AILA: Interactive Document Labeling Assistant for Document Classification through Attention-based Deep Neural Networks

Minsuk Choi[1], Cheonbok Park[1], Soyoung Yang[1], Yonggyu Kim[2], Jaegul Choo[1], and Sungsoo (Ray) Hong[3]
[1] Korea University, Seoul, Republic of Korea, {mchoi, cb_park, dudrrm, jchoo}@korea.ac.kr
[2] Ajou University, Suwon, Republic of Korea, ygkim2535@gmail.com
[3] University of Washington, dub group, Seattle, WA, USA, rayhong@uw.edu

ABSTRACT

Document labeling is a critical step in building various machine learning applications. However, the step can be time-consuming and arduous, requiring a significant amount of human effort. To support an efficient document labeling environment, we present a system called *Attentive Interactive Labeling Assistant* (AILA). At its core, AILA uses *Interactive Attention Module* (IAM), a novel module that visually highlights words in a document that labelers may pay attention to when labeling a document. IAM utilizes *attention-based Deep Neural Networks*, which not only support a prediction of which words to highlight, but also enable labelers to indicate words that should be assigned high attention weights while labeling to improve the future quality of word prediction. We evaluated the labeling efficiency and accuracy by comparing the conditions with and without IAM in our study. The results showed that the participants' labeling efficiency increased significantly under the condition with IAM than under the condition without IAM, while the two conditions maintained roughly the same labeling accuracy.

CCS CONCEPTS

• **Human-centered computing → Visualization systems and tools**; • **Computing methodologies → Information extraction**;

KEYWORDS

Document labeling, document classification, natural language processing, attention model, deep neural networks

CHI 2019, May 4–9, 2019, Glasgow, Scotland UK

ACM ISBN 978-1-4503-5970-2/19/05…$15.00
https://doi.org/10.1145/3290605.3300460

ACM Reference Format:
Minsuk Choi, Cheonbok Park, Soyoung Yang, Yonggyu Kim, Jaegul Choo, and Sungsoo (Ray) Hong. 2019. AILA: Interactive Document Labeling Assistant for Document Classification through Attention-based Deep Neural Networks. In *CHI Conference on Human Factors in Computing Systems Proceedings (CHI 2019), May 4–9, 2019, Glasgow, Scotland UK*. ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3290605.3300460

1 INTRODUCTION

Acquiring an accurately labeled text corpus is necessary for training machine learning (ML) models in various text classification applications such as spam detection [6, 18], fake news filtering [2, 8], and sentiment analysis [29, 41]. It is often the case that the labeled dataset is not available, which necessitates researchers and practitioners to build their own datasets. However, it was previously identified that designing user interfaces (UIs) for motivating labelers' efficient and accurate labeling entails substantial challenges [7, 21]. For instance, although crowdsourcing could be a powerful platform in collecting labeled data at scale [11], existing studies have discussed the difficulties of enabling workers to maintain their attention [40] and/or coping with a crowd's inconsistent performances regarding labeling accuracy [26]. In addition, supporting labeling tasks that require domain knowledge, such as labeling incorrect statements in a legal document or labeling a particular disease name given a person's medical record, poses new challenges. There exist pressing needs in understanding better design for motivating labelers' labeling performances [4, 9, 21, 34], but such a domain is relatively under-explored.

Motivated by these challenges, we present an *Attentive Interactive Labeling Assistant*, or in short, AILA, which supports efficient human labeling. For our initial efforts, AILA's scope of the target task is a document classification problem, especially a binary classification of relatively short documents (e.g., a document with a few sentences). Labeling each document as positive or negative with respect to a particular criterion, such as positive/negative sentiment, is a basic classification task, but our design can be applied in different labeling tasks such as ordinal or categorical classification

tasks. In general, AILA works as follows: Once a labeler assigns particular labels on a small number of documents, AILA's back-end ML model is trained using the results. Next, AILA performs the prediction on the remaining documents and interactively recommends a subset of the documents so that a labeler can label relevant documents sequentially.

Our back-end ML model is called an *Interactive Attention Module* (IAM). IAM adopts Recurrent Neural Networks (RNNs), a type of Deep Neural Networks (DNN) that is widely used for text classification [36]. IAM leverages the *attention mechanism* [42], which has shown its effectiveness in numerous DNN architectures, to support efficient labeling environment via two different approaches. First, IAM utilizes the attention mechanism to visually emphasize words of a given document that the model predicts to be useful for labeling. Second, the attention mechanism works not only as a means for labelers to check the prediction outcome from the model, but also as a handle for labelers to steer the model; IAM allows labelers to highlight keywords important for labeling documents and updates the model accordingly for future prediction. The importance of interactive ML that involves humans in training data and correcting the classification has been extensively discussed in previous works [12, 35].

To confirm the effectiveness of IAM in supporting a user's labeling task, we conducted a within-subject study between the condition that adopts IAM and the condition that does not. Our findings indicate that participants' labeling task was significantly accelerated with IAM than without IAM. Meanwhile, there was no significant difference in terms of the labeling accuracy between the two conditions. Additionally, in the usability study, we found that participants perceived our system as a useful tool that helps them label documents with much less effort.

In summary, we offer the following contributions:

- **Technical contributions**: we present IAM, a novel module that couples human labelers and ML models to accurately predict words that are important for labeling documents and then visually emphasizes the words.
- **System contributions**: we propose AILA, a system that adopts IAM and orders documents in a coherent manner to present an efficient document labeling environment.
- **Empirical contributions**: we report our study results on the expected effect of IAM and the advantages and disadvantages of using AILA.

2 RELATED WORK

We review the existing ML models that focus on text classification. Next, we discuss related work on the UI design for supporting labelers' text classification tasks.

Text classification models using attention mechanism

In recent years, computational performances for text classification have been greatly improved as neural models have evolved. Such models include Convolutional Neural Networks (CNNs)
[19, 32, 44] and long short-term memory (LSTM), which is known to be a successful architecture of RNNs [14, 38]. In particular, the studies show that classification accuracy significantly improved when the model used *word embedding*, a technique that represents a word as a high-dimensional vector, along with an *attention mechanism*, which allows the model to capture important parts of the data [3].

Researchers have applied the attention mechanism to build text representation models for prediction tasks. For instance, Ling et al. proposed an attention-based model to perform word embedding [24], but their approach mainly focused on a word level of prediction. Yang et al. utilized an attention mechanism for text classification at both word and sentence levels [43]. An advanced self-attention model has taken into account the semantic and syntactic relationships of words within a sentence [23, 25]. Self-attention models have often adopted the notion of a multi-head attention, which can simultaneously incorporate different types of word relationships [37].

Most of the aforementioned approaches work in a completely automated manner without involving humans in the loop. However, their performances often leave room for improvement, where humans can intervene in the process.

UI design for interactive document labeling

Traditionally, studies show that experts commonly put a great deal of effort for accurate labeling of data [28, 39]. Because labeling can be time-consuming and often arduous, understanding better design that offloads labelers' cognitive burden has recently been an active area of research in the HCI communities. Many approaches are based on *active learning*, which aims to identify a small set of data for humans to label while achieving a satisfactory prediction accuracy.

For instance, Bernard et al. introduced a visual labeling UI in a 2D embedding view. The UI applied active learning and automatically suggested the most influential data to label [4]. Kuleszat et al. also suggested the technique called structured labeling that is designed for the labelers' consistent labeling [21]. Label and Learn [34] allows users to predict the classifier's expected performance so as to avoid inaccurate labeling. Finally, ELA [9] provides a rich set of interactive labeling capabilities through topic modeling and automatic keyword generation.

Although existing approaches adopt active learning to improve the UI design for properly supporting labelers' tasks,

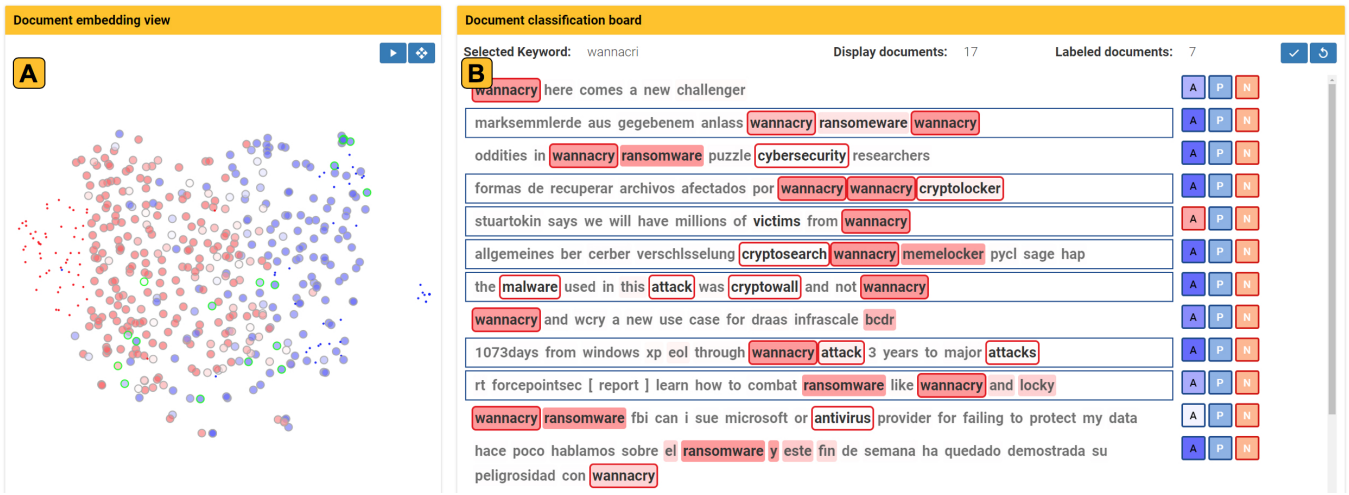

Figure 1: Overview of AILA. (A) The *document embedding view* lays out the documents using t-distributed stochastic neighbor embedding (t-SNE) algorithm. The circle represents a document that has not yet been labeled, and the color of the circle represents the predicted score by the model. The document selected on the document classification board is represented by a green border. The colored dot is the document that the user has labeled. (B) The *document classification board view* presents a document list where the attention weight of every word is highlighted in a red-colored background. A labeler can select important words for labeling, which is displayed with a red-colored border line. For each document, labelers can use "A" (As predicted), "P" (Positive), or "N" (Negative) buttons to label. When labelers select a label, the selected documents are bordered by the selected label color.

to the best of our knowledge, no approach has adopted attention models in designing labeling UIs and/or leveraging human labeling results to train attention-based models as a main classifier. In applying attention models in ML pipelines, we identify the two following design opportunities. First, using the attention weight that attention module calculates for predicting classification, the system may visually emphasize words that users may pay attention to for labeling a document. Such a visual emphasis can guide labelers to focus on relevant words with less cognitive effort and improve labeling performance. Second, enabling labelers to select the relevant words they used for labeling a document can be an effective means to train the attention model itself. Based on the notion of active learning, presenting such interactivity between labelers and the model can improve labelers' overall labeling performance.

3 INTERACTIVE ATTENTION MODULE

An attention model in neural networks has been inspired from computational neuroscience, but it also has strong connections to information visualization [17, 20]. Most animals focus on particular parts of visual input to quickly respond while ignoring a majority region perceived as unimportant [20]. Based on this observation, the studies identified that animals do neural computations by searching the most important information in visual signals [10]. In building deep neural networks (DNNs), the idea of attention models has been applied to capture the relevant parts of the object to improve the prediction performance.

Our core component called IAM in the attention-based DNN model enables human labelers to interact with our model and gradually improve a labeling environment in two ways. First, IAM predicts the attention weight of an individual word in a document (see words in red in Fig. 1 (B)) so that labelers can focus on the words that might be important for labeling. At the same time, IAM leverages user interaction to collect the words that the labelers perceived as important for labeling documents (see words in rectangles in Fig. 1 (B)) to improve the quality of the IAM module for attention weight prediction. Second, IAM presents input to a classifier in our model that predicts the class of a document to improve the classification accuracy. The prediction output is shown in the color of the buttons labeled "A", as shown in Fig. 1 (B), which stands for "As predicted".

In this section, we articulate how the IAM and the classifier work together. Next, we present quantitative experiments that show IAM's reliability.

Model pipeline

Fig. 2 shows the overview of our model. Our model takes a *labeled input*, documents with labeled classification, and an *attention input*, which is a set of words that labelers indicated to be important for labeling documents. Each word in a labeled document is initially represented as an embedding

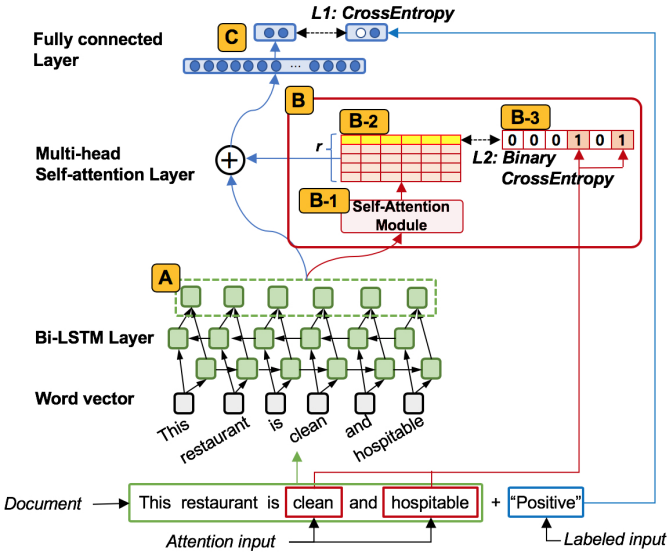

Figure 2: Overview of our model: (A) word vector conversion, (B) IAM, and (C) the classifier.

vector. A Bi-LSTM layer converts the word vector to a hidden-state vector that encodes the context of a document (see Fig. 2 (A)). This vector and the attention input are both used in IAM (see Fig. 2 (B)).

IAM extends Lin et al.'s multi-head self-attentive sentence embedding model [23]. Using the hidden-state vector calculated in the Bi-LSTM layer, the multi-head self-attention module shown in Fig. 2 (B-1) identifies those words that labelers may pay attention to and present the output in the r different forms, or heads, of attention weight vectors. The output is then stored in Fig. 2 (B-2). Next, IAM utilizes a labeler's attention loss by comparing the r heads of attention weight vectors and the attention input stored in Fig. 2 (B-3). Specifically, IAM applies a *binary cross entropy loss* function between the first-head attention weight vector from r heads of attention weight vectors and the labeler's attention input in generating attention weights. Using the IAM output, or the recalculated first-head attention weight vector, the context vectors are obtained as the weighted sum of r multi-head attention weights and the hidden-state vector derived from the Bi-LSTM layer. After r context vectors are concatenated as a single vector, two consecutive fully-connected layers transform it to the final prediction over different classes (see Fig. 2 (C)). In predicting the class of a document, the classifier is trained via the conventional cross entropy loss.

IAM's core novelty lies in how it defines the attention loss. Specifically, IAM considers both the attention loss specified by labelers (L2) in addition to the conventional classification loss (L1) to achieve a user-dependent attention module. In a conventional design, the attention weight is indirectly trained through the classification loss, which has limitations in reflecting labelers' intent. IAM improves the conventional design by training the model jointly using the two losses. Each loss is used to train the classifier and the attention weight one by one (i.e., using L1 and then L2) to emphasize the particular words that reflect each labeler's personalized attention. A labeler's attention loss, or L2, can be computed as

$$L_a(U, A) = -(\sum_{i=1}^{n} u_i \log a_i + (1 - u_i) \log(1 - a_i)),$$

where n indicates the total number of words in a document, U is the labeler's attention input as shown in Fig. 2 (B-3), A is the first-head attention shown in Fig. 2 (B-2) colored in yellow, u_i is the labeler's attention input of word i, and a_i is the first-head attention weight of word i.

Reliability Assessment

To assess IAM's reliability, we investigated the following questions: **Q1**. Can adding the attention loss from labelers lead to an accurate detection of important words in a document, even with few labeled inputs? **Q2**. To achieve a certain degree of classification accuracy, how many of the labeled inputs would be required?

Q1. Assessing attention weight prediction. We used 200,000 reviews that have rating scores between 1 and 5 from Yelp Dataset Challenge 2015 [1] as documents to be labeled. Among 200,000 documents, we randomly selected 400 documents that have scores of 1 or 5 to be the labeled inputs used for training two models: one that adopts IAM and the other that doesn't adopt IAM. The reason that we only considered scores of 1 or 5 to train models was to achieve clear training outcome, as documents rated as 1 may have few positive words while the documents with 5 may have few negative words. Using the two models we trained differently, we predicted the attention weights of additional 100 documents from the Yelp dataset. Documents with every rating score were considered at this stage. Consequently, we obtained two results, generated based on the model that did not adopt IAM, **R1**, and the other that adopted IAM, **R2**. Each result contained the attention weights of every word in the documents. As attention inputs were required for training the IAM based model, three authors manually collected 426 words. We note that the 426 words were used at once for training, which would yield a different outcome than that of a case where labelers indicate the attention input one by one. We ran statistical analyses as follows:

We first measured whether applying IAM can better detect "some" words in a document than not applying IAM can. To measure the detectability of words, we measured the standard deviation (SD) regarding the attention weights of each document in R1 and R2. A higher SD mean a larger

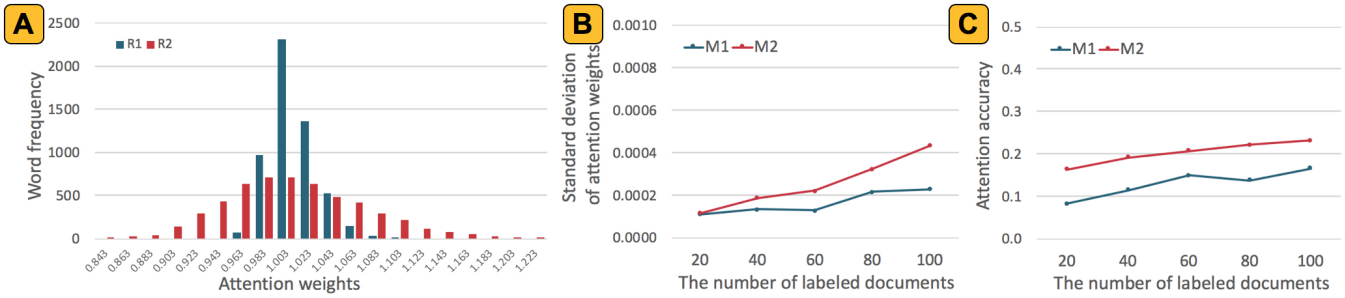

Figure 3: Two distributions that show the attention weights of R1 and R2 (A). Trends that show how the word detectability (B) and the proportion of important words (C) change depending on the number of labeled input.

variance between the attention weights, which indicates that the model can visually emphasize "some" words. In this analysis, we hypothesized that the SD observed in R2 will be higher than those in R1. Our paired samples t-test showed that the SD in R2 was significantly smaller than that in R1 (R1: M=0.0003, SD=0.00021, and R2: M=0.0022, SD=0.00132, t(99)=-16.458, p=0.00). Fig. 3 (A) shows the two distributions of every attention weight between R1 and R2. The graph shows that R1's distribution is more centered than the distribution of R2. We found that the attention weights between words in documents in R2 have a higher standard deviation than those in R1. Such characteristics of R2 would help labelers focus on certain words in a document.

In the second round of analysis, we first aimed to understand whether the detected words can be actually important in making a labeling decision. To do so, we measured precision metrics that show the proportion of how many of the words whose attention weights are higher than $\mu + \sigma$ in a document (i.e., top 15.7% of the words that have the highest attention weights; true positive and false positive) are actually listed in Hu's positive and negative English words (i.e., true positive) [16]. We hypothesized that the proportion would be higher in R2 than in R1. We conducted paired samples t-test using the proportions. The results showed that the proportions in R2 was significantly higher than those in R1 (R1: M=0.1684, SD=0.10585, and R2: M=0.2350, SD=0.16441, conditions; t(99)=-5.936, p=0.00). Next, we measured the recall metric; among all the words in a document that are also listed in Hu's list (i.e., true positive and false negative), how many of them are included in the top 15.7% words in terms of attention weights. The paired samples t-test found that the proportions that indicate recall show no significant differences between R1 and R2 (R1: M=0.2877, SD=0.22239, and R2: M=0.2922, SD=0.23173, conditions;t(99)=-.409, p=0.683). Our findings show that (1) IAM can better detect important words for making a labeling decision than the existing approaches can, but (2) using IAM does not mean that it can better detect *every* important words in a document.

Aside from our measurements, we plotted how the word detectability (i.e., SD) and the proportion of important words (i.e., precision) vary depending on how many labeled inputs are used for training the two models trained without using IAM (M1) and trained using IAM (M2). Fig. 3 (B) shows the trend of the word detectability between M1 and M2. The graph shows no difference when 20 labeled inputs are used. But M2's word detectability becomes better than that of M1 as more labeled inputs are used for training. Fig. 3 (C) shows the trend of the proportion of important words between M1 and M2. The plot shows M2's better performance over M1 regarding this measurement in general.

Q2. Assessing document classification prediction. To understand the second perspective, we randomly selected input labels from 200,000 documents. To see how the prediction accuracy varies depending on the size, we chose different sizes of input labels starting from 50 documents up to 250, increased by an increment of 50. In total, we had six sets of labeled inputs. In building our input sets, we selected half of the inputs from positive documents (i.e., the documents rated as '5') and the other half from negative documents (i.e., the documents rated as '1'). For each of the differently-sized labeled inputs, we trained two models, one without using IAM (M1) and another using IAM (M2). In training M2, we used the same 426 words in Q1 as attention inputs throughout. Using M1 and M2, we predicted the sentiment of 100 documents rated either 1 or 5, which we randomly selected. The 100 documents were all different from the labeled inputs used for training models. The prediction accuracy was computed based on how accurately the models predicted the sentiment (i.e., documents with 1 to be predicted as negative and 5 to be positive). We repeated this process 10 times and yielded the average accuracy (and also the confidence intervals), which is shown in Fig. 4. The results show that there is no difference between M1 and M2 when the size of labeled input is 50, but M2 shows more accurate results than M1 between the size of 100 and 200. The accuracy of M2

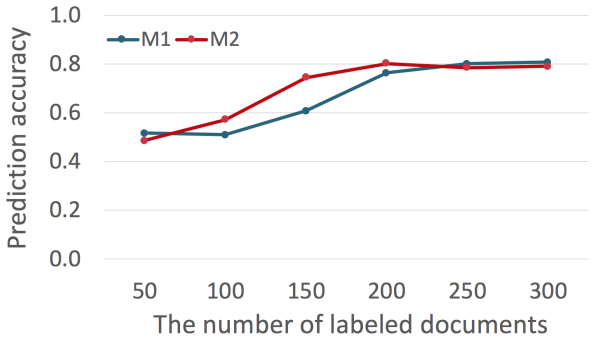

Figure 4: Change of document prediction accuracy according to the size of labeled inputs.

converges to around 0.8 after training the classifier with 200 labeled inputs, whereas the accuracy of M1 converges to the point after using 250.

4 AILA

AILA presents interactive labeling environment for classifying a document. We discuss AILA's major design considerations and how we reflected such considerations in designing AILA's UI. We then introduce AILA's architecture and implementation in detail.

Design Considerations

This subsection discusses major design considerations (DCs) that we believe are critical for supporting an efficient and accurate document labeling environment and articulates how AILA's features are designed based on the consideration.

DC1. Supporting "one-by-one" labeling documents. We adopt the fashion of allowing labelers to classify one data point at a time as such fashion is one of the most reliable ways to produce accurate datasets. The flip side of such design is the potential cognitive burden that labelers may need to cope with. We attempt to offload their burden as follows.

AILA presents visual aids that may help labelers focus on the important parts of a document for labeling so that they don't need to serially scan the entirety of the words in a document. Using IAM, AILA visualizes the attention weights using a sequential color scale of red to present the predicted degree of importance that labelers may focus on in a document, as shown in Fig. 1 (B). Aside from this visual emphasis, AILA allows a labeler to specify the words that (s)he perceived to be important in understanding the theme of the document (see Fig. 1 (B)). In supporting such an indication process, asking a labeler to set the same words every time as they appear can be inefficient. To improve upon this, AILA highlights words in all of the other documents upon a labeler's selecting a word in one document.

Next, AILA presents information about the predicted labels of unlabeled documents. Specifically, AILA provides "As predicted" button in addition to "Positive", and "Negative" buttons for each document to help a labeler in making labeling decisions (see the three buttons that are labeled with "A", "P", "N" in Fig. 1 (B)). The color of the "A" button is displayed as blue if a document is predicted as positive, or red if negative. The color intensity of the button increases as the predicted score gets higher. If the score is not biased to one side, the color of the button remains white. If a labeler chooses to press the "A" button, the system receives the user's input as the same label the model predicted.

DC2. Presenting visual guidance relevant for supporting labeling in real-time. In building an interactive system, selecting information relevant to a user's task and presenting the outcomes in real-time are critical to warrant usability [15]. AILA executes the two learning routine types based on two data types it receives from labelers and the update in its front-end with newly computed outcomes accordingly upon the completion of learning routines.

First, it learns labelers' labeling outcome to improve the prediction accuracy of unlabeled documents (i.e., a labeler hits any button between "A", "P", or "N" in Fig. 1 (B)). For successfully accomplishing the first learning routine, it is critical to randomly sequence labeled data points that capture a good balance between every possible category (in our case, positive and negative), which is critical for reducing the potential of biased learning. To achieve this, AILA provides a training function, which collects multiple labeled data in a buffer. When a batch of newly labeled data points is ready, the function trains the model by randomly shuffling the sequence of the data points. Upon training, the color of every "A" button is updated.

Second, AILA learns from the words that labelers indicated to be important and recompute the attention weight of each word. The words indicated as important is presented with a red border line as Fig. 1 (B) shows. Technically, the attention weights in each document sum up to one as the attention weight of each word is the softmax output of previous hidden layer weight. Therefore, the smaller the number of words in a document, the larger the average attention weight becomes due to the smaller denominator. To normalize, we scale the $1/n$ point to 0.5 point and add a log scale value to emphasize values that are greater than 0.5. Upon a labeler's indication of an important word, the attention scores of every word in the documents are computed and updated in AILA's UI.

DC3. Ordering documents. Labelers may put additional effort into labeling multiple documents when a system orders documents without considering labeling difficulties or thematic consistency. With the lack of such considerations for ordering documents, labelers may encounter blocks or experience

"contextual jump". To reduce the labeler's burden of classifying multiple documents with varying topics, we discuss possible document ordering strategies.

Entropy score: The attention weights between words in a document allow a labeler to recognize which words to pay more attention to for labeling. In general, labeling can become easier for a document with concentrated attention weights on a subset of words. On the other hand, labelers may have to read through more words if the attention weights are evenly distributed across the words. This may increase the perceived effort for labeling. We can measure the unevenness of attention weights within a document using standard deviation or an entropy score (i.e., lower entropy score means higher variances between attention weights) [30].

Prediction score: A document's prediction score shows how certain the model predicts a document's category. When our model predicts the label of a document, all the values of a last fully-connected layer (shown in Fig. 2 (C)) pass the softmax function, which normalizes values by making all the values sum up to 1. This means that all the positive and negative values of last decision layer sum to 1. We can compare these positive and negative scores. By calculating the absolute distance of the positive and negative scores, we can measure the degree to which the model is certain about predicting a document's label. Specifically, if the absolute difference between the positive and negative scores is bigger, that means the model's prediction certainty is higher. We can assume that the label with a higher prediction score can be easier to label whereas a lower score may increase the difficulty level of labeling.

Each document's level of labeling difficulty can be estimated using the the entropy score and the prediction score. Using the two, we can divide the documents into the following types. Type 1: documents that has high prediction score and low entropy. This document type would be relatively easy to label. Type 2: both prediction and entropy are high or low, which would be intermediate level, and Type 3: having low prediction score with high entropy, which would be relatively difficult to label. In ordering documents, placing Type 1 serially may lessen labelers' burden, while the model may learn little based on people's labeling. Type 2 may impose more burden in labeling, but the model may improve it's accuracy than getting input from Type 1. Finally, Type 3 many be the hardest documents, therefore acting as a block. In our implementation, we present Type 1 and Type 2 first, then Type 3 at the last.

We suggest one way to categorize the three types of documents. We define the intermediate "region" as when the range of the attention entropy score and the prediction score are both [0, 1], so the virtual line L can be represented such as $y = x$. The distance between an arbitrary document and L can be expressed as $|d_x - d_y|/\sqrt{2}$, where d_x is the attention entropy score and d_y is the prediction score of each document. We define the document as effective if the distance of L and a document is less than a certain criterion, which we set as 0.3 in AILA.

Additional way ordering documents: The document embedding view presented in (Fig. 1 (A) can help labelers in selecting documents and lists those on the right side of AILA. The view embeds the document on the 2D screen using the t-distributed stochastic neighbor embedding (t-SNE) [27] algorithm, which we will discuss in greater detail in DC4. Finally, AILA provides the keyword search function.

DC4. Adopting Visual Information-seeking Mantra. Visual information-seeking mantra presents useful interaction design guidelines and enables users to comprehend the overall landscape of a large dataset [33]. The mantra has been widely adopted in a variety of everyday information-seeking systems and has guided people to achieve efficient and effective exploratory information-seeking [22]. In building AILA, we adopted the mantra to help labelers to see the overview of documents and identify their overall progress.

First, AILA presents the view named "document embedding view", which lays out the documents on a 2D screen using the t-SNE (t-distributed Stochastic Neighbor Embedding) algorithm (See Fig. 1 (A)) [27]. The algorithm considers a document as a set of words that form a vector to project the document on a Cartesian space. To obtain the word vector representation, we use the GloVe model [31], where we can get pre-trained word vectors from. In the embedding view, a dot represents the documents that the user has labeled, whereas a bigger circle signifies a document that has not been labeled. The intensity of the color in a circle indicates the prediction score for the document, which implies that the higher the intensity, the higher the prediction score is. If a labeler selects some documents using the embedded view, the selected documents are colored as green and displayed in the document classification board (See Fig. 1 (B)).

Second, AILA provides a view that helps a labeler understand their progress as well as how the model progresses through time (See Fig. 5). A donut chart in Fig. 5 (A) shows the progress of a labeler's work. The number in the middle

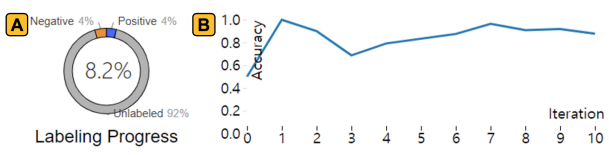

Figure 5: Visualization of labeling progress: AILA provides (A) a donut chart that shows progress in labeling and (B) a line chart that shows model's prediction accuracy

of the chart represents the percentage of documents labeled by a labeler. The percentage of positively labeled documents is presented as as a blue slice while the percentage of negatively labeled documents is shown as a red slice on the donut chart. The proportion for unlabeled documents remains gray.

The line chart in Fig. 5 (B) shows the training accuracy of a model measured at each point where the model is retrained with a batch of newly labeled data points. The accuracy here means the percentage of documents of which the AILA's prediction and the classification that a labeler has indicated match each other out of every document that the labeler has classified so far. The label that a labeler indicated is used as ground truth. In general, the accuracy tends to change largely at the beginning because the model varies depending on a labeler's labeling decision. As the model retrains itself based on newly labeled data points, the accuracy tends to increase, which indicates that the model is stabilized at a high level of accuracy prediction percentage. In presenting the second view that shows the labeling progress along with the model's prediction accuracy, we include these two visualizations above the two views presented in Fig. 1.

Architecture and implementation details

System architecture. Fig. 6 shows AILA's overall architecture. AILA has four parts as follows: Fig. 6 (A) Data preprocessing, Fig. 6 (B) Document analysis module, Fig. 6 (C) Document classifier, and Fig. 6 (D) Interactive labeling interface. The system follows this process.

Data preprocessing model in AILA loads documents and creates indices of the loaded documents and words from the entire dataset. The module performs stemming at each word, which analyzes words by finding the roots of each word, and generates a term-document matrix and document vectors (Fig. 6 (A)). The Text classification model and attention weights are trained in real time by the user-labeled documents. Document Classifier presented in Fig. 6 (C) is trained upon new inputs. Also, upon receiving new input, it recalculates and evaluates the prediction scores and the attention weights for unlabeled documents. The document analysis module selects a document and sorts the document by the prediction score and the attention weight(Fig. 6 (B)). When the system learns the user-labeled documents in real time, users can interact with the system through the Interactive labeling interface (Fig. 6 (D)).

Implementation details. AILA's front-end UI and the back-end model communicate through WebSocket protocol. The front-end was implemented based on JavaScript libraries including jQuery and D3.js [5]. The back-end was built using Python. A Web server was built based on Flask. Additionally, the DNN module was written in PyTorch library. For our sentence classification model of IAM, we used bidirectional

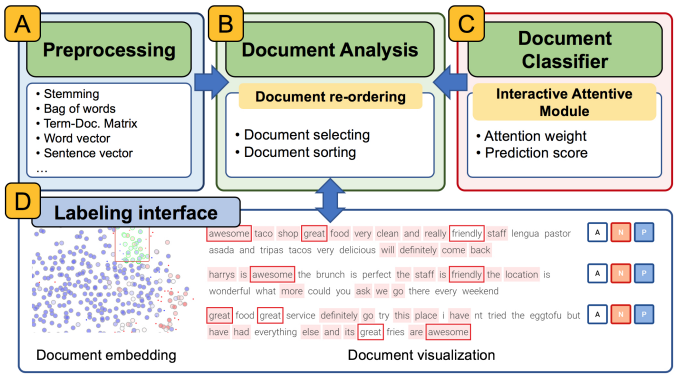

Figure 6: System architecture of AILA: (A) Data preprocessing, (B) Statistical computation engine, (C) Document Classifier, and (D) Interactive labeling UI.

LSTM model with 300 dimensions in each direction, and used max pooling across all LSTM hidden states to get the sentence embedding vector, then used a 2-layer rectified linear units (ReLU) output multilayer perceptron (MLP) with 512 hidden states to output the classification result. In addition, our self-attention MLP has a hidden layer with 350 units, and we chose the matrix embedding to have 5 rows and a coefficient of 1 for the penalization term.

5 STUDY

In our study, we especially focused on understanding (1) whether a labeling environment that adopts IAM can facilitate labelers' efficient labeling performance, and (2) whether AILA can present usable labeling environment in general.

Methodology

IAM within-subject study. To understand the effects of IAM, we conducted a within-subject study. In this study, we hypothesized that the adoption of IAM in designing UIs for document labeling would increase labelers' task efficiency, as paying attention to the visual stimuli chosen by IAM can offload more of the labelers' cognitive load than the condition where they may need to serially scan words in a document can. Meanwhile, we see it is possible that such improved labeling efficiency can entail decreased labeling accuracy.

To conduct the within-subject study, we prepared two UIs: one that did not present IAM (C1) and one that presented IAM (C2). We recruited 15 participants through email lists used for recruiting study participants from Korea University, who reported that they frequently use Yelp to see reviews in order to find places to visit. Participants' ages ranged from 22 to 39 (μ = 25.4, σ = 4.35). 11 reported themselves as male and 4 reported oneself as female. Upon their arrival at the lab, we explained the overall process of the study and features that they can operate in AILA to label documents. Participants

then tried out AILA for five minutes to familiarize themselves with the system before the study. In that stage, participants were able to randomly access 200,000 reviews we collected from Yelp Dataset Challenge 2015 [1]. Before the main study, we asked if they had become familiar with AILA, and all participants responded positively. Once they said they were ready to label, we asked them to label datasets twice using C1 and C2, respectively. For each condition, participants used the three buttons that are presented in AILA (A - As predicted, P - Positive, and N - Negative) to label documents for 10 minutes. A time period of 10 minutes was chosen as labelers may be able to retain one's attention without further effort [40]. Therefore, we concluded that the amount of time is enough for capturing the effects that we wanted to observe.

The two conditions were presented in a counterbalanced order. To remove the occurrence of the learning effect in within-subject design, we prepared the two datasets, D1 and D2, which we randomly sampled without replacement from the whole dataset. D1's mean word count was 51.750 (SD: 24.808) and D2's mean was 53.026 (SD: 23.482). Every participant labeled D1 for the first condition then D2 for the second condition. Because of this experimental design, the differences in the difficulty level and the length between D1 and D2 as well as the ordering effect between C1 and C2 are all be counterbalanced (i.e., Half of participants in C1 labeled using D1 and another half labeled used D2). We note that we did not exclude the documents that participants were seeing when learning AILA in building D1 and D2, as we expected that participants may focus on AILA's features rather than on the documents themselves, and the chance of them seeing the same documents in D1 and D2 would be scarce.

To measure the labelers' task efficiency, we counted how many documents participants were able to label within the given time threshold. To determine whether there exist the potential trade-offs between labeling efficiency and labeling accuracy, we also measured participants' labeling accuracy in C1 and C2. We note that we only collected the reviews that have a score of 1 or 5 in collecting D1 and D2 for measuring the labelers' labeling accuracy. We assumed a labeler to label a document accurately if one labeled a score 1 documents as negative or score 5 documents as positive.

AILA usability study. After the participants finished using the two conditions, we presented the condition with IAM (C2) one more time (i.e., AILA that presents its full features), and let them freely use it so that they could recall every feature. We then asked six questions listed in NASA TLX questionnaires [13] that are related to the perceived workload for using a system to measure their perceived workload of using AILA. After they submitted the questionnaires, we conducted a semi-structured, closing interview where our main focus was in capturing the participants' general impression about AILA, usability, and AILA's strengths and drawbacks they perceived while using it. We also elicited missing features from the participants that may improve labeling environment in general.

Results

Effect of IAM. In terms of the number of labels that participants finished in the 10 minutes in C1 and C2, the paired-sample t-test found that the participants labeled significantly more documents when they used C2 than when using C1 (C1: M=71.333, SD=33.070 and C2:M=90.933, SD=42.010, conditions; $t(14)=-3.602$, $p<0.005$). Fig. 7 presents a bee swarm plot overlaid with a box plot. In terms of the normality test regarding the number of labels participants performed, the outcomes approximately followed normal distribution; outcomes in C1 showed a skewness of 0.933 (SE=0.580) and a kurtosis of -0.195 (SE=1.121), and C2 showed a skewness of 0.476 (SE=0.580) and a kurtosis of -0.450 (SE=1.121). To understand the labeling efficiency and accuracy trade-offs, we conducted another paired-sample t-test. We found that there was no significant difference between C1 and C2 (C1: M=0.952, SD=0.036 and C2:M=0.964, SD=0.038,conditions; $t(14)=-1.087$, $p=0.295$). These results show that presenting IAM can improve labeling efficiency without significant labeling efficiency and accuracy trade-offs.

Usability of AILA. Fig. 8 shows NASA TLX survey results. Participants generally agreed that the required mental and physical demands for labeling documents using AILA were low (Required mental effort was not demanding: $\mu = 4.13$, $\sigma=1.11$, Strongly disagree=1.0, Strongly agree=5.0, hereinafter, required physical effort was not demanding: $\mu = 4.53$, $\sigma=1.49$). Temporal demand (i.e., didn't feel time pressure) and effort (i.e., how hard they labeled) were slightly above average. We

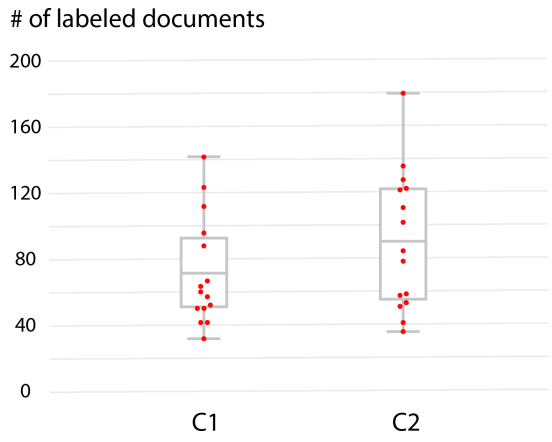

Figure 7: Number of labels participants accomplished in our within-subject study

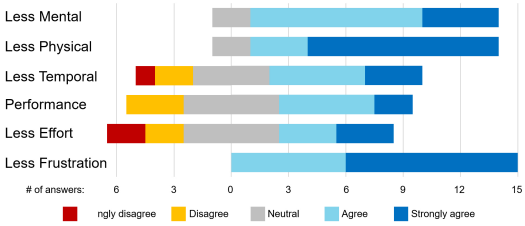

Figure 8: NASA TLX survey results of AILA

assume these results are related to the instruction we gave to participants in the experimental study where we asked them to label the documents with *sincere effort* for the fixed amount of 10 minutes. In summary, participants generally agreed that the outcome they made using AILA was successful (Tasks were performed successfully: μ = 3.47, σ=1.65), and strongly agreed that they didn't experience frustration (Had little frustration: μ = 4.6, σ=1.31)).

Through the interview, we found a series of perceived advantages and disadvantages of AILA. Regarding the visually externalized attention weights, participants remarked that such a design choice helped them label the documents efficiently because the keywords encoded with colors were useful in assessing the sentiment without comprehending the whole context of a document. For instance, they remarked: *"Scanning colored words helped me classify sentences pretty easily."* (P1). *"I felt like I could label many words correctly even if I didn't read the full text."* (P5). On the other hand, although participants agreed that learning AILA's core features didn't take too much time in general, some participants mentioned that they encountered a learning curve.

6 DISCUSSION

We discuss the core findings from the reliability assessment of our model and studies along with limitations and future research opportunities.

AILA's model improved labelers' labeling environment by presenting (1) words that may be important for labeling a document and (2) a prediction of a label of a document. Our assessment found that our model detected a greater number of important words than existing approaches. However, the findings show that using our model does not mean that it can detect every single important word in a document, which presents interesting research opportunities. Also, we found that our model can exhibit over 80% of document classification accuracy with roughly 200 labeled input. As our case was predicting the label of documents with a few sentences, we expect more labeled inputs and attention inputs may be required in applying our attention-based approach to (1) more complicated tasks that involve more than two classification labels, or (2) labeling of different medium than a document, such as images or videos. Such different labeling environment will entail substantial design and computational challenges, which would present unique research opportunities.

Our study results found that the visual stimuli presented using IAM enabled labeling of more documents within a given time threshold. However, the stimuli didn't improve or deteriorate the labeling accuracy. The labeling accuracy was over 90% in both conditions of our study, as the labeling task was straightforward. More complex labeling tasks, such as tasks that require complex thinking with professional domain knowledge, may yield a different outcome. In designing a system for supporting such cases, the building of a model carefully tuned towards target task types would be critical. Aligning with the task efficiency gain, the usability study results indicate that the participants' perceived mental and physical efforts for labeling tended to be low.

One of the most exciting observations we found is the possibility of applying attention mechanism in improving ML data pipelines in general. As ML becomes more widely applied, understanding better design for improving the ML pipeline has been investigated by the HCI community and beyond. Design fashions, such as active learning, human-in-the-loop, or interactive ML, discuss the usefulness and importance of human involvement in improving ML models. We think that the attention mechanism can present a good design rationale for researchers who aim to improve ML data pipeline, for example, in building UIs for labeling, assessing, debugging, or comparing ML models.

7 CONCLUSION

In this work, we presented AILA, a system that adopts a model that uses Interactive Attention Module (IAM). In implementing IAM, our back-end ML engine utilized a text classification model based on a DNN while our front-end presented an interactive UI that elicits labelers' inputs that caught their attention in labeling a document, which were used for gradually improving our model. We expect that the interactive attention mechanism we proposed can be applied in supporting a broader range of labeling tasks that involve different types of media such as images or videos, labelers who have knowledge in specific domains such as medical doctors or lawyers, and different sizes of groups with their unique group dynamics.

ACKNOWLEDGMENTS

This work was partially supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIP) (No. NRF-2016R1C1B2015924) and R&D program for Advanced Integrated-intelligence for IDentification (AIID) through the NRF funded by Ministry of Science and ICT (2018M3E3A1057288).

REFERENCES

[1] 2015. Yelp Dataset Challenge. https://www.yelp.com/dataset/challenge. Accessed: 2018-10-05.

[2] Hunt Allcott and Matthew Gentzkow. 2017. Social media and fake news in the 2016 election. *Journal of Economic Perspectives* (2017). https://doi.org/10.1257/jep.31.2.211

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

[4] Jürgen Bernard, Matthias Zeppelzauer, Michael Sedlmair, and Wolfgang Aigner. 2018. VIAL: a unified process for visual interactive labeling. *The Visual Computer* (2018). https://doi.org/10.1007/s00371-018-1500-3

[5] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. 2011. D^3 data-driven documents. *IEEE Transactions on Visualization & Computer Graphics* (2011). https://doi.org/10.1109/TVCG.2011.185

[6] Carlos Castillo, Debora Donato, Aristides Gionis, Vanessa Murdock, and Fabrizio Silvestri. 2007. Know your neighbors: Web spam detection using the web topology. In *Proceedings of the SIGIR conference on Research and development in information retrieval*. https://doi.org/10.1145/1277741.1277814

[7] Joseph Chee Chang, Saleema Amershi, and Ece Kamar. 2017. Revolt: Collaborative crowdsourcing for labeling machine learning datasets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. https://doi.org/10.1145/3025453.3026044

[8] Niall J Conroy, Victoria L Rubin, and Yimin Chen. 2015. Automatic deception detection: Methods for finding fake news. In *Proceedings of the ASIS&T Annual Meeting: Information Science with Impact: Research in and for the Community*. https://doi.org/10.1002/pra2.2015.145052010082

[9] Enrico Bertini Cristian Felix, Aritra Dasgupta. 2018. The Exploratory Labeling Assistant: Mixed-Initiative Label Curation with Large Document Collections. In *Proceedings of the ACM Symposium on User Interface Software and Technology*. https://doi.org/10.1145/3242587.3242596

[10] Robert Desimone and John Duncan. 1995. Neural mechanisms of selective visual attention. *Annual review of neuroscience* (1995). https://doi.org/10.1146/annurev.ne.18.030195.001205

[11] Long Tran-Thanh Edoardo Manino and Nicholas R. Jennings. 2018. On the Efficiency of Data Collection for Crowdsourced Classification. In *Proceedings of the International Joint Conference on Artificial Intelligence*. https://doi.org/10.24963/ijcai.2018/217

[12] Jerry Alan Fails and Dan R Olsen Jr. 2003. Interactive machine learning. In *Proceedings of the international conference on Intelligent user interfaces*. https://doi.org/10.1145/604045.604056

[13] Sandra G Hart and Lowell E Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In *Advances in psychology*. https://doi.org/10.1016/S0166-4115(08)62386-9

[14] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* (1997). https://doi.org/10.1162/neco.1997.9.8.1735

[15] Sungsoo Ray Hong, Min-Joon Yoo, Bonnie Chinh, Amy Han, Sarah Battersby, and Juho Kim. 2018. To Distort or Not to Distort: Distance Cartograms in the Wild. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. https://doi.org/10.1145/3173574.3174202

[16] Minqing Hu and Bing Liu. 2004. Mining and Summarizing Customer Reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. https://doi.org/10.1145/1014052.1014073

[17] Laurent Itti, Christof Koch, and Ernst Niebur. 1998. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on pattern analysis and machine intelligence* (1998). https://doi.org/10.1109/34.730558

[18] Nitin Jindal and Bing Liu. 2007. Review spam detection. In *Proceedings of the international conference on World Wide Web*. https://doi.org/10.1145/1242572.1242759

[19] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188* (2014).

[20] Stephen Kosslyn. 1989. Understanding Charts and Graphs. *Applied Cognitive Psychology* (1989). https://doi.org/10.1002/acp.2350030302

[21] Todd Kulesza, Saleema Amershi, Rich Caruana, Danyel Fisher, and Denis Charles. 2014. Structured Labeling for Facilitating Concept Evolution in Machine Learning. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. https://doi.org/10.1145/2556288.2557238

[22] Jin Ha Lee, Sungsoo Ray Hong, Hyerim Cho, and Yea-Seul Kim. 2015. VIZMO game browser: accessing video games by visual style and mood. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. https://doi.org/10.1145/2702123.2702264

[23] Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130* (2017).

[24] Wang Ling, Yulia Tsvetkov, Silvio Amir, Ramon Fermandez, Chris Dyer, Alan W Black, Isabel Trancoso, and Chu-Cheng Lin. 2015. Not all contexts are created equal: Better word representations with variable attention. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. https://doi.org/10.18653/v1/D15-1161

[25] Yang Liu, Chengjie Sun, Lei Lin, and Xiaolong Wang. 2016. Learning natural language inference using bidirectional LSTM model and inner-attention. *arXiv preprint arXiv:1605.09090* (2016).

[26] Rahul Sukthankar Liyue Zhao, Gita Sukthankar. 2011. Incremental Relabeling for Active Learning with Noisy Crowdsourced Annotations. In *IEEE International Conference on Privacy, Security, Risk and Trust and IEEE International Conference on Social Computing*. https://doi.org/10.1109/PASSAT/SocialCom.2011.193

[27] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* (2008).

[28] Kathleen M MacQueen, Eleanor McLellan, Kelly Kay, and Bobby Milstein. 1998. Codebook development for team-based qualitative analysis. *CAM Journal* (1998). https://doi.org/10.1177/1525822X980100020301

[29] Bo Pang, Lillian Lee, et al. 2008. Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval* (2008). https://doi.org/10.1561/1500000011

[30] Paul Pathria, R. K.; Beale. 2011. *Statistical Mechanics (Third Edition)*. Academic Press. 51–52 pages.

[31] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. https://doi.org/10.3115/v1/D14-1162

[32] Dinghan Shen, Yizhe Zhang, Ricardo Henao, Qinliang Su, and Lawrence Carin. 2017. Deconvolutional latent-variable model for text sequence matching. *arXiv preprint arXiv:1709.07109* (2017).

[33] Ben Shneiderman. 1996. The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings IEEE Symposium on Visual Languages*. https://doi.org/10.1109/VL.1996.545307

[34] Yunjia Sun, Edward Lank, and Michael Terry. 2017. Label-and-Learn: Visualizing the Likelihood of Machine Learning Classifier's Success During Data Labeling. In *Proceedings of the International Conference on Intelligent User Interfaces*.

[35] Justin Talbot, Bongshin Lee, Ashish Kapoor, and Desney S Tan. 2009. EnsembleMatrix: interactive visualization to support machine learning with multiple classifiers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. https://doi.org/10.1145/1518701.1518895

[36] Soujanya Poria Tom Young, Devamanyu Hazarika and Erik Cambria. 2017. Recent Trends in Deep Learning Based Natural Language Processing. (2017). http://arxiv.org/abs/1708.02709

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*. http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf

[38] Wenlin Wang, Zhe Gan, Wenqi Wang, Dinghan Shen, Jiaji Huang, Wei Ping, Sanjeev Satheesh, and Lawrence Carin. 2017. Topic compositional neural language model. *arXiv preprint arXiv:1712.09783* (2017).

[39] Cynthia Weston, Terry Gandell, Jacinthe Beauchamp, Lynn McAlpine, Carol Wiseman, and Cathy Beauchamp. 2001. Analyzing interview data: The development and evolution of a coding system. *Qualitative sociology* (2001). https://doi.org/10.1023/A:1010690908200

[40] Karen Wilson and James H. Korn. 2007. Attention during Lectures: Beyond Ten Minutes. *Teaching of Psychology* (2007). https://doi.org/10.1080/00986280701291291

[41] Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*. https://doi.org/10.3115/1220575.1220619

[42] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*. http://proceedings.mlr.press/v37/xuc15.html

[43] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. https://doi.org/10.18653/v1/N16-1174

[44] Yizhe Zhang, Dinghan Shen, Guoyin Wang, Zhe Gan, Ricardo Henao, and Lawrence Carin. 2017. Deconvolutional paragraph representation learning. In *Advances in Neural Information Processing Systems*. http://papers.nips.cc/paper/7005-deconvolutional-paragraph-representation-learning