Corpus-level and Concept-based Explanations for Interpretable Document Classification

TIAN SHI, Virginia Tech, USA
XUCHAO ZHANG, Virginia Tech, USA
PING WANG, Virginia Tech, USA
CHANDAN K. REDDY, Virginia Tech, USA

Using attention weights to identify information that is important for models’ decision making is a popular approach to interpret attention-based neural networks. This is commonly realized in practice through the generation of a heat-map for every single document based on attention weights. However, this interpretation method is fragile and it is easy to find contradictory examples. In this paper, we propose a corpus-level explanation approach, which aims to capture causal relationships between keywords and model predictions via learning the importance of keywords for predicted labels across a training corpus based on attention weights. Based on this idea, we further propose a concept-based explanation method that can automatically learn higher-level concepts and their importance to model prediction tasks. Our concept-based explanation method is built upon a novel Abstraction-Aggregation Network, which can automatically cluster important keywords during an end-to-end training process. We apply these methods to the document classification task and show that they are powerful in extracting semantically meaningful keywords and concepts. Our consistency analysis results based on an attention-based Naïve Bayes classifier also demonstrate that these keywords and concepts are important for model predictions.

CCS Concepts: • Information systems → Sentiment analysis; Clustering and classification; Information extraction.

Additional Key Words and Phrases: Attention mechanism, model interpretation, document classification, sentiment classification, concept-based explanation.

ACM Reference Format:

1 INTRODUCTION

Attention Mechanisms [2] have boosted performance of deep learning models in a variety of natural language processing (NLP) tasks, such as sentiment analysis [35, 47], semantic parsing [46], machine translation [30], reading comprehension [9, 17] and others. Attention-based deep learning models have been widely investigated not only because they achieve state-of-the-art performance, but also because they can be interpreted by identifying important input information via visualizing heat-maps of attention weights [11, 40, 42, 45], namely attention visualization. Therefore, attention
mechanisms help end-users to understand models and diagnose trustworthiness of their decision making.

However, the attention visualization approach still suffers from several drawbacks: 1) The fragility of attention weights can easily make end-users find contradicting examples, especially for noisy data and cross-domain applications. For example, a model may attend on punctuation or stop-words. 2) Attention visualization cannot automatically extract high-level concepts that are important for model predictions. For example, when a model assigns news articles to *Sports*, relevant keywords may be *player*, *basketball*, *coach*, *nhl*, *golf*, and *nba*. Obviously, we can build three concepts/clusters for this example, i.e., roles (*player*, *coach*), games (*basketball*, *soccer*), and leagues (*nba*, *nhl*). 3) Attention visualization still relies on human experts to decide if keywords attended by models are important to model predictions.

There have been some studies that attempt to solve these problems. For example, Jain and Wallace [20], Serrano and Smith [38] focused on studying if attention can be used to interpret a model, however, there are still problems in their experimental designs [48]. Yeh et al. [53] tried to apply a generic concept-based explanation method to interpret BERT models in the text classification task, however, they did not obtain semantically meaningful concepts for model predictions. Antognini and Faltings [1] introduced a concept explanation method that first extracts a set of text snippets as concepts and infers which ones are described in the document, and then it explained the predictions of sentiment with a linear aggregation of concepts. In this paper, we propose a general-purpose corpus-level explanation method and a concept-based explanation method based on a novel Abstraction-Aggregation Network (AAN) to tackle the aforementioned drawbacks of attention visualization. We summarize the primary contributions of this paper as follows:

- To solve the first problem, we propose a **corpus-level explanation** method, which aims to discover causal relationships between keywords and model predictions. The importance of keywords is learned across the training corpus based on attention weights. Thus, it can provide more robust explanations compared to attention visualization case studies. The discovered keywords are semantically meaningful for model predictions.

- To solve the second problem, we propose a **concept-based explanation method** (case-level and corpus-level) that can automatically learn semantically meaningful concepts and their importance to model predictions. The concept-based explanation method is based on an AAN that can automatically cluster keywords, which are important to model predictions, during the end-to-end training for the main task. Compared to the basic attention mechanisms, the models with AAN do not compromise on classification performance or introduce any significant number of new parameters.

- To solve the third problem, we build a **Naïve Bayes Classifier** (NBC), which is based on an **attention-based bag-of-words document representation** technique and the causal relationships discovered by the corpus-level explanation method. By matching predictions from the model and NBC, i.e., consistency analysis, we can verify if the discovered keywords are important to model predictions. This provides an automatic verification pipeline for the results from the corpus-level explanation and concept-based explanation methods.

The rest of this paper is organized as follows: In Section 2, we introduce related work of feature-based and concept-based explanation. In Section 3, we first present details of our proposed abstraction-aggregation network (AAN), and then discuss corpus-level and concept-based explanation methods. In Section 4, we evaluate different self-attention and AAN based models on three different datasets. We also show how corpus-level and concept-based explanations can help in the interpretation of attention-based classification models and can potentially provide a better understanding of the training corpus. Our discussion concludes in Section 5.
2 RELATED WORK
Increasing the interpretability on machine learning models has become an important topic of research in recent years. Most prior work [14, 27, 29, 41] focus on interpreting models via feature-based explanations, which alters individual features such as pixels and word-vectors in the form of either deletion [36] or perturbation [43]. However, these methods usually suffer from the reliability issues when adversarial perturbations [12] or even simple shifts [23] of the input data. Moreover, the feature-based approaches explain the model behavior locally [36] for each data samples without a global explanation [13, 21] on how the models make their decisions. In addition, feature-based explanation is not necessarily the most effective way for human understanding.

To alleviate the issues of feature-based explanation models, some researchers have focused on explaining the model results in the form of high-level human concepts [4, 6, 8, 44, 50, 54, 56]. Unlike assigning the importance scores to individual features, the concept-based methods use the corpus-level concepts as the interpretable unit. For instance, concept "wheels" can be used for detecting the vehicle images and concept "Olympic Games" for identifying the sports documents. However, most of the existing concept-based approaches require human supervision in providing hand-labeled examples of concepts, which is labor intensive and some human bias can be introduced in the explanation process [19, 21, 39]. Recently, automated concept-based explanation methods [5, 53] are proposed to identify higher-level concepts that are meaningful to humans automatically. However, they have not shown semantically meaningful concepts on text data. In text classification area, most of the existing approaches focus on improving the classification performance, but ignore the interpretability of the model behaviors [52]. Liu and Avci [27] utilize the feature attribution method to help users interpret the model behavior. Bouchacourt and Denoyer [5] propose a self-interpretable model through unsupervised concept extraction. However, it requires another unsupervised model to extract concepts. Different from these studies, our corpus-level explanation method can be generally applied to self-attention mechanisms, and our concept-based explanation method, which is based on a hierarchical attention network, can automatic learn concepts during the end-to-end training.

3 PROPOSED WORK
In this section, we first introduce the classification framework and our Abstraction-Aggregation Network (AAN). Then, we discuss the corpus-level explanation, concept-based explanation, and attention-based Naive Bayes Classifier.

3.1 The Proposed Model
3.1.1 Basic Framework. A typical document classification model is equipped with three components, i.e., an encoder, an attention or pooling layer and a classifier. 1) Encoder: An encoder reads a document, denoted by \( d = (w_1, w_2, ..., w_T) \), and transforms it to a sequence of hidden states \( H = (h_1, h_2, ..., h_T) \). Here, \( w_t \) is the one-hot representation of token \( t \) in the document. \( h_t \) is also known as a word-in-context representation. Traditionally, the encoder consists of a word embedding layer followed by a LSTM [18] sequence encoder. Recently, pre-trained language models [9, 34, 51] have emerged as an important component for achieving superior performance on a variety of NLP tasks including text classification. Our model is adaptable to any of these encoders. 2) Attention/Pooling: The attention or pooling (average- or max-pooling) layer is used to construct a high-level document representation, denoted by \( v^{\text{doc}} \). In attention networks, the attention weights show the contributions of words to the representations [26, 52]. Compared with pooling, attention operations can be well interpreted by visualizing attention weights [52]. 3) Classifier: The document representation is passed into a classifier to get the probability distribution over
Fig. 1. The proposed Abstraction-Aggregation Network and different interpretation methods.

different class labels. The classifier can be a multi-layer feed-forward network with activation layer followed by a softmax layer, i.e., $y = \text{softmax}(W_2 \cdot \text{ReLU}(W_1 \cdot d + b_1) + b_2)$, where $W_1, W_2, b_1$ and $b_2$ are model parameters.

To infer parameters, we can minimize the averaged cross-entropy error between predicted and ground-truth labels. Here, loss function is defined as $L_{\theta} = - \sum L_l \frac{1}{L} \hat{y} \log(y)$, where $\hat{y}$ represents the ground-truth label and $L$ is the number of class labels. The model is trained in an end-to-end manner using back-propagation.

3.1.2 Abstraction-Aggregation Network. In order to use different explanation methods, especially concept-based explanation, to interpret deep neural networks, we propose a novel AAN for the Attention/Pooling layer, which first captures keywords for different concepts from a document, and then aggregates all concepts to construct the document representation (see Fig. 1).

AAN has two stacked attention layers, namely, abstraction-attention (abs) and aggregation-attention (agg) layers. In the abs layer, for each attention unit $k$, we calculate the alignment score $u_{k,t}^{\text{abs}}$ and attention weight $\alpha_{k,t}^{\text{abs}}$ as follows:

$$u_{k,t}^{\text{abs}} = (g_k^{\text{abs}})^\top h_t,$$

$$\alpha_{k,t}^{\text{abs}} = \frac{\exp(u_{k,t}^{\text{abs}})}{\sum_{T=1}^{T} \exp(u_{k,t}^{\text{abs}})},$$

where $g_k^{\text{abs}}$ are model parameters. Here, we do not apply linear transformation and tanh activation when calculating alignment scores for two reasons: 1) **Better intuition**: Calculating attention between $g_k^{\text{abs}}$ and $h_t$ in Eq. (1) is the same as calculating a normalized similarity between them. Therefore, abstraction-attention can also be viewed as a clustering process, where $g_k^{\text{abs}}$ determines the centroid of each cluster. In our model, concepts are related to the clusters discovered by AAN.

2) **Fewer parameters**: Without the linear transformation layer, the abstraction-attention layer only introduces $K \times |h_t|$ new parameters, where $|h_t|$ is the dimension of $h_t$ and $K \ll |h_t|$. The $k^{th}$
representation is obtained by \( \nu_{abs}^k = \sum_{t=1}^T \alpha_{abs}^k t \). We use \( K \) to denote the total number of attention units.

In the agg layer, there is only one attention unit. The alignment score \( u_{agg}^k \) and attention weight \( \alpha_{agg}^k \) are obtained by

\[
\begin{align*}
    u_{agg}^k &= (g_{agg})^\top \tanh(W_{agg} \nu_{abs}^k + b_{agg}), \\
    \alpha_{agg}^k &= \exp(u_{agg}^k) \frac{1}{\sum_{k=1}^K \alpha_{agg}^k},
\end{align*}
\]

where \( W_{agg}, b_{agg} \) and \( g_{agg} \) are model parameters. The final document representation is obtained by

\[
\nu_{doc} = \sum_{k=1}^K \alpha_{agg}^k \nu_{abs}^k,
\]

and it should be noted that AAN is different from hierarchical attention [52], which aims to get a better representation. However, AAN is used to automatically capture concepts/clusters.

We have also applied two important techniques to obtain semantically meaningful concepts.

1) Diversity penalty for abstraction-attention weights: To encourage the diversity of concepts, we introduce a new penalization term to abstraction-attention weights \( A = [\alpha_{abs}^1, \alpha_{abs}^2, ..., \alpha_{abs}^K] \in \mathbb{R}^{T \times K} \), where \( \alpha_{abs}^k = (\alpha_{abs}^k, \alpha_{abs}^k_2, ..., \alpha_{abs}^k_T)^\top \). We define the penalty function as

\[
\mathcal{L}_{div} = \frac{1}{K} \| A^\top A - I \|_F,
\]

where \( \| \cdot \|_F \) represents the Frobenius norm of a matrix. Hence, the overall loss function is expressed as \( \mathcal{L} = \mathcal{L}_\theta + \mathcal{L}_{div} \).

2) Dropout of aggregation-attention weights: In the aggregation-attention layer, it is possible that \( \alpha_{agg}^k \approx 1 \) for some \( k \), and other attention weights tend to be 0. To alleviate this problem, we apply dropout with a small dropout rate to aggregation-attention weights \( (\alpha_{agg}^1, \alpha_{agg}^2, ..., \alpha_{agg}^K) \), namely attention weights dropout. It should be noted that a large dropout rate has a negative impact on the explanation, since it discourages the diversity of concepts. More specifically, the model will try to capture keywords in the dropped abstraction-attention units by the other units.

3.2 Explanation

In this section, we discuss corpus-level and concept-based explanations. Given a corpus \( C \) with \(|C|\) documents, we use \( d \) or \( \xi \) to represent a document. Let us also use \( \theta \) to denote all parameters of a model and \( \mathcal{V} \) to represent the vocabulary, where \(|\mathcal{V}|\) is the size of \( \mathcal{V} \). Throughout this paper, we will assume that both prior document probability \( p(d) \) and prior label probability \( p(\theta|y=l) \) are constants. For example, in a label-balanced dataset, \( p(\theta|y=l) \approx 1/K \).

We will first apply the attention weights visualization technique to the proposed AAN model. Here, the document representation can be directly expressed by the hidden states, i.e.,

\[
\nu_{agg} = \sum_{t=1}^T \sum_{k=1}^K \alpha_{agg}^k \alpha_{abs}^k t h_t,
\]

where

\[
\alpha_{agg}^d = \sum_{k=1}^K \alpha_{agg}^k \alpha_{abs}^k
\]

gives the contribution of word \( w_t \) to the document representation. Therefore, we can interpret any single example via visualizing the combined weights \( \alpha_{agg}^d \).
3.2.1 Corpus-Level Explanation. Different from case-level explanation (attention weights visualization), which focuses on per-sample features, corpus-level explanation aims to find causal relationships between keywords captured by the attention mechanism and model predictions, which can provide robust explanation for the model. To achieve this goal, we learn distributions of keywords for different predicted labels on the training corpus based on attention weights.

Formally, for a given word \( w \in V \) and a label \( l \) predicted by a model \( \theta \), the importance of the word to the label can be estimated by the probability \( p_\theta(w|y = l) \) across the training corpus \( C_{\text{train}} \) since the model is trained on it. Therefore, \( p_\theta(w|y = l) \) can be expanded as follows:

\[
p_\theta(w|y = l) = \sum_{\xi \in C_{\text{train}}^l} p_\theta(w, \xi|y = l),
\]

where \( C_{\text{train}}^l \subset C_{\text{train}} \) consists of documents with model predicted label \( l \). For each document \( \xi \in C_{\text{train}}^l \), probability \( p_\theta(w, \xi|y = l) \) represents the importance of word \( w \) to label \( l \), which can be defined using attention weights, i.e.,

\[
p_\theta(w, \xi|y = l) := \frac{\sum_{t=1}^{T} \alpha_t^{\xi} \cdot \delta(w_t, w)}{\sum_{\xi' \in C_{\text{train}}} f_{\xi'}(w) + \gamma},
\]

where \( f_{\xi'}(w_t) \) is frequency of \( w_t \) in document \( \xi' \) and \( \gamma \) is a smoothing factor. \( \delta(w_t, w) = \begin{cases} 1 & \text{if } w_t = w \\ 0 & \text{otherwise} \end{cases} \)

is a delta function. The denominator is applied to reduce noises from stop-words and punctuation. For the sake of simplicity, we will use \( p_\theta(w, l, C) \) to denote \( p_\theta(w_t|y = l) \), where \( C \) corresponds to the corpus Eq. (4), and can be different from \( C_{\text{train}} \) in our applications. The denominator in Eq. (5) is always determined by the training corpus.

With respect to the applications: 1) Since Eq. (4) captures the importance of words to model predicted labels, we can use it as a criterion for finding their causal relationships. In experiments, we can collect top-ranked keywords for each label \( l \) for further analysis. 2) We can also use corpus-level explanation to measure the difference between two corpora (i.e., \( C_{\text{test1}} \) and \( C_{\text{test2}} \)). Formally, we can compare \( \frac{|C_{\text{train}}|}{|C_{\text{test1}}|} \cdot p_\theta(w, l, C_{\text{test1}}) \) with \( \frac{|C_{\text{train}}|}{|C_{\text{test2}}|} \cdot p_\theta(w, l, C_{\text{test2}}) \) across different words and class labels. The difference can be evaluated by Kullback-Leibler divergence [25]. In addition, we can get mutual keywords shared across different domains based on these distributions.

It should be noted that the corpus-level explanation discussed in this section can be applied to interpret different attention-based networks.

3.2.2 Concept-Based Explanation. The corpus-level explanation still suffers from the drawback that it cannot automatically obtain higher-level concepts/clusters for those important keywords. To alleviate this problem, we propose concept-based explanation for our AAN model. In AAN, each abstraction-attention unit can capture one concept/cluster. Here, we will take distribution of concepts into consideration. Formally, we express \( p_\theta(w_t|y = l) \) as follows:

\[
p_\theta(w|y = l) = \sum_{k=1}^{K} p_\theta(w|c_k, y = l) p_\theta(c_k|y = l)
\]

where \( p_\theta(w|c_k, y = l) \) captures the distribution of \( w \) across \( C_{\text{train}} \) for the \( k \)th concept and label \( l \), while \( p_\theta(c_k|y = l) \) captures the distribution of the concept \( c_k \) across \( C_{\text{train}} \) for label \( l \). They can be computed using the following equations.

\[\text{Here, the label is the model’s prediction, not the ground-truth label, because our goal is to explain the model.}\]
Corpus-level and Concept-based Explanation

\[ p_\theta(w|c_k, y = l) = \sum_{\xi \in C_{\text{train}}} p_\theta(w, \xi|c_k, y = l), \]

\[ p_\theta(c_k|y = l) = \sum_{\xi \in C_{\text{train}}} p_\theta(c_k, \xi|y = l), \]  \hspace{1cm} (6)

where we define

\[ p_\theta(w, \xi|c_k, y = l) := \frac{\sum_{t=1}^{T} \alpha_{k,t}^{\text{abs},\xi} \cdot \delta(w_t, w)}{\sum_{\xi' \in C_{\text{train}}} f_{\xi'}(w) + \gamma} \]  \hspace{1cm} (7)

and

\[ p_\theta(c_k, \xi|y = l) := \frac{\alpha_{k}\xi^{\text{agg},\xi}}{|C_{\text{train}}|}, \]  \hspace{1cm} (8)

where \( \alpha_{k,t}^{\text{abs},\xi} \) represents \( \alpha_{k,t}^{\text{abs}} \) for document \( \xi \). Based on Eq. (6), we are able to obtain scores (importance) and most relevant keywords for different concepts for a given label \( l \).

3.2.3 Consistency Analysis. In corpus-level and concept-based explanations, we have obtained causal relationships between keywords and predictions, i.e., \( p_\theta(w|y = l) \). However, we have not verified if these keywords are really important to predictions. To achieve this goal, we build a Naïve Bayes classifier (NBC) [10] based on these causal relationships. Formally, for each testing document \( d \), the probability of getting label \( l \) is approximated as follows:

\[ p_\theta(y = l|d) = \frac{p_\theta(d|y = l)p_\theta(y = l)}{p(d)} \]  \hspace{1cm} (9)

\[ \propto p_\theta(d|y = l) = \prod_{t=1}^{T} p_\theta(w_t|y = l), \]

where \( p_\theta(w_t|y = l) \) is obtained by Eq. (4) or Eq. (6) on the training corpus. We further approximate Eq. (9) with

\[ p_\theta(y = l|d) = \prod_{w \in d'} (p_\theta(w|y = l) + \lambda), \]  \hspace{1cm} (10)

where \( d' \subset d \) is an attention-based bag-of-words representation for document \( d \). It consists of important keywords based on attention weights. \( \lambda \) is a smoothing factor. Here, we can conduct consistency analysis by comparing labels obtained by the model and NBC, which may also help estimate the uncertainty of a model [55].

4 EXPERIMENTS

4.1 Datasets

We conducted experiments on three publicly available datasets. Newsroom is used for news categorization, while IMDB and Beauty are used for sentiment analysis. The details of the three datasets are as follows: 1) Newsroom [15]: The original dataset, which consists of 1.3 million news articles, was proposed for text summarization. In our experiments, we first determined the category of each article based on the URL, and then, randomly sample 10,000 articles for each of the five categories, including business, entertainment, sports, health, and technology. 2) IMDB [31]: This dataset contains 50,000 movie reviews from the IMDB website with binary (positive or negative) labels. 3) Beauty [16]: This dataset contains product reviews in the beauty category from Amazon. We converted the original ratings (1-5) to binary (positive or negative) labels and sampled 20,000 reviews for each label. For all three datasets, we tokenized reviews using the BERT tokenizer [49]
Table 1. Statistics of the datasets used.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#docs</th>
<th>Avg. Length</th>
<th>Scale</th>
</tr>
</thead>
<tbody>
<tr>
<td>Newsroom</td>
<td>50,000</td>
<td>827</td>
<td>1-5</td>
</tr>
<tr>
<td>IMDB</td>
<td>50,000</td>
<td>292</td>
<td>1-2</td>
</tr>
<tr>
<td>Beauty</td>
<td>40,000</td>
<td>91</td>
<td>1-2</td>
</tr>
</tbody>
</table>

Table 2. Averaged accuracy of different models on Newsroom, IMDB, and Beauty testing sets.

<table>
<thead>
<tr>
<th>Model</th>
<th>Newsroom</th>
<th>IMDB</th>
<th>Beauty</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNN</td>
<td>90.18</td>
<td>88.56</td>
<td>88.42</td>
</tr>
<tr>
<td>LSTM-SAN</td>
<td>91.26</td>
<td>90.68</td>
<td>92.00</td>
</tr>
<tr>
<td>BERT-SAN</td>
<td>92.28</td>
<td>92.60</td>
<td>93.72</td>
</tr>
<tr>
<td>DistilBERT-SAN</td>
<td>92.66</td>
<td>92.52</td>
<td>92.82</td>
</tr>
<tr>
<td>RoBERTa-SAN</td>
<td>91.16</td>
<td>92.76</td>
<td>93.40</td>
</tr>
<tr>
<td>Longformer-SAN</td>
<td>92.04</td>
<td>93.74</td>
<td>94.50</td>
</tr>
</tbody>
</table>

and randomly split them into train/development/test sets with a proportion of 8/1/1. Statistics of the datasets are summarized in Table 1.

4.2 Models and Implementation Details

We compare different classification models including several baselines, variants of our AAN model, and Naïve Bayes classifiers driven by a basic self-attention network (SAN) [38] and AAN.

- **CNN** [22]: This model extracts key features from a review by applying convolution and max-over-time pooling operations [7] over the shared word embedding layer.

- **LSTM-SAN, BERT-SAN, DistilBERT-SAN, RoBERTa-SAN, and Longformer-SAN**: All these models are based on the SAN framework. In LSTM-SAN, the encoder consists of a word embedding layer and a Bi-LSTM encoding layer, where embeddings are pre-loaded with 300-dimensional GloVe vectors [33] and fixed during training. BERT [49], DistilBERT [37], RoBERTa [28], and Longformer [3] leverage different pre-trained language models, which have 110M, 66M, 125M, and 125M parameters, respectively.

- **AAN + C(𝑐) + Drop(𝑟)**: These are variants of AAN. C(𝑐) and Drop(𝑟) represent the number of concepts and dropout rate, respectively.

We implemented all deep learning models using PyTorch [32] and the best set of parameters are selected based on the development set. For CNN based models, the filter sizes are chosen to be 3, 4 and 5, and the number of filters is set to 100 for each size. For LSTM based models, the dimension of hidden states is set to 300 and the number of layers is 2. All parameters are trained with the ADAM optimizer [24] with a learning rate of 0.0002. Dropout with a rate of 0.1 is also applied in the classification layer. For all explanation tasks, we set the number of concepts to 10 and dropout-rate to 0.02. Our codes and datasets are available at https://github.com/tshi04/ACCE.

4.3 Performance Results

We use accuracy as the evaluation metric to measure the performance of different models. All quantitative results have been summarized in Tables 2 and 3, where we use bold font to highlight the highest accuracy on testing sets in Table 2. Comparing LSTM-SAN with BERT, DistilBERT, RoBERTa and Longformer, we first find that different pre-trained language model-based encoders are better than the conventional LSTM encoder with pre-trained word embeddings. In Table 3, we replace self-attention on top of pre-trained language models with the abstraction-aggregation
Table 3. Averaged accuracy of BERT and Longformer-based AAN models on Newsroom, IMDB, and Beauty testing sets.

<table>
<thead>
<tr>
<th></th>
<th>Newsroom BERT</th>
<th>Newsroom Longformer</th>
<th>IMDB BERT</th>
<th>IMDB Longformer</th>
<th>Beauty BERT</th>
<th>Beauty Longformer</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAN Framework</td>
<td>92.28</td>
<td>92.04</td>
<td>92.60</td>
<td>93.74</td>
<td>93.72</td>
<td>94.50</td>
</tr>
<tr>
<td>AAN + C(10) + Drop(0.01)</td>
<td>92.54</td>
<td>91.72</td>
<td>92.22</td>
<td>92.96</td>
<td>93.38</td>
<td>93.42</td>
</tr>
<tr>
<td>AAN + C(10) + Drop(0.02)</td>
<td>92.14</td>
<td>91.64</td>
<td>92.14</td>
<td>92.86</td>
<td>93.58</td>
<td>93.75</td>
</tr>
<tr>
<td>AAN + C(10) + Drop(0.05)</td>
<td>92.14</td>
<td>91.60</td>
<td>91.82</td>
<td>92.66</td>
<td>93.05</td>
<td>93.80</td>
</tr>
<tr>
<td>AAN + C(20) + Drop(0.10)</td>
<td>92.30</td>
<td>91.48</td>
<td>91.50</td>
<td>92.12</td>
<td>93.25</td>
<td>93.60</td>
</tr>
<tr>
<td>AAN + C(20) + Drop(0.01)</td>
<td>92.02</td>
<td>91.98</td>
<td>91.64</td>
<td>92.78</td>
<td>93.70</td>
<td>93.48</td>
</tr>
<tr>
<td>AAN + C(20) + Drop(0.02)</td>
<td>92.44</td>
<td>91.84</td>
<td>91.80</td>
<td>93.04</td>
<td>93.55</td>
<td>93.88</td>
</tr>
<tr>
<td>AAN + C(20) + Drop(0.05)</td>
<td>92.54</td>
<td>91.86</td>
<td>91.92</td>
<td>93.14</td>
<td>93.68</td>
<td>93.42</td>
</tr>
<tr>
<td>AAN + C(20) + Drop(0.10)</td>
<td>92.52</td>
<td>91.98</td>
<td>92.10</td>
<td>92.96</td>
<td>93.72</td>
<td>93.88</td>
</tr>
</tbody>
</table>

Table 4. Case-level concept-based explanation. Here, each ID is associated with a concept, i.e., abstraction-attention unit. Scores and weights (following each keyword) are calculated with Eqs. (7) and (8). ‘-’ represents special characters.

<table>
<thead>
<tr>
<th>ID</th>
<th>Score</th>
<th>Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>0.180</td>
<td>com(0.27), boston(0.26), boston(0.16), boston(0.1), m(0.02)</td>
</tr>
<tr>
<td>6</td>
<td>0.162</td>
<td>marketing(0.28), ad(0.06), ##fs(0.05), investors(0.03), said(0.03)</td>
</tr>
<tr>
<td>1</td>
<td>0.148</td>
<td>campaign(0.14), firm(0.14), money(0.06), brand(0.04), economist(0.03)</td>
</tr>
<tr>
<td>2</td>
<td>0.122</td>
<td>economist(0.2), said(0.16), professional(0.16), investors(0.08), agency(0.06)</td>
</tr>
<tr>
<td>9</td>
<td>0.116</td>
<td>boston(0.89), boston(0.11)</td>
</tr>
<tr>
<td>7</td>
<td>0.108</td>
<td>bloomberg(0.16), cni(0.11), global(0.09), money(0.06), cable(0.05)</td>
</tr>
<tr>
<td>5</td>
<td>0.103</td>
<td>-(0.96), -(0.03), s(0.01)</td>
</tr>
<tr>
<td>4</td>
<td>0.047</td>
<td>investment(0.76), money(0.14), investment(0.06), investment(0.02), investors(0.01)</td>
</tr>
<tr>
<td>3</td>
<td>0.016</td>
<td>, (0.64), -(0.36)</td>
</tr>
<tr>
<td>10</td>
<td>0.000</td>
<td>, (0.93), -(0.07)</td>
</tr>
</tbody>
</table>

network (AAN). We observe that different AAN models do not significantly lower the classification accuracy, which indicates we can use AAN for the concept-based explanation task without losing the overall performance. Here, the strategy of aggregation-attention weights dropout is necessary when training AAN models. In Table 9, we show that AAN models without randomly dropping aggregation-attention weights attain poor interpretability in concept-based explanation.

4.4 Heat-maps and Case-level Concept-based Explanation

First, we investigate if AAN attends to relevant keywords when it is making predictions, which can be accomplished by visualizing attention weights (see Fig. 2). This is a Business news article from Newsroom and we observe that the most relevant keyword that AAN detects is boston. Other important keywords include investment, economist, marketing and com. Compared with Fig. 2, our case-level concept-based explanation provides more informative results. From Table 4, we observe that AAN makes the prediction based on several different aspects, such as corporations (e.g., com), occupations (e.g., economist), terminology (e.g., marketing) and so on. Moreover, boston may be
Fig. 2. Attention-weight visualization for an interpretable attention-based classification model.

Table 5. This table shows 20 most important keywords for model predictions on different training sets. Keywords are ordered by their scores. For IMDB, we use bold to highlight the common keywords shared with Beauty.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Label</th>
<th>Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMDB</td>
<td>Negative</td>
<td>worst, awful, terrible, bad, <strong>disappointed</strong>, boring, <strong>disappointing</strong>, waste, horrible, sucks, fails, disappointment, lame, dull, poorly, poor, worse, mess, dreadful, pointless</td>
</tr>
<tr>
<td></td>
<td>Positive</td>
<td>8, 7, <strong>excellent, loved</strong>, 9, enjoyable, superb, enjoyed, <strong>highly, wonderful</strong>, entertaining, best, beautifully, good, <strong>great</strong>, brilliant, terrific, funny, hilarious, fine</td>
</tr>
<tr>
<td>Beauty</td>
<td>Negative</td>
<td>disappointed, nothing, unfortunately, made, not, waste, disappointing, terrible, worst, horrible, makes, no, sadly, disappointment, t, awful, sad, bad, never, started</td>
</tr>
<tr>
<td></td>
<td>Positive</td>
<td>great, love, highly, amazing, pleased, perfect, works, best, happy, awesome, makes, recommend, excellent, wonderful, definitely, good, glad, well, fantastic, very</td>
</tr>
<tr>
<td>Business</td>
<td>inc, corp, boston, massachusetts, economic, cambridge, financial, economy, banking, auto, automotive, startup, company, mr, finance, biotechnology, somerville, retailer, business, airline</td>
<td></td>
</tr>
<tr>
<td>Entertainment</td>
<td>singer, actress, actor, star, fox, comedian, hollywood, sunday, rapper, fashion, celebrity, contestant, filmmaker, bachelor, insider, porn, oscar, rocker, host, monday</td>
<td></td>
</tr>
<tr>
<td>Sports</td>
<td>quarterback, coach, basketball, baseball, soccer, nba, sports, striker, tennis, hockey, nll, nhl, football, olympic, midfielder, golf, player, manager, outfielder, nascar</td>
<td></td>
</tr>
<tr>
<td>Health</td>
<td>dr, health, pediatric, obesity, cardiovascular, scientists, researcher, medicine, psychologist, diabetes, medical, psychiatry, aids, fitness, healthcare, autism, psychology, neuroscience, fox, tobacco</td>
<td></td>
</tr>
<tr>
<td>Technology</td>
<td>tech, cyber, electronics, wireless, lifestyle, silicon, gaming, culture, telecommunications, scientist, company, google, smartphone, technology, francisco, broadband, privacy, internet, twitter</td>
<td></td>
</tr>
</tbody>
</table>

related with corporation (e.g., bostonglobe or gerritsen of boston) or city, thus, it appears in both concepts 8 (corporations) and 9 (locations).
Corpus-level and Concept-based Explanation

### 4.5 Corpus-Level Explanation

Corpus-level explanation aims to find the important keywords for the predictions. In Table 5, we show 20 most important keywords for each predicted label and we assume these keywords determine the predictions. In the last section, we will demonstrate this assumption by the consistency analysis. The scores of keywords have been shown in Fig. 3.

In addition to causal relationships, we can also use these keywords to check if our model and datasets have bias or not. For example, *boston* and *massachusetts* play an important role in predicting business, which indicates the training set has bias. By checking our data, we find that many business news articles are from *The Boston Globe*. Another obvious bias example is that the numbers 8, 7 and 9 are important keywords for IMDB sentiment analysis. This is because the original ratings scale from 1 to 10 and many reviews mention that “rate this movie 8 out of 10”.

Moreover, from Figs. 3 (a) and (b), we find that for a randomly split corpus, distributions of keywords across training/development/test sets are similar to each other. This guarantees the model achieves outstanding performance on testing sets. If we apply a model trained on IMDB to Beauty (see Fig. 3 (c)), it can only leverage the cross-domain common keywords (e.g., *disappointed*).
Table 6. Concept-based explanation (Business). Scores are calculated using Eq. (6).

<table>
<thead>
<tr>
<th>ID</th>
<th>Score</th>
<th>Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>0.173</td>
<td>inc, corp, massachusetts, boston, mr, ms, jr, ltd, mit, q</td>
</tr>
<tr>
<td>1</td>
<td>0.168</td>
<td>economy, retailer, company, startup, #maker, airline, chain, bank, utility, billionaire</td>
</tr>
<tr>
<td>7</td>
<td>0.151</td>
<td>biotechnology, banking, tech, startup, pharmaceuticals, mortgage, financial, auto, commerce, economic</td>
</tr>
<tr>
<td>6</td>
<td>0.124</td>
<td>economic, health, banking, finance, insurance, healthcare, economy, housing, safety, commerce</td>
</tr>
<tr>
<td>4</td>
<td>0.107</td>
<td>financial, economic, banking, auto, automotive, securities, housing, finance, monetary, biotechnology</td>
</tr>
<tr>
<td>9</td>
<td>0.086</td>
<td>boston, massachusetts, cambridge, washington, detroit, frankfurt, harvard, tokyo, providence, paris</td>
</tr>
<tr>
<td>5</td>
<td>0.056</td>
<td>-, #as, -, -, itunes, inc, corp, northeast, -, llc</td>
</tr>
<tr>
<td>2</td>
<td>0.054</td>
<td>economist, executive, spokesman, analyst, economists, #gist, ceo, director, analysts, president</td>
</tr>
<tr>
<td>3</td>
<td>0.026</td>
<td>-, -, -, #item, #sp, the, =, t, ob</td>
</tr>
<tr>
<td>10</td>
<td>0.000</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 7. Concept-based explanation (Sports). Scores are calculated using Eq. (6).

<table>
<thead>
<tr>
<th>ID</th>
<th>Score</th>
<th>Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.176</td>
<td>quarterback, player, striker, champion, pitcher, midfielder, outfilder, athlete, goaltender, forward</td>
</tr>
<tr>
<td>7</td>
<td>0.147</td>
<td>nhl, mets, soccer, nets, yankees, nascar, mls, reuters, doping, twitter</td>
</tr>
<tr>
<td>6</td>
<td>0.139</td>
<td>tennis, sports, soccer, golf, doping, hockey, athletic, athletics, injuries, basketball</td>
</tr>
<tr>
<td>4</td>
<td>0.119</td>
<td>baseball, basketball, nba, nhl, sports, football, tennis, olympic, hockey, golf</td>
</tr>
<tr>
<td>8</td>
<td>0.119</td>
<td>jr, ', a, j, fox, espn, nl, u, boston, ca</td>
</tr>
<tr>
<td>2</td>
<td>0.100</td>
<td>coach, manager, commissioner, boss, gm, trainer, spokesman, umpire, coordinator, referee</td>
</tr>
<tr>
<td>9</td>
<td>0.060</td>
<td>philadelphia, indianapolis, boston, tampa, louisville, buffalo, melbourne, manchester, baltimore, atlanta</td>
</tr>
<tr>
<td>5</td>
<td>0.055</td>
<td>', ', #as, - ', #a, ' sides, newcomers, chelsea, jaguars</td>
</tr>
<tr>
<td>3</td>
<td>0.022</td>
<td>', ', ), #kus, #gre, the, whole, lever, #wa</td>
</tr>
<tr>
<td>10</td>
<td>0.000</td>
<td>, '), finishes, bel, gymnastics, ', #ditional, becomes, tu, united</td>
</tr>
</tbody>
</table>

and loved) to make predictions. However, we achieve 71% accuracy, which is much better than random predictions. In Table 5, we use bold font to highlight these common keywords.

### 4.6 Corpus-level Concept-based Explanation

Corpus-level concept-based explanation further improves the corpus-level explanation by introducing clustering structures to keywords. In this section, we still use the AAN trained on Newsroom as an example for this task. Table 6 shows concepts and relevant keywords for AAN when it assigns an article to Business. Here, we observe that the first-tier salient concepts consist of concepts 8 (corporations) and 1 (business terminology in general). The second-tier concepts 7, 6, and 4 are related to economy, finance, mortgage, and banking, which are domain-specific terminology. They share many keywords. Concepts 9 and 2 are associated with locations and occupations, respectively, which receive relatively lower scores. Concepts 5, 3, and 10 are not quite meaningful. We have also shown results for Newsroom sports in Table 7, where we find that 1 (sports terminology) and 7 (leagues and teams) are the first-tier salient concepts. The second-tier salient concepts 6 and 4 are about games and campaigns. Concepts 7, 6, and 4 also share many keywords. Concepts 8 (corporations and channels), 2 (occurrences and roles) and 9 (locations) are the third-tier salient concepts. Concepts 5, 3, 10 are also meaningless. From these tables, we summarize some commonalities: 1) Domain-specific terminologies (i.e., concepts 1, 7, 6, and 4) play an important role in predictions. 2) Locations (i.e., concept 9) and Occupations/Roles (i.e., concept 2) are less important. 3) Meaningless concepts (i.e., concepts 5, 3, and 10), such as punctuation, have the least influence.
Table 8. Consistency between the model and NBC. CS represents consistency score, CP/NCP denote percentage of incorrect predictions when NBC predictions are consistent/not consistent with model predictions.

<table>
<thead>
<tr>
<th>Model</th>
<th>Newsroom</th>
<th>IMDB</th>
<th>Beauty</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CS</td>
<td>NCP</td>
<td>CP</td>
</tr>
<tr>
<td>BERT-SAN</td>
<td>83.96</td>
<td>21.59</td>
<td>4.72</td>
</tr>
<tr>
<td>BERT-AAN</td>
<td>84.36</td>
<td>20.20</td>
<td>5.57</td>
</tr>
</tbody>
</table>

Table 9. Concept-based explanation (Sports) for AAN without applying dropout to attention weights.

<table>
<thead>
<tr>
<th>CID</th>
<th>Weight</th>
<th>Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.8195</td>
<td>quarterback, athletic, olympic, basketball, athletics, qb, hockey, outfielder, sports</td>
</tr>
<tr>
<td>2</td>
<td>0.0865</td>
<td>nascar, celtics, motorsports, nba, boston, augusta, nhl, tennis, leafs, zurich</td>
</tr>
<tr>
<td>3</td>
<td>0.0370</td>
<td>offensive, eli, bird, doping, nba, jay, rod, hurdle, afc, peyton</td>
</tr>
<tr>
<td>4</td>
<td>0.0098</td>
<td>premier, american, mets, nl, field, yankee, national, aaron, nba, olympic</td>
</tr>
<tr>
<td>5</td>
<td>0.0015</td>
<td>dustin, antonio, rookie, dante, dale, dylan, lineman, ty, launch, luther</td>
</tr>
<tr>
<td>6</td>
<td>0.0000</td>
<td>the, -, -, -, year, whole, vie, very, tr, too, to</td>
</tr>
</tbody>
</table>

### 4.7 Consistency Analysis

In this section, we leverage the method proposed in Section 3.2.3 to respectively build a NBC for BERT-SAN and BERT-AAN on the training set. Then, we apply them to the testing set to compare if NBC predictions and the model predictions are consistent with each other. We approximate the numerator of Eq. (5) with five words (can repeat) with highest attention weights in each document. In Eq. (4), $\gamma$ is set to be 1000. In Eq. (10), we set $\lambda = 1.2$ for text categorization and $\lambda = 1.0$ for sentiment analysis. $d'$ consists of five words with highest attention weights.

We use the accuracy (consistency score) between labels predicted by NBC and the original model to evaluate the consistency. Table 8 shows that around 85% of predictions are consistent. This demonstrates that keywords obtained by the corpus-level and concept-based explanation methods are important to predictions. They can be used to interpret attention based models. Moreover, from CP and NCP scores, we observe a significantly higher probability that the model makes an incorrect prediction if it is inconsistent with NBC prediction. This finding suggests us to use consistency score as one criterion for uncertainty estimation.

### 4.8 Dropout of Aggregation-Attention Weights

For AAN, we apply dropout to aggregation-attention weights during training. In Table 9, we show an example without using the attention weight dropout mechanism. We observed that the weight for concept 1 is much higher than the other concepts. In addition, keywords for each concept are not semantically coherent.

### 5 CONCLUSION

In this paper, we proposed a general-purpose corpus-level explanation approach to interpret attention-based networks. It can capture causal relationships between keywords and model predictions via learning importance of keywords for predicted labels across the training corpus based on attention weights. Experimental results show that the keywords are semantically meaningful for predicted
labels. We further propose a **concept-based explanation** method to identify important concepts for model predictions. This method is based on a novel *Abstraction-Aggregation Network* (AAN), which can automatically extract concepts, i.e., clusters of keywords, during the end-to-end training for the main task. Our experimental results also demonstrate that this method effectively captures semantically meaningful concepts/clusters. It also provides relative importance of each concept to model predictions. To verify our results, we also built a *Naïve Bayes Classifier* based on an **attention-based bag-of-word document representation** technique and the causal relationships. Our consistency analysis results demonstrate that the discovered keywords are important to the predictions.

**ACKNOWLEDGMENTS**

This work was supported in part by the US National Science Foundation grants IIS-1707498, IIS-1838730, and Amazon AWS credits.

**REFERENCES**


