cross-entropy loss (Nam et al. 2014), which is a widely used loss function for multilabel classification. The weighted binary cross-entropy loss associated with all the documents in collection D is given by:

$$L_C(D) = -\frac{1}{|D|}\sum_{d\in D}\sum_{c\in C} w_c(y(d,c)\log(s(d,c)) \\ + (1-y(d,c))\log(1-s(d,c))),$$

where $y(d,c) = 1$ if the class c is present in document d, and $y(d,c) = 0$ otherwise, $s(d,c)$ is the prediction probability of document d belonging to class c, and w_c is the class-specific weight for class c. This weight w_c is used to handle the class imbalance by increasing the importance of infrequent classes (upsampling), and we empirically set it to $w_c = \sqrt{|D|/n_c}$, where n_c is the number of documents belonging to the class c. Note that we require the sentences to be labeled with the class which they describe. However, the attention mechanism can also assign high attention-weights to the sentences that provide a negative-signal for a class. For example, if a document can exclusively belong to only one class A and B, the text describing one of the classes (say A) will also provide a negative signal for the other class (B), and hence, will get high attention-weight for both classes A and B. To constraint that the attention is only focused on the classes that are actually present in the document, we introduce *attention loss*, which penalizes the attention on the absent classes, and is given by

$$L_S(D) = -\frac{1}{|D|}\sum_{d\in D}\frac{1}{|d|}\sum_{i=1}^{|d|}\sum_{c\in C}(1-y(d,c))\log(1-a(d,i,c)),$$

where $a(d,i,c)$ is the attention weight for class c on the ith sentence of the document d. To estimate the CAWA, we minimize the weighted sum of both $L_C(D)$ and $L_S(D)$, given by $L(D) = \alpha L_C(D) + (1-\alpha)L_S(D)$, where α is a hyperparameter to control the relative contribution of $L_C(D)$ and $L_S(D)$ towards the final loss.

Segment inference

We can directly use the estimated attention-weights to assign a class to each sentence, corresponding to the class with the maximum attention-weight. However, to ensure the consensus between the predicted sentence-level classes and document's classes, we use a linear combination of the attention-weights and document's predicted class-probabilities to assign a class to each sentence, i.e.,

$$l(d,i) = \underset{c}{\operatorname{argmax}}(\beta \times a(d,i,c) + (1-\beta) \times y(d,c)), \quad (1)$$

where $l(d, i)$ is the predicted class for the ith sentence of d and β is a hyperparameter to control the relative contribution of attention-weights and document's classification probability. Additionally, $y(d, c)$ acts as a global bias term, and makes the sentence-level predictions less prone to random variation in the attention weights.

Experimental methodology

Datasets

We performed experiments on five multilabel text datasets from different domains: *Movies* (Bamman, O'Connor, and Smith 2014), *Ohsumed* (Hersh et al. 1994), *TMC2007*[2], *Patents*[3], *Delicious* (Zubiaga et al. 2009). For both the credit attribution and multilabel classification tasks, we used the same training and test dataset split as used in (Manchanda and Karypis 2018). For the credit attribution, the test dataset is synthetic, and each test document corresponds to multiple single-label documents concatenated together (thus, giving us ground truth segment labels for a document). Additionally, we also use a validation dataset, created in a similar manner to this test dataset, for the hyperparameter selection.

Baselines

Although the number of methods that have been specifically developed to solve the credit attribution problem is small, any multilabel classifier can be used to perform credit attribution, by training on the multilabel documents and predicting the classes of the individual sentences. Thus, apart from the credit attribution specific approaches, we compare CAWA against several multilabel classification approaches. Specifically, we chose our baselines from diverse domains such as graphical models, deep neural networks, dynamic programming, etc., as described below:

- *SEGmentation with REFINEment (SEG-REFINE)* (Manchanda and Karypis 2018): A dynamic programming based approach that constrains neighboring sentences to belong to the same topic.
- *Multi-Label Topic Model (MLTM)* (Soleimani and Miller 2017): A generative approach, that generates the classes of a document from the classes of its sentences, which are further generated from the classes of its words.
- *Deep Neural Network with Attention (DNN+A):* As mentioned earlier, any multilabel classifier can be used to perform credit attribution. DNN+A is a neural network based multilabel classifier. For a fair comparison, we use the same architecture as CAWA for DNN+A, except the components specific to CAWA (attention loss and average pooling layer).
- *Deep Neural Network without Attention (DNN-A):* Same as DNN+A, except the attention, i.e., each class gives equal emphasis on all the sentences.
- *Multi-Label k-Nearest Neighbor (ML-KNN)* (Zhang and Zhou 2007): ML-KNN is a popular method for multilabel classification. It uses the k nearest neighbors and Bayesian inference to assign classes to the text example.
- *Binary Relevance - Multinomial Naive Bayes (BR-MNB):* Binary relevance amounts to independently training a binary classifier for each class. We use Multinomial Naive Bayes as the per class binary classifier, which is a popular classical approach for text classification.

Performance Assessment Metrics

Credit attribution: For evaluation on the credit attribution task, we look into two different metrics. The first is *per-point prediction accuracy* (PPPA) and the second is Segment OVerlap score (SOV) (Rost, Sander, and Schneider 1994). PPPA corresponds to the fraction of sentences that are predicted correctly. As a single-point measure, PPPA does not take into account the correlation between the neighboring sentences. On the other hand, SOV measures how well the observed and the predicted segments align with each other. It takes several factors into consideration including the number of segments in a document, the averaged segment length, and the distribution of the length values. As a result, it allows some variations at the boundaries of the segments by assigning some allowance and can handle extreme cases (e.g., penalizing wrong predictions) reasonably by providing a sliding scale of segment overlap.

Multilabel classification: To evaluate CAWA on the multilabel classification task, we looked into three metrics: F1, AUC_μ and AUC_M. For a given document, F1 score is the harmonic mean of the precision and recall based on the predicted classes and the observed classes. We report the mean of F1 score over all the test documents. Area Under the Receiver Operating Characteristic Curve (AUC) (Bradley 1997) gives the probability that a randomly chosen positive example ranks above a randomly chosen negative example. We report AUC both under the micro (AUC_μ) and macro (AUC_M) settings. AUC_M computes the metric independently for each class and then takes the average (hence treating all classes equally), whereas AUC_μ aggregates the contributions of all classes to compute the metric.

Parameter selection

We chose the values of α and β individually for all the datasets, using grid search in the range $\{0.0, 0.1, \ldots, 1.0\}$, based on the best validation SOV score. For CAWA, DNN+A, and DNN-A, the number of nodes in the each of the hidden layer, all representations' length, as well as the batch size for training the CAWA was set to 256. For regularization, we used a dropout (Srivastava et al. 2014) of 0.5 between all layers, except the output layer. For optimization, we used the ADAM (Kingma and Ba 2014) optimizer. We trained all the models for 100 epochs, with the learning-rate set to 0.001. The keys and values embeddings are initialized randomly. For average pooling in CAWA, we fixed the kernel-size to three. For ML-KNN, we used cosine similarity measure to find the nearest neighbors which is a commonly used similarity measure for text documents. We chose the number of neighbors (k) for ML-KNN based on the best SOV score of the validation

[2] https://c3.nasa.gov/dashlink/resources/138/

[3] http://www.patentsview.org/download/

Table 1: Performance comparison results.

Dataset	Model[*]	SOV	PPPA	F1	AUC_μ	AUC_M
Movies	CAWA	**0.50**	0.38	**0.65**	0.81	0.78
	SEG-REF	0.49	0.36	0.63	0.81	0.80
	MLTM	**0.50**	**0.40**	**0.65**	0.82	0.80
	DNN+A	0.33	0.27	0.62	0.84	0.82
	DNN-A	0.33	0.27	0.61	**0.85**	0.83
	ML-KNN	0.38	0.30	0.63	0.83	0.81
	BR-MNB	0.39	0.31	0.53	0.82	**0.84**
Ohsumed	CAWA	**0.65**	**0.55**	0.64	0.93	0.89
	SEG-REF	0.63	0.47	0.65	**0.94**	**0.92**
	MLTM	0.56	0.47	0.60	0.93	0.91
	DNN+A	0.44	0.37	**0.67**	**0.94**	**0.92**
	DNN-A	0.33	0.31	0.58	**0.94**	**0.92**
	ML-KNN	0.48	0.38	0.59	0.90	0.87
	BR-MNB	0.29	0.30	0.31	0.82	0.71
TMC2007	CAWA	0.56	**0.47**	0.68	0.95	0.91
	SEG-REF	**0.59**	0.44	0.68	0.95	0.90
	MLTM	0.49	0.43	0.64	**0.96**	**0.92**
	DNN+A	0.43	0.37	0.68	**0.96**	**0.92**
	DNN-A	0.35	0.34	0.59	**0.96**	**0.92**
	ML-KNN	0.45	0.35	**0.71**	0.95	0.89
	BR-MNB	0.30	0.33	0.62	0.89	0.72
Patents	CAWA	**0.58**	**0.50**	0.61	0.88	0.86
	SEG-REF	0.56	0.45	0.61	0.86	0.85
	MLTM	0.55	0.48	0.59	0.85	0.84
	DNN+A	0.53	0.43	**0.64**	**0.89**	0.87
	DNN-A	0.51	0.42	0.63	**0.89**	**0.88**
	ML-KNN	0.45	0.37	0.51	0.82	0.80
	BR-MNB	0.50	0.43	0.50	0.87	0.86
Delicious	CAWA	**0.50**	**0.39**	**0.52**	0.85	0.84
	SEG-REF	0.48	0.36	0.49	0.85	0.85
	MLTM	0.49	0.37	0.50	0.84	0.83
	DNN+A	0.22	0.18	0.38	0.87	0.86
	DNN-A	0.21	0.17	0.36	**0.88**	**0.87**
	ML-KNN	0.24	0.19	0.35	0.82	0.80
	BR-MNB	0.25	0.19	0.05	0.76	0.73

[*] The models CAWA, SEG-REFINE (abbreviated SEG-REF above) and MLTM have been specifically designed to solve the credit attribution problem, while the models DNN+A, DNN-A, ML-KNN and BR-MNB are multilabel classification approaches.

set. For ML-KNN and BR-MNB, we used the implementation as provided by scikit-multilearn[4]. For MLTM and SEG-REFINE, we find the hyperparameters based on the best validation SOV score. Further details are available at https://github.com/gurdaspuriya/cawa.

Results and Discussion

Credit attribution

The metrics SOV and PPPA in Table 1 show the performance for various methods on the credit attribution task. The credit attribution specific approaches (CAWA, SEG-REFINE, and MLTM) perform considerably better than the other multilabel approaches (DNN+A, DNN-A, ML-KNN, and BR-MNB). CAWA performs better than the SEG-REFINE and MLTM on the PPPA metric for the *Ohsumed*, *TMC2007*, *Patents* and *Delicious* datasets. The average performance gain for the CAWA on the PPPA is 6.2% compared to MLTM and 9.8% compared to SEG-REFINE. Additionally, CAWA also performs at par, if not better, than the SEG-REFINE and MLTM on the SOV metric. This shows that CAWA is able to find contiguous segments, without compromising on the sentence-level accuracy.

Table 2: Sentence classification performance on similar classes.

Dataset	Class	Model	F1
Ohsumed	Nutritional/ metabolic	CAWA	**0.68**
		SEG-REFINE	0.64
	Endocrine disease	CAWA	**0.39**
		SEG-REFINE	0.26
Patents	Electricity	CAWA	**0.53**
		SEG-REFINE	0.48
	Physics	CAWA	**0.41**
		SEG-REFINE	0.33
Delicious	Health	CAWA	**0.50**
		SEG-REFINE	0.47
	Recipes	CAWA	**0.62**
		SEG-REFINE	0.58

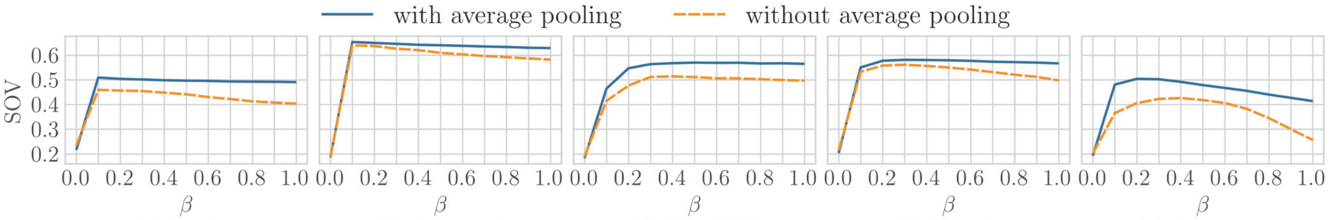
To validate our hypotheses that CAWA can accurately model the semantically similar classes as compared to the SEG-REFINE, we looked into the performance of both CAWA and SEG-REFINE on the two most similar classes for each of the *Ohsumed*, *Patents* and *Delicious* datasets. To measure the similarity between the two classes, we calculated the Jaccard similarity (Jaccard 1901) between these classes, based on the number of documents in which they occur. For each of these selected classes, we calculated the F1 score based on the predicted and actual classes of the sentences in the segmentation dataset. Table 2 shows the results for this analysis. For the *Ohsumed* dataset, the two selected classes are Nutritional/Metabolic disease and Endocrine disease, which are very similar. Likewise, the selected classes for the *Patents* and *Delicious* dataset are also similar. We see that, for all the selected classes, CAWA performs better than SEG-REFINE, illustrating the effectiveness of CAWA on modeling semantically similar classes. We further investigate the effect of various parameters of CAWA on the credit attribution task later in the Ablation study section.

In addition, DNN+A also performs considerably better than the DNN-A on both SOV and PPPA metrics for all the datasets. This shows the effectiveness of the proposed attention architecture on modeling the multilabel documents.

Multilabel classification

The metrics F1, AUC_μ and AUC_M in Table 1 show the performance of different methods on the classification task. Similar to the credit attribution task, CAWA, in general, performs better than the competing credit attribution approaches (SEG-REFINE and MLTM) on the F1 metric, with an average performance gain of 4.1% over MLTM and 1.6% over SEG-REFINE. This shows that the classes predicted for the sentences by CAWA correlate better with the document classes as compared to the classes predicted by the competing credit attribution approaches. This can be attributed to the way we calculate the sentence classes (Equation (1)), which ensures the consensus between the pre-

[4] http://scikit.ml/



(a) Movies (b) Ohsumed (c) TMC2007 (d) Patents (e) Delicious

Figure 6: Change in the SOV with β as it is increased from 0 to 1 for the two cases (i) average-pooling layer is used, and (ii) average-pooling layer is not used. The plots correspond the values of α corresponding to the best validation performance.

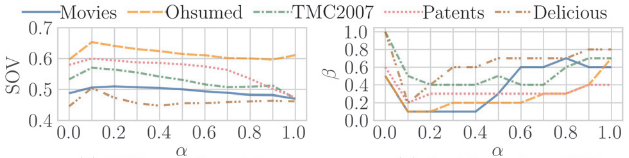
dicted sentence-level and document-level classes. Additionally, CAWA performs at par with the competing credit attribution approaches on the AUC_μ and AUC_M metrics, further illustrating the effectiveness of CAWA.

Compared to the approaches specific to the multilabel classification task, either CAWA or DNN+A achieve the best performance on the F1 metric on all but the TMC2007 dataset, where ML-KNN achieves the best performance. This further verifies the effectiveness of the proposed attention architecture on correctly modeling the multilabel documents. On the AUC metrics, we see that DNN+A outperforms CAWA. This is the result of attention loss, which while helping the network to perform credit attribution, damages its ability to perform global document classification. As discussed earlier, the vanilla attention mechanism (as used in the DNN+A) can assign high attention-weights to the sentences that provide a negative-signal for a class. Attention loss constrains that the attention is only focused on the classes that are actually present in the document. Thus, while DNN+A also leverages the negative signals for the classes to make its predictions, CAWA, by design, ignores these negative signals, which adversely affects its multilabel classification performance. We further investigate the effect of attention loss on the multilabel classification task later in the Ablation study section.

Ablation study

Effect of average pooling and β: Figure 6 shows the change in SOV metric with change in β for all the datasets. For each dataset, we plot the SOV metric as β is increased from 0.0 to 1.0 for the two cases (i) average-pooling layer is used, and (ii) average-pooling layer is not used. For both the cases, when $\beta = 0$, each sentence gets the same class, which is the class with the maximum prediction probability for the complete document. As β increases, the effect of the attention-weights starts pitching in, leading to each sentence getting its own class, thus a sharp jump in the performance on the SOV metric. However, as the β increases, the contribution of attention weights outpowers the overall document class probabilities, and the predicted sentence-classes become more prone to noise in the attention weights, thus leading to performance degradation for large β.

Comparing the performance curves of the case when the average-pooling layer is used to the one when it is not used, the average pooling leads to better performance for all values of β. Thus, average pooling effectively constrains the nearby sentences to have similar attention weights, leading to better performance on the SOV metric.

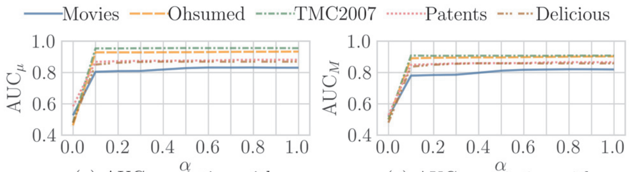


(a) SOV variation with α (b) β with the best SOV

Figure 7: Sub-figure (a) shows the change in SOV with change in α. Sub-figure (b) shows the β values for which the maximum SOV is obtained for each α.

Effect of α: Figure 7 shows the change in performance on the SOV metric with change in α for all the datasets. Figure 7(a) reports the maximum value of SOV for each α over all the β values. Figure 7(b) reports the corresponding value of β for each α that gives the maximum performance on the SOV metric. For all the cases, as the α increases from 0.0 to 0.1, SOV shows a sharp increase, which can be attributed to the effect of classification loss ($L_C(D)$) pitching in. Additionally, we see that as the α increases, the corresponding value of β giving the maximum performance also increases in general. As the α increases, the contribution of attention loss decreases, thus requiring more contribution from the attention weights to accurately predict the sentence classes. This explains the increase in the values of β values, as the value of α increases. The exceptionally high value of β when $\alpha = 0$ can be explained as follows: $\alpha = 0$ corresponds to the case when we are only minimizing the attention loss ($L_S(D)$), and ignoring the loss for predicting the document's classes ($L_C(D)$). The multilabel classifier does not get trained at all in this case, leading to $y(d, c)$ getting random values. Therefore, β takes large values to ignore the contribution of $y(d, c)$ (which is random) towards the sentence-level labels, so as to make correct predictions.

Figures 8(a) and 8(b) show the change in performance on the AUC_{mu} and AUC_M metrics with change in α, respectively. For both the metrics, the performance increases with an increase in α, i.e.., the performance on the AUC metrics is negatively impacted by the attention loss. As explained earlier, attention loss ignores the sentences that provide the negative signals for the classes to make its predictions, thus,



(a) AUC_μ variation with α

(a) AUC_M variation with α

Figure 8: Sub-figures (a) and (b) shows the change in AUC_μ and AUC_M, with change in α, respectively.

adversely affects the multi-label classification performance.

The average pooling, α and β have the same effect on the PPPA metric too.

Conclusion

In this paper, we proposed *Credit Attribution With Attention (CAWA)*, an end-to-end attention-based network to perform credit attribution on documents. CAWA addresses the limitations of the prior approaches by (i) modeling the semantically similar classes by modeling each sentence as a distribution over the classes, instead of mapping to a single class; and (ii) leveraging a simple average pooling layer to constrain the neighboring sentences to have similar class distribution. A loss function is proposed to constrain that the attention is only focused on the classes present in the document. The experiments demonstrate the superior performance of CAWA over the competing approaches. Our work makes a step towards leveraging distant-supervision for credit attribution and envision that our work will serve as a motivation for other applications that rely on the labeled training data, which is expensive and time-consuming.

Acknowledgment

This work was supported in part by NSF (1447788, 1704074, 1757916, 1834251), Army Research Office (W911NF1810344), Intel Corp, and the Digital Technology Center at the University of Minnesota. Access to research and computing facilities was provided by the Digital Technology Center and the Minnesota Supercomputing Institute.

References

Angelidis, S., and Lapata, M. 2018. Multiple instance learning networks for fine-grained sentiment analysis. *TACL*.

Badjatiya, P.; Kurisinkel, L. J.; Gupta, M.; and Varma, V. 2018. Attention-based neural text segmentation. In *ECIR*.

Bamman, D.; O'Connor, B.; and Smith, N. A. 2014. Learning latent personas of film characters. In *ACL*, 352.

Beeferman, D.; Berger, A.; and Lafferty, J. 1999. Statistical models for text segmentation. *Machine learning* 34(1-3).

Bradley, A. P. 1997. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern recognition* 30(7):1145–1159.

Choi, F. Y. 2000. Advances in domain independent linear text segmentation. In *NAACL*, 26–33.

Glavaš, G.; Nanni, F.; and Ponzetto, S. P. 2016. Unsupervised text segmentation using semantic relatedness graphs. In *Association for Computational Linguistics*.

Grosz, B., and Hirschberg, J. 1992. Some intonational characteristics of discourse structure. In *Second international conference on spoken language processing*.

Hajime, M.; Takeo, H.; and Manabu, O. 1998. Text segmentation with multiple surface linguistic cues. In *ACL*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *IEEE CVPR*.

Hearst, M. A. 1997. Texttiling: Segmenting text into multi-paragraph subtopic passages. *Computational linguistics*.

Hersh, W.; Buckley, C.; Leone, T.; and Hickam, D. 1994. Ohsumed: an interactive retrieval evaluation and new large test collection for research. In *SIGIR'94*.

Jaccard, P. 1901. Étude comparative de la distribution florale dans une portion des alpes et des jura. *Bull Soc Vaudoise Sci Nat* 37:547–579.

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

Koshorek, O.; Cohen, A.; Mor, N.; Rotman, M.; and Berant, J. 2018. Text segmentation as a supervised learning task. In *NAACL*, 469–473.

Lei, T.; Barzilay, R.; and Jaakkola, T. 2016. Rationalizing neural predictions. In *EMNLP*, 107–117.

Manchanda, S., and Karypis, G. 2018. Text segmentation on multi-label documents: A distant-supervised approach. In *ICDM*, 1170–1175. IEEE.

Nam, J.; Kim, J.; Mencía, E. L.; Gurevych, I.; and Fürnkranz, J. 2014. Large-scale multi-label text classification–revisiting neural networks. In *ECML PKDD*.

Ramage, D.; Hall, D.; Nallapati, R.; and Manning, C. D. 2009. Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora. In *EMNLP*, 248–256.

Ramage, D.; Manning, C. D.; and Dumais, S. 2011. Partially labeled topic models for interpretable text mining. In *Proceedings of the 17th SIGKDD*. ACM.

Rost, B.; Sander, C.; and Schneider, R. 1994. Redefining the goals of protein secondary structure prediction. *Journal of molecular biology* 235(1):13–26.

Soleimani, H., and Miller, D. J. 2017. Semisupervised, multilabel, multi-instance learning for structured data. *Neural computation* 29(4):1053–1102.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: A simple way to prevent neural networks from overfitting. *JMLR* 15(1):1929–1958.

Tür, G.; Hakkani-Tür, D.; Stolcke, A.; and Shriberg, E. 2001. Integrating prosodic and lexical cues for automatic topic segmentation. *Computational linguistics* 27(1):31–57.

Zhang, M.-L., and Zhou, Z.-H. 2007. Ml-knn: A lazy learning approach to multi-label learning. *Pattern recognition*.

Zubiaga, A.; García-Plaza, A. P.; Fresno, V.; and Martínez, R. 2009. Content-based clustering for tag cloud visualization. In *ASONAM*, 316–319. IEEE.