DeepLens: Interactive Out-of-distribution Data Detection in NLP Models

Da Song*
University of Alberta
Edmonton, AB, Canada
dsong4@ualberta.ca

Zhijie Wang*
University of Alberta
Edmonton, AB, Canada
zhijie.wang@ualberta.ca

Yuheng Huang
University of Alberta
Edmonton, AB, Canada
yuheng18@ualberta.ca

Lei Ma
University of Alberta, Canada
The University of Tokyo, Japan
ma.lei@acm.org

Tianyi Zhang
Purdue University
West Lafayette, IN, USA
tianyi@purdue.edu

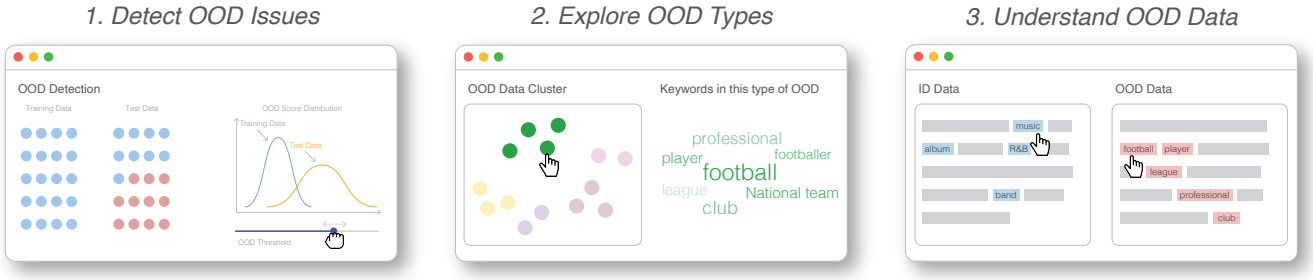

Figure 1: DeepLens is an interactive system for supporting out-of-distribution (OOD) data detection in NLP models. The developer can detect OOD issues by dynamically adjusting the threshold and observing the changes in the icon array and OOD score distribution. DeepLens also helps the developer explore OOD types by clustering similar texts and visualizing keywords. To understand OOD data, the developer can check the highlighted keywords and compare them with in-distribution (ID) data.

ABSTRACT

Machine Learning (ML) has been widely used in Natural Language Processing (NLP) applications. A fundamental assumption in ML is that training data and real-world data should follow a similar distribution. However, a deployed ML model may suffer from out-of-distribution (OOD) issues due to distribution shifts in the real-world data. Though many algorithms have been proposed to detect OOD data from text corpora, there is still a lack of interactive tool support for ML developers. In this work, we propose DeepLens, an interactive system that helps users detect and explore OOD issues in massive text corpora. Users can efficiently explore different OOD types in DeepLens with the help of a text clustering method. Users can also dig into a specific text by inspecting salient words highlighted through neuron activation analysis. In a within-subjects user study with 24 participants, participants using DeepLens were able to find nearly twice more types of OOD issues accurately with 22% more confidence compared with a variant of DeepLens that has no interaction or visualization support.

CCS CONCEPTS

• **Human-centered computing** → **Interactive systems and tools**; • **Computing methodologies** → **Natural language processing**.

KEYWORDS

Interactive Visualization, Out-of-distribution Detection, Machine Learning, NLP

ACM Reference Format:
Da Song, Zhijie Wang, Yuheng Huang, Lei Ma, and Tianyi Zhang. 2023. DeepLens: Interactive Out-of-distribution Data Detection in NLP Models. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23), April 23–28, 2023, Hamburg, Germany*. ACM, New York, NY, USA, 17 pages. https://doi.org/10.1145/3544548.3580741

1 INTRODUCTION

Machine Learning (ML) techniques and ML models have shown superior performance in many applications, e.g., autonomous driving [49], virtual assistant [8], and medical diagnosis [32]. Modern ML techniques usually assume the training data and test data follow a similar distribution. However, such an assumption can hardly be satisfied in the real world. Instead, data distribution shift and out-of-distribution (OOD) samples often result in performance degradation

*Both authors contributed equally to this work.

CHI '23, April 23–28, 2023, Hamburg, Germany

ACM ISBN 978-1-4503-9421-5/23/04…$15.00
https://doi.org/10.1145/3544548.3580741

of deployed ML models [17, 21, 37]. Such performance drop further brings concerns about the reliability and trustworthiness of ML models. In particular, failing to predict OOD samples may lead to serious outcomes in high-stake and safety-critical applications such as health care [32]. In 2013, Google Flu Trends (GFT) model failed to predict the flu season, missing the peak of that flu season by 140 percent [30]. One significant factor of the failure is that GFT did not take into account how users' search behavior had changed since 2012 [30]. In this case, the data distribution shifted in 2013 compared with 2012 and eventually led to poor model performance.

To alleviate the effects of OOD data, many techniques have been proposed for OOD detection [4, 21, 33–35, 44, 58]. For a given data instance, these techniques typically first calculate a specific score (OOD score), and then compare it with a pre-defined threshold to determine if the data instance is OOD. However, in practice, only identifying OOD data is not sufficient for ML developers. For instance, in the previous example of GFT, after knowing there is an OOD issue, developers still need to dig into the OOD data and figure out why they are considered OOD and what their characteristics are. This is a time-consuming process. Though a recent technique [9] has been proposed to address this challenge, it is only designed for image data, not text data. Compared with images, which are more glanceable for humans [15], more cognitive efforts are required to read and understand text data. Without appropriate tool support, it can be challenging and time-consuming for developers to investigate a massive amount of OOD text data at scale.

In this paper, we explore interactive tool support for helping users quickly detect and contextualize OOD samples from large text corpora. We present DeepLens, a novel interactive system that enables users to *detect*, *explore*, and *understand* OOD issues. DeepLens is built upon maximum softmax probability (MSP), a popular calibration-based OOD detection method for text data [4, 21]. To help users explore different types of OOD data in text corpora, DeepLens first clusters similar OOD data by topics and then renders the frequent words in each cluster in a word cloud to help users examine and understand the topic of each cluster. As users delve into individual OOD instances, DeepLens highlights salient words in each instance via neuron activation analysis method [1]. In this way, DeepLens helps users quickly understand a long text without reading it in detail. DeepLens also renders in-distribution and out-of-distribution data side by side to help users compare and contrast them.

To evaluate the usability and efficiency of DeepLens, we conducted a within-subjects user study with 24 programmers with various levels of expertise in ML and NLP. We created a comparison baseline by disabling the cluster view and the highlighting view in DeepLens. The results show that participants using DeepLens were able to find more types of OOD data on four different NLP tasks. The mean difference in the number of OOD types found by each participant using DeepLens and the baseline tool is 3.54 vs. 1.25 (Welch's t-test: $p < 0.0001$). Participants using DeepLens also felt more confident about OOD issues they found in the ML models. The median values are 6 vs. 5 on a 7-point Likert scale (Welch's t-test: $p = 0.002$). These results demonstrate that DeepLens can significantly improve ML developers' productivity when dealing with out-of-distribution issues in NLP models.

In summary, the main contribution of this paper is DeepLens, an interactive system that helps users detect, explore, and understand OOD data in large text corpora. We have open-sourced our system on GitHub [1]. A within-subjects user study demonstrates the effectiveness of DeepLens in detecting and analyzing different types of OOD issues on a variety of NLP tasks.

2 BACKGROUND AND RELATED WORK

2.1 Out-of-Distribution Issues in ML Systems

A fundamental assumption in machine learning theory is that training and test data follow a similar distribution [42]. However, after model deployment, it is not uncommon to encounter real-world data that is *out-of-distribution* compared with the training data. Previous studies demonstrate that when feeding OOD samples, ML models can provide erroneous predictions with high confidence [14, 45]. Such errors can have serious consequences when the predictions inform real-world decisions such as medical diagnosis, e.g. falsely classifying a healthy sample as pathogenic or vice versa [3, 50, 52].

Over the years, there has been an ongoing effort in trying to understand OOD issues in ML systems. Moreno-Torres et al. [43] present a unified framework to analyze the distribution shift. Given a classification task $\mathcal{X} \rightarrow \mathcal{Y}$, the joint probability of $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ can be represented as $p(y, x) = p(y|x)p(x)$. Moreno-Torres et al. [43] then categorize OOD into two types: (1) *covariate shift*, where the input distribution $p(x)$ changes, and (2) *concept shift*, where the relationship between the input and class variables $p(y|x)$ changes. Arora et al. [4] further extend this taxonomy to NLP tasks. They assume text data can be represented as background features (e.g. genre) that are invariant across different labels, and semantic features (e.g. sentiment words) that are discriminative for the prediction task. Therefore, they define the change of background features as *background shift* and the distribution change of semantic features as *semantic shift*. In this work, we follow the OOD taxonomy and terminologies by Arora et al. [4], since our work focuses on OOD issues in text data.

2.2 OOD Detection

There is a large body of literature on OOD detection in the ML community. Most of the prior work calculates an OOD score for each input, and uses a threshold to separate ID data from OOD data. Hendrycks et al. [21] first propose a simple method to detect OOD samples, representing one of the earliest attempts in this direction. They utilize the probability of a prediction (i.e., model confidence) as the indicator for OOD issues, in which a lower probability yields a higher OOD score. However, since DL models often "confidently" make errors [45], only leveraging model confidence hinders further improvement of OOD detection. To address this issue, some recent work proposes to train a calibrated model, so it can give predictions with low confidence on OOD data. The calibrated model can be obtained via data augmentation [22, 57, 62], adversarial training [7, 10, 20], and uncertainty modelling [5, 40]. Another line of work to address the model confidence barrier is to leverage other indicators for OOD detection [4, 33–35, 44, 58]. One of the representative works is ODIN [33], which uses temperature scaling and

[1] https://github.com/momentum-lab-workspace/DeepLens

input perturbation for OOD score computation. Furthermore, OOD detection can also be achieved by estimating the ID distribution and measuring how far the input instance is from the ID distribution [26, 29, 31]. In this work, we develop DeepLens on top of an OOD algorithm by Arora et al. [4], which utilizes maximum softmax probability (MSP) for OOD detection in text corpora.

So far, most efforts have been put into improving the accuracy of OOD detection algorithms. However, only providing an OOD score and a list of OOD samples is insufficient for humans to understand and reason about OOD data. DeepLens fills the gap by providing an interactive system that helps developers explore OOD data detected from large text corpora and understand their characteristics.

2.3 Interactive Support for OOD Detection

In the past two years, there has been a growing interest in providing interactive tool support for detecting distribution change [9, 46, 47, 59–61]. OoDAnalyzer [9], an interactive system for analyzing OOD issues in image data. It provides a grid-based visualization that shows individual OOD images in a grid view. Furthermore, it allows users to zoom into individual OOD instances and highlights the parts of an image that contributes significantly to the prediction result. The main difference between OoDAnalyzer and DeepLens is that OoDAnalyzer focuses on image data while DeepLens focuses on text data. Compared with text data, images are more glanceable. Thus, the interface design in OoDAnalyzer is not applicable to OOD analysis in text corpora. To fill the gap, DeepLens leverages a text clustering method and also highlights salient words in individual text documents to help users explore and understand OOD instances.

Data drift detection is closely related to OOD detection. Yeshchenko et al. propose Visual Drift Detection (VDD) [61], a visualization and interaction system for detecting and analyzing business process drift. By utilizing a set of interactive charts, VDD presents the business process drift (event sequence data) in a time-dependent way. Wang et al. present ConceptExplorer [59], a visual analytics system for analyzing concept drifts from multi-source time-series data. Yang et al. propose DriftVis [60], an visual analytics system for analyzing concept drift in streaming data. It utilizes an incremental Gaussian mixture model to detect samples with concept drift and presents prediction-level visualization that reveals the performance change of the target model. DriftVis is specifically designed for concept drift (i.e., semantic shift in NLP), while DeepLens does not have a specialized design for a particular type of distribution shift and thus can be applied to both shift types.

3 USER NEEDS AND DESIGN RATIONALE

In this section, we first analyze ML practitioners' needs for interactive OOD detection based on the literature review. Then, we discuss how our proposed system supports these needs through a system overview.

3.1 User Needs in Detecting and Diagnosing OOD Issues in ML

To understand the needs of ML practitioners, we conduct a literature review of previous work that has done a formative study of OOD detection [9, 47, 60, 61], has done a user study [46, 59], or has discussed the challenges in handling OOD issues [38, 51]. Based on this review, we summarize five major user needs for OOD detection.

N1: Automatically detect OOD data. Manually inspecting individual instances to identify data distribution shifts is time-consuming and cumbersome [59, 61]. By working closely with their industry partners, Yeshchenko et al. [61] found that industrial practitioners demanded the distribution shift be identified promptly and precisely. Therefore, DeepLens should automatically detect OOD data based on user-defined criteria. The expert review in Wang et al. [59] also confirmed the necessity of automated OOD detection in large datasets.

N2: Understanding why a sample is detected as OOD. Recent studies [38, 47, 59, 60] show that only detecting OOD samples is insufficient. In practice, ML practitioners are often also eager to know why those samples are out of distribution. For instance, Yang et al. [60] interviewed four ML practitioners and found that, instead of simply obtaining the detected OOD samples, ML practitioners desired to know why and where the distribution shift occurred. In another interview study with four data scientists, Palmeiro et al. [47] found that data scientists wanted to know which parts of the dataset include data shift as well as the patterns of data shift.

N3: Identifying different types of OOD data. Different OOD instances may have different characteristics. Therefore, ML practitioners want to categorize OOD data to better understand their commonalities and variations, so that they can come up with a more comprehensive strategy to consider the impacts brought by data shift [9, 51, 59, 61]. For example, Wang et al. [59] highlighted that users should be able to discriminate different types of data and verify the distribution shift of each type. In an interview study with both ML developers and ML users, Chen et al. [9] reported that the ML developers and ML users both desired to visually explore different types of OOD samples and their relationships to reduce the samples that need to be inspected.

N4: Comparing OOD with ID data. Chen et al. [9] found that comparing OOD samples with ID samples under the same predicted label was helpful for users to confirm potential OOD issues. Olson et al. [46] conducted a user study with sixty ML users and found that users often compared OOD samples with ID samples to understand the characteristics of OOD samples.

N5: Investigating OOD issues from both global and local perspectives. When inspecting OOD issues, users tend to first explore different categories of potential OOD data and then delve into a category of interest to compare an OOD sample with similar ID samples [9]. Yeshchenko et al. [61] highlighted the importance of supporting "drill-down" and "roll-up" analysis on OOD data to allow users to flexibly investigate OOD issues from different granularity.

3.2 Design Rationale

To support **N1**, DeepLens leverages a calibration-based method [21] to automatically detect OOD data in a large text corpus. Users can observe the percentage of OOD instances in the test data in the *Distribution View* (Fig. 2 Ⓐ) and adjust to what extent an instance should be considered as OOD via the threshold slider. To help users better understand why some instances are detected as OOD data (**N2**), DeepLens allows users to compare an OOD instance and

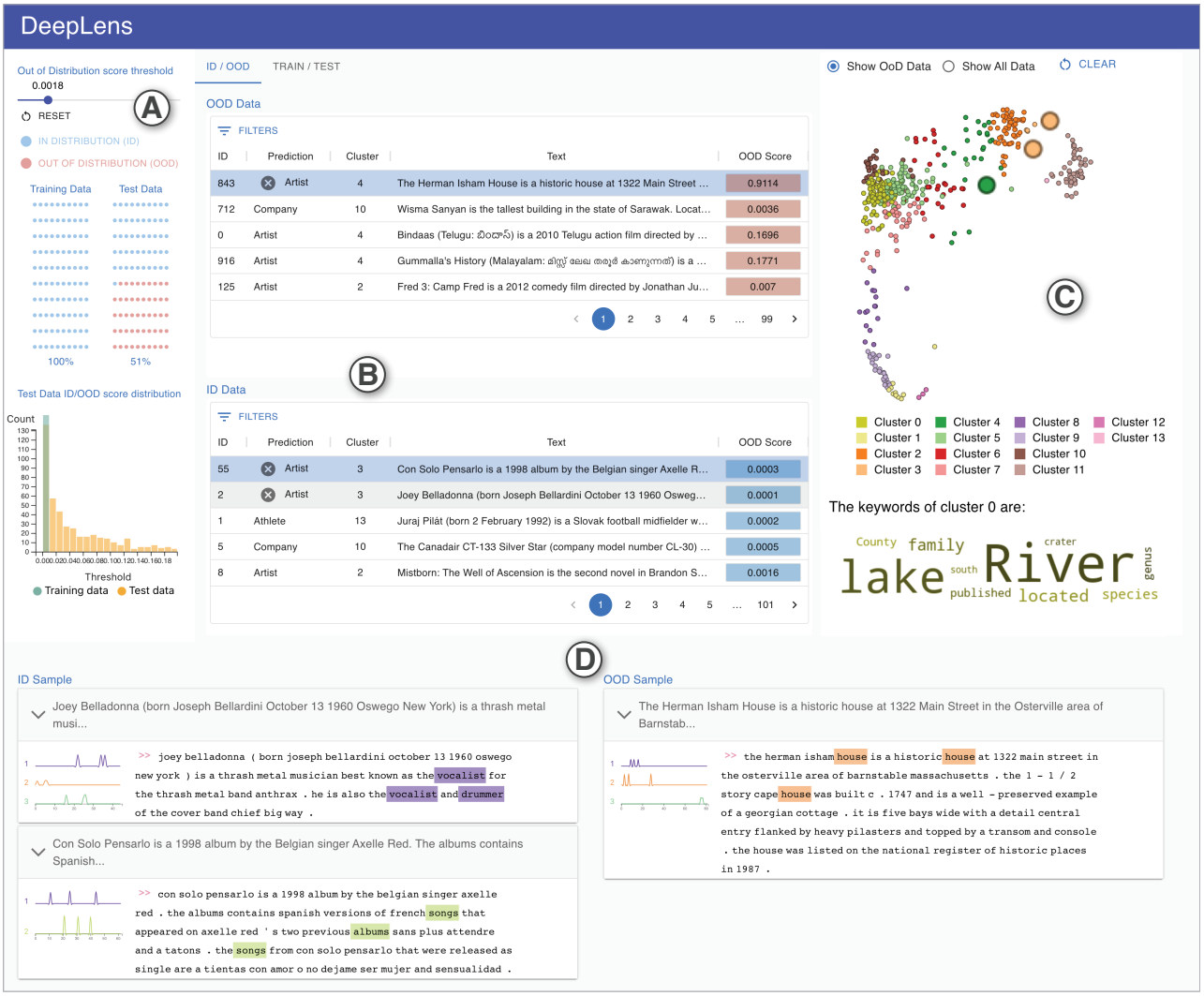

Figure 2: DeepLens, an interactive system for detecting and identifying OOD samples in the text data. (A) The *Distribution View* allows users to adjust thresholds and inspect OOD issues in test data dynamically. (B) The *Instance View* displays the in-distribution (ID) and out-of-distribution (OOD) data in two separate interactive data grids. (C) The *Cluster View* displays the clustering results and keywords for each cluster for exploring potential OOD types. (D) The *Highlighting View* shows the highlighted salient words on selected data instances to ease users' reading efforts.

an ID instance side by side and examine the commonalities and variations between them (Fig. 2 Ⓑ). Furthermore, as some instances are lengthy, DeepLens leverages neuron activity analysis [23] to identify and highlight salient words in an instance, so users can quickly grasp the underlying topic(s) in the instance (Fig. 2 Ⓓ). By directly comparing the differences between highlighted words in an OOD data and an ID data, users can easily identify potential topic differences between those two instances without the necessity of reading through the entire text document.

To assist users in identifying different types of OOD data (**N3**), DeepLens clusters the detected OOD instances and renders them in a scatter plot (Fig. 2 Ⓒ). The common words in a cluster are visualized as a word cloud to help users understand its underlying topics (Fig. 2 Ⓒ). This cluster view, together with the word cloud, helps users obtain a global understanding of when and where data shifts occur in the dataset (**N5**). To support the "drill-down" analysis mentioned in (**N5**), DeepLens allows users to delve into a specific cluster by clicking on a node in that cluster or a cluster legend. The instance view will be filtered accordingly. To support the "roll-up" analysis mentioned in (**N5**), DeepLens highlights the user-selected instances in the cluster view, so users can easily see where the selected instances are in the global view. Finally, to support **N4**, DeepLens allows users to filter the instances by prediction labels in the *Instance View* and then select OOD and ID instances with the same prediction label to compare side-by-side. The salient word

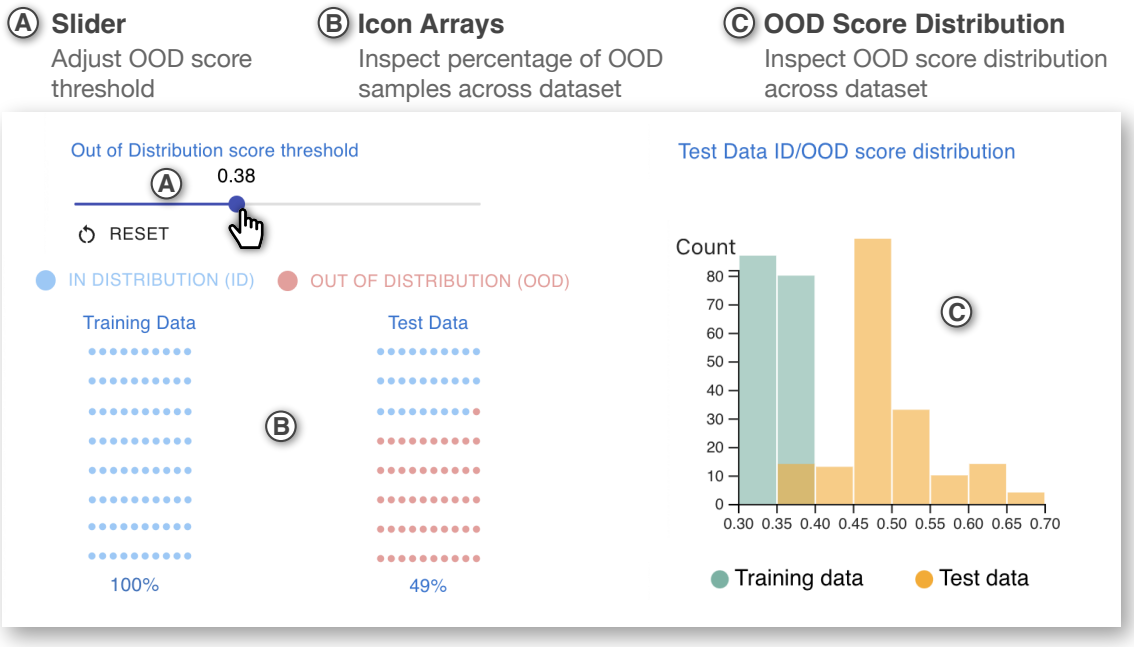

Figure 3: Users can adjust the threshold of the OOD score and inspect OOD issues in the *Distribution View*.

highlighting feature also helps users quickly see the commonalities and variations between the OOD and ID instances.

4 DESIGN AND IMPLEMENTATION

In this section, we follow the 3-step usage of DeepLens (Fig. 1) to introduce its design and implementation: (1) detect OOD issues, (2) explore OOD types, and (3) understand OOD data.

4.1 Interactive OOD Text Detection

OOD Detection Method. Given a data instance x, an OOD detection method first computes an OOD score $s(x)$. If $s(x) > \epsilon$ (a pre-defined threshold), then x is considered as an OOD sample. DeepLens leverages a calibration-based method, *MSP* (maximum softmax probability) [21], to compute the OOD score. A higher MSP means the model is highly confident with the prediction, thus a lower MSP indicates the given data instance x is more likely to be an OOD sample. Given a probabilistic classifier C,

$$s(x) = 1 - \max_k C(y = k|x) \quad (1)$$

where $k \in 1, \ldots, N$ denotes class label k and y denotes the prediction of C. Note that a probabilistic classifier typically exists in an NLP model even if it is not for classification. For instance, a probabilistic classifier exists when projecting hidden states into the vocabulary.

Distribution View. DeepLens allows users to adjust the threshold of OOD detection dynamically via adjusting the slider (Fig. 3 Ⓐ). When the threshold is updated, the *icon arrays* (Fig. 3 Ⓑ) will also be updated accordingly. We chose to use *icon arrays* since it provides a discrete-event representation and has been proven to lead to more accurate interpretation of numbers and require lower numeracy, compared with alternative visualizations such as pie charts and bar charts [13, 28]. Users can inspect icon arrays at a glance to quickly understand how many OOD samples might exist in the test data. To validate if an optimal threshold is set, users can check the OOD score distribution (Fig. 3 Ⓒ) across training and test data. Ideally, an optimal threshold should distinguish ID and OOD data as accurately as possible. DeepLens provides a pre-computed threshold to help non-experts efficiently decide the threshold.

Instance View. This view contains two separate scrollable data grids of the ID and OOD data (Fig. 4). When users change the threshold of the OOD score, the instance view will update accordingly. The rows of each data grid are individual data instances, and the columns are: *index*, *model prediction result*, *clustering result*, *raw text*, and *OOD score* of each data instance. By default, the data grids including *OOD data* and *ID data* are sorted in descending and ascending orders according to OOD scores, respectively. Users can also filter, search, or sort each data grid to explore a data instance.

4.2 OOD Text Categorization and Exploration

Text Clustering. To help users efficiently explore topics of detected OOD samples, DeepLens uses a text clustering algorithm to categorize different types of texts. Algorithm 1 depicts the clustering algorithm. Given an NLP model M, the algorithm first extracts hidden features for each text of new test data (Line 1-5). While the extracted features $\mathcal{F}$ are usually sparse and high-dimensional vectors, DeepLens applies *PCA* (principal component analysis) [27] to reduce their dimensions to p (line 6). Then, DeepLens uses *KMeans* clustering algorithm [36] to cluster processed hidden features $\mathcal{F}_p$ (Line 7-9). To decide an optimal number of clusters n_{opt}, DeepLens leverages *Silhouette* method [53] (Line 10). During implementation,

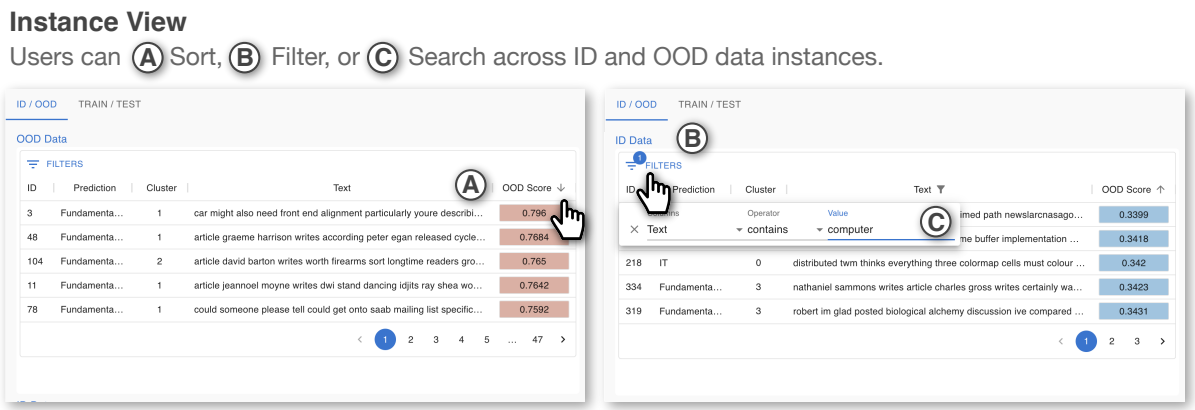

Figure 4: Interaction with the *Instance View*.

Algorithm 1: A semantic text clustering algorithm.

Input: an NLP model M, new test data $\mathcal{X}$, maximum number of clusters N_{max}, PCA dimension p
Output: cluster label $\mathcal{Y}$

1 $\mathcal{F} \leftarrow \{\}$;
2 **for** x *in* $\mathcal{X}$ **do**
3 $\quad f \leftarrow$ extract_features(M, x);
4 $\quad \mathcal{F}$.append(f);
5 **end**
6 $\mathcal{F}_p \leftarrow \text{PCA}(\mathcal{F}, p)$;
7 **for** $n \leftarrow 1, \ldots, N_{max}$ **do**
8 $\quad \mathcal{Y}_n = \text{KMeans}(\mathcal{F}_p, \text{n})$;
9 **end**
10 $n_{opt} \leftarrow \max_n \text{Silhouette}(\mathcal{Y}_n, \mathcal{F}_p)$;
11 **return** $\mathcal{Y}_{n_{opt}}$;

we set the maximum number of clusters as $N_{max} = 200$, and the PCA dimension as $p = 128$. These numbers are decided empirically.

Keywords Summarization. After each cluster is determined, DeepLens summarizes a few keywords from an individual cluster to help users identify its potential topic(s). To achieve this, DeepLens first filters out "stop words" [6] from each data and then uses CountVectorizer algorithm [48] to extract keywords. For each cluster, DeepLens displays top-10 frequent keywords as a word cloud. Through inspecting the word cloud, users can quickly understand what kind of text patterns the selected cluster might include.

Cluster View. DeepLens integrates the clustering results and summarized keywords in the cluster view (Fig. 5). The cluster view consists of a scatter plot and a word cloud. Each node in the scatter plot represents an individual data instance. The position of each node is determined by the first three components of the hidden features of each text after PCA. The color assigned to each node represents the cluster index. When users hover on a node, a tool-tip will pop up showing the prediction label and OOD score of the corresponding data instance (Fig. 5 Ⓑ). When users click on a node, the corresponding data instance will also be selected in the instance view. These features allow users to contextualize the clustering results with specific data instances and texts. Users can also focus on one cluster by clicking on the legend (Fig. 5 Ⓐ). Once a cluster is selected, DeepLens will update the word cloud (Fig. 5 Ⓒ) and filter out data excluded in the selected cluster in the instance view.

4.3 OOD Text Explanation

Salient Words Selection. The previous sections introduce how users can efficiently inspect OOD issues and identify potential OOD types. By inspecting the cluster view, users might already have hypotheses about potential OOD types and their topics based on several data instances. DeepLens further supports digging into specific instance(s). To achieve this, DeepLens uses neuron activation analysis [23] to select salient words in a text. By checking small groups of highlighted words, users can avoid reading a long paragraph of text in detail. We describe the algorithm of *salient words selection* in Algorithm 2. At a high level, DeepLens leverages ecco [1] to extract and factorize neuron activation information. For a given text x, DeepLens first extract neuron activation values $\mathcal{A}$ by passing it through a large pre-trained language model M (Line 2). Then DeepLens uses Non-negative matrix factorization to factorize the extracted activation values into n components (Line 3). In this way, DeepLens can group similar words in a text into n groups. To further reduce users' mental demands, we filter out groups containing stop words or special tokens (e.g., punctuation) (Line 4-14). Finally, for each group, DeepLens only highlights 10 words with the highest activation values. This helps preserve only the most important words in a group. In our implementation, we use a pre-trained BERT [12] released on HuggingFace [2] without any fine-tuning. The number of factors n is set to 10, and only top-10 salient words are highlighted.

Highlighting View. In this view, DeepLens leverages the visualization of ecco [1] to visualize highlighted salient words (Fig. 6). The sparklines on the left of each text box visualize the positions of similar salient words in the text. Each sparkline represents a group

[2] https://huggingface.co/bert-base-uncased

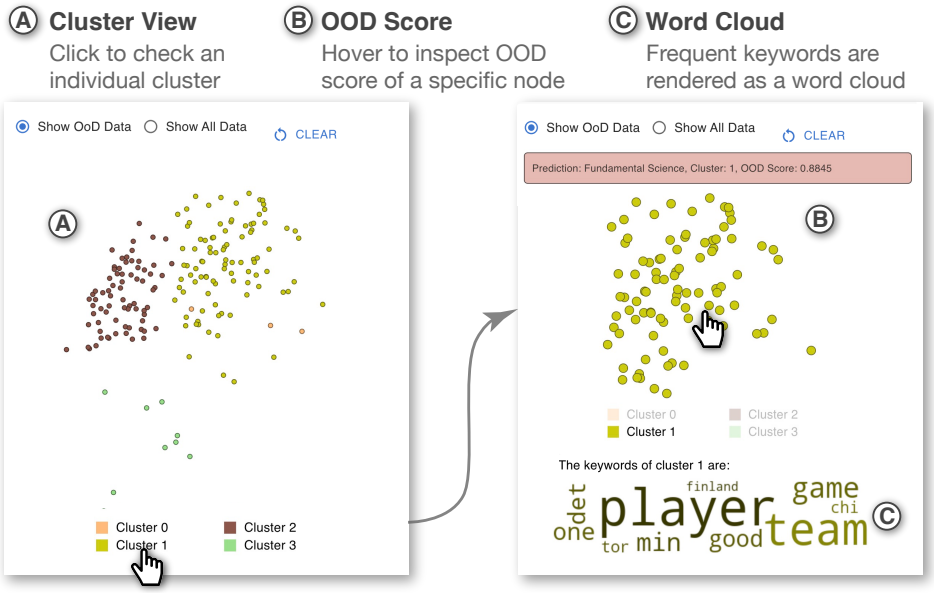

Figure 5: Users can inspect and interact with text clustering results in the *Cluster View*.

Algorithm 2: Salient words selection.

Input: A text input x, a pre-trained language model M, number of factors n, number of words in text l
Output: Highlighted keywords $\mathcal{S}$

$\mathcal{S} \leftarrow \emptyset$;
$\mathcal{A} \leftarrow$ get_activation(M, x);
$\tilde{\mathcal{S}} \leftarrow$ NMF$(\mathcal{A}, n)$;
for $i \leftarrow 1, \ldots, n$;
do
  for $j \leftarrow 1, \ldots, l$;
  do
    if x_j *is stop word or special tokens* **then**
      $\mathcal{S}_i \leftarrow \emptyset$;
    else
      $\mathcal{S} \leftarrow \mathcal{S} \bigcup \tilde{\mathcal{S}}_i$;
    end
  end
end
return $\mathcal{S}$;

of similar salient words. The color of a sparkline is the same as the color of the corresponding group of salient words. The x-axis is the index of a word in the text, and the y-axis indicates the saliency score of a word. Users can hover on different lines to inspect different groups of salient words. For each group of salient words, a darker color indicates a higher activation value. While previous studies have shown that it is important to allow users to understand a concept through a contrastive way [41], DeepLens allows users to pin multiple ID or OOD instances at the same time to contextualize the OOD topics. When users click a node in the *cluster view* or an instance in the *instance view*, this instance will be pinned at the top of the *instance view*. At the same time, the salient words of each selected instance will be displayed in the *highlighting view*.

4.4 Implementation

We implement and deploy DeepLens as a web application. The interface of DeepLens (Fig. 2) is implemented with Material UI [3]. We use D3.js [4] for visualizing scatter plots. All machine learning models were implemented with PyTorch and Scikit-learn and trained on one NVIDIA A6000 GPU. We deployed DeepLens on an AWS EC2 for ease of access during the user study.

5 USAGE SCENARIO

Suppose Alice is a model developer and she has trained an ML model to classify text documents into two different topics: *IT (information technology)* and *Fundamental Science*. Her model achieves 94% accuracy on the training data. However, her model's performance significantly drops when Alice deploys it online. Alice suspects such performance degradation is caused by out-of-distribution (OOD) issues in the new data. Alice runs an OOD detection program on the new data and finds that 49% of the new data (400 data instances) are detected as OOD data. Alice wants to understand the characteristics of these data, so she can strategically apply data augmentation techniques to improve the training data. However, she finds it time-consuming to manually check the 400 OOD samples and understand why they are categorized as OOD. Furthermore, since each text document is lengthy, Alice finds it hard to glance it over and quickly understand the gist of the document.

Alice decides to give DeepLens a try. She first checks the *Distribution View*, where two icon arrays (Fig. 3 Ⓑ) showing the proportion

[3] https://mui.com
[4] https://d3js.org

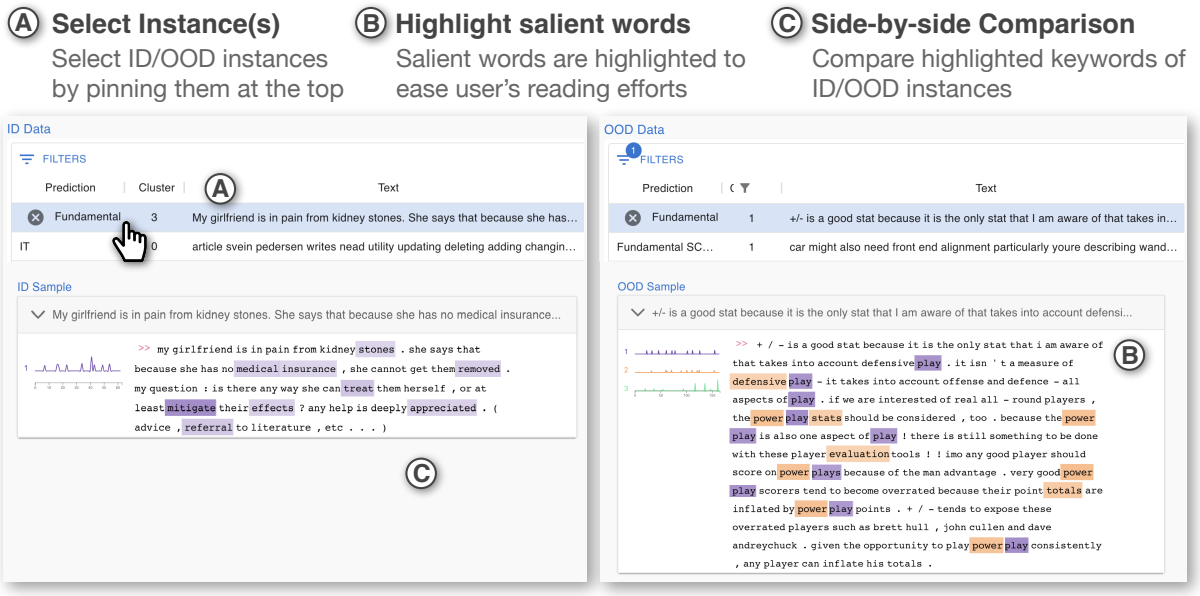

Figure 6: The *Highlighting View* helps users quickly understand specific ID/OOD data instances.

of in-distribution and out-of-distribution samples in training and test data, respectively. She finds that compared with training data, a large proportion of test data are out-of-distribution samples. She also attends to the *OOD score distribution* (Fig. 3 Ⓒ), where the OOD score distribution of training and test data are different. She confirms that OOD samples exist in her test data and could have caused her model's performance degradation.

Alice wonders what kinds of instances are detected as OOD samples by DeepLens. Therefore she turns to the *Instance View* (Fig. 2 Ⓑ), where the OOD and ID data are displayed in two separate data grids. Alice finds that instances with high OOD scores are very suspicious. Since browsing each data instance will take too much time, Alice decides to switch to the *Cluster View* (Fig. 2 Ⓒ) to get an overview first. This view contains a scatter plot where each data instance is rendered as a node, and similar data instances are clustered and colored with the same color. By default, the *Cluster View* only shows OOD data. Alice finds it obvious that each cluster contains a different number of OOD nodes. Then she explores the exact number of OOD nodes in each cluster by hovering over the legends. There are many OOD nodes in Cluster 1 and 2, but much fewer in Cluster 0 and 3 (Fig. 5 Ⓐ). Alice thinks Cluster 1 and 2 may include two new OOD types.

Now Alice wants to take a deeper look at Cluster 1. By clicking the corresponding legend, the *Cluster View* is updated and only displays OOD nodes of Cluster 1 (Fig. 5 Ⓑ). The *Instance View* also filters out the data instances that are not in Cluster 1. The *word cloud* shows that Cluster 1 has several frequent keywords, e.g., "players", "game", "team" (Fig. 5 Ⓒ). Alice realizes the topic of this cluster could be *Sports*. This is an obvious sign of a new category in the OOD data, which is not initially included in the training data.

Alice clicks a node in Cluster 1. Then, the selected instance is pinned at the top of the *Instance View*. While the selected sentence is very long and takes time to read, Alice decides to check the highlighted words in the *Highlighting View* (Fig. 6 Ⓑ). These salient words help her focus on the essential information and ignore the unnecessary words in the sentence. Alice notices that there are a few words highlighted, e.g., "defensive", "power play". These keywords further confirm Alice's belief that a new category is *Sports*.

To further validate her belief, Alice continues to check whether this category exists in the in-distribution data. She clicks on the first sentence in the ID table of the *Instance View*, which is predicted as *Fundamental Science* (Fig. 6 Ⓐ). The salient words highlighted in this sentence are: "medical", "stones", "mitigate", and "treat" (Fig. 6 Ⓒ). All these words are usually from *Fundamental Science* articles. Then, Alice clicks into several other OOD instances near the current OOD instance and confirms that *Sports* is a new topic in the OOD data.

6 USER STUDY

To evaluate the effectiveness and usability of DeepLens, we conducted a within-subjects user study with 24 programmers with various levels of machine learning expertise. To better understand the value of proposed features in DeepLens, we compared DeepLens with a variant of DeepLens as the baseline by disabling the *Cluster View* and the *Highlighting View*.

6.1 Participants

We recruited 24 participants through mailing lists of the ECE and CS departments at the University of Alberta [5]. All participants have basic knowledge about machine learning. 10 participants were Master students, 10 were Ph.D. students, 3 were professional developers, and 1 was a data scientist. Participants were asked to self-report their machine learning expertise. 12 participants had 2-5 years of

[5]This human-participated study is approved by the university's research ethics office.

Table 1: Designed tasks for the user study.

#	Task	Description	ID data	OOD data	Distribution shift type
1	Topic classification	Predict topic of a paragraph of text from Wikipedia.	DBPedia top-4 [64]†	DBPedia rest [64]†	Semantic shift
2	Sentiment analysis	Predict sentiment of a review text.	IMDB [39]	Yelp [64]	Background shift
3	Emotion recognition	Recognize the emotion from a given text.	Emotion Negative-2 [54]‡	Emotion rest [54]‡	Semantic shift
4	Fake news detection	Detect if a news article is real or fake.	PolitiFact [55]	COVID-19 Fake News [11] GossipCop [55]	Background shift

† DBPedia dataset has 14 classes. We denote DBPedia top-4 as a subset including the first 4 classes (Company, Educational Institution, Artist, and Athlete) according to class IDs, and DBPedia rest as a subset including the other 10 classes.
‡ Emotion dataset has 6 classes. We denote Emotion Negative-2 as a subset including 2 negative Emotion classes (Sadness and Fear), and Emotion rest as a subset including the other 4 classes.

experience, 1 had more than 5 years, and 11 only had 1 year. Regarding NLP experience, 5 participants had 2-5 years of experience, and 19 only had 1 year. 20 participants mentioned that they had heard about out-of-distribution or distribution shift problems before. All study sessions were conducted on Zoom. Both DeepLens and baseline were deployed as web applications, therefore participants were able to access our study sessions from their own PCs.

6.2 Tasks

We designed four tasks that cover different kinds of distribution shifts in the NLP domain. Table 1 shows the details of each user study task. When designing tasks, we follow these requirements: (1) the tasks should be representative ones in the NLP domain and (2) the tasks should cover two different types of distribution shift. To achieve these goals, we collected four tasks from prior work about OOD detection in NLP and well-known public benchmarks for NLP models. For each task, we adopt a BERT model [12] as the backbone and fine-tune its performance on the ID data. For a fair comparison, the baseline tool and DeepLens use the same pre-computed OOD threshold on each task. More details such as models' training settings and example interfaces for each task can be found in Appendix A.

6.3 Protocol

Each user study session took about 60 minutes. At the beginning of each session, we asked participants for their consent to record. Participants were assigned two tasks about identifying OOD issues, one to be completed with DeepLens and the other to be the baseline tool. To mitigate the learning effect, both task assignment order and tool assignment order were counterbalanced across participants. In total, 6 participants experienced each task. Participants were asked to watch a 5-min tutorial video of the assigned tool before starting each task, followed by a 5-min practice period to familiarize themselves with the tool. Then, participants were given 20 minutes to use the assigned tool to inspect and identify OOD issues within the given model and dataset. In particular, participants were asked to answer/report:

(1) What kind of data distribution shift does it belong to?
(2) How many different types of OOD data did you find?
(3) For each different type of OOD, please explain why you think it is OOD and list the indexes of the OOD instance that belong to this type.

After completing each task, participants filled out a post-task survey to give feedback about what they liked or disliked. Participants were also asked to answer five NASA Task Load Index (TLX) questions [19] as a part of the post-task survey. After completing both two tasks, participants filled out a final survey, where they directly compared two assigned tools. At the end of the study session, each participant received a $25 Amazon gift card as compensation for their time.

7 RESULTS

In this section, we report and analyze the difference in participants' performance when using DeepLens and the baseline tool. We denote the participant as P# in the following.

7.1 User Performance

Table 2 shows participants' performance on four tasks of identifying OOD issues. We found that all 24 participants using DeepLens correctly identified the type of data distribution shift (i.e., background vs. semantic shift) in the assigned model, while 2 out of 24 participants using the baseline method failed.

To further assess participants' performance, we manually inspected participants' answers to check their correctness. A correct OOD type should (1) include the word(s) that are representative of a group of data instances, and (2) be significantly different from ID data. Overall, we found that participants using DeepLens were able to find more types of OOD on all four different tasks compared with participants using the baseline tool. Regarding Task 1 and Task 3 (semantic shift), the average number of OOD types found by each participant using DeepLens is 7 and 3.5 respectively. By contrast, the average number is 2.3 and 1.2 respectively when using the baseline tool. The Welch's t-test suggests that the performance differences are significant in both cases (p-value < 0.001). Regarding Task 2 and Task 4 (background shift), most participants using the baseline tool were only able to find 1 type of OOD (mean: 1

Table 2: User performance in four different tasks.

		Task 1 Semantic Shift		Task 2 Background Shift		Task 3 Semantic Shift		Task 4 Background Shift	
		Baseline	DeepLens	Baseline	DeepLens	Baseline	DeepLens	Baseline	DeepLens
# of participants correctly identified shift type		5	6	5	6	6	6	6	6
# of correct OOD Types found per participant	Min	1/10	4/10	1/2	1/2	0/4	3/4	0/2	2/2
	Med	2/10	7.5/10	1/2	2/2	1/4	3.5/4	0.5/2	2/2
	Max	4/10	8/10	1/2	2/2	2/4	4/4	1/2	2/2
	Mean	2.3/10	7/10	1/2	1.7/2	1.2/4	3.5/4	0.5/2	2/2
Welch's t-test		$\Delta = 4.7$, $t = 6.14,\ p < 0.001$		$\Delta = 0.7$, $t = 3.16,\ p = 0.010$		$\Delta = 2.3$, $t = 6.14, p < 0.001$		$\Delta = 1.5$, $t = 6.71, p = 0.001$	

and 0.5 respectively). By contrast, participants using DeepLens were able to find 1.7 and 2 types of OOD on average in these two tasks respectively. The Welch's t-test suggests that the performance differences are significant (p-value = 0.010 and p-value = 0.001).

We analyzed the post-task survey responses and the recordings to understand why participants using DeepLens performed better. We found that DeepLens users' success mainly come from the *Cluster View* and the *Highlighting View*. First, the cluster view significantly sped up the process of finding OOD types. 23 out of 24 participants had heavily utilized the clustering view to explore OOD data. By contrast, participants using the baseline tool had to inspect OOD instances one by one. P16 wrote, "*[when using the baseline tool], it is tedious to go through all the data point one by one especially when there are a lot of them.*" In the post-task survey, 22 out of 24 participants also agreed that the clustering results were helpful. P17 said, "*by going through clusters, I can find trends faster than by going through individual data points.*" Besides, the summarized keywords of each cluster were also found helpful. Based on the recordings, 19 out of 24 participants started their exploration from these keywords. By contrast, participants using the baseline tool usually started their exploration by randomly picking an instance. P20 said, "*[when using DeepLens,] I can use the keywords extracted in the cluster and put that in the filter to find more OOD instances of the same type.*" P21 commented, "*In my using experience, [DeepLens] helps me a lot in quickly summarizing background shift keywords.*

In addition, the highlighting view in DeepLens helped participants avoid incorrect OOD types. In our user study, the median number of incorrect OOD types found per participant using DeepLens is 0, while the corresponding number of participants using the baseline tool is 1. The mean difference of incorrect OOD types is 0.52 vs. 1.64 (Welch's t-test: $p = 0.009$). One specific reason is that when using DeepLens, participants were able to compare the OOD data with the ID data to confirm a new OOD type. In the post-task survey, 17 out of 24 participants marked the comparison of ID and OOD data as helpful. Furthermore, when comparing the ID and OOD data, the highlighted keywords also helped participants avoid misunderstanding a long text. In the post-task survey, 18 out of 24 participants agreed that seeing the highlighted keywords was helpful. P9 said, "*when verifying my hypothesis about whether a certain sentence belongs to OOD data, DeepLens is helpful because it shows several highlighted keywords and reduces my time consumption.*" By

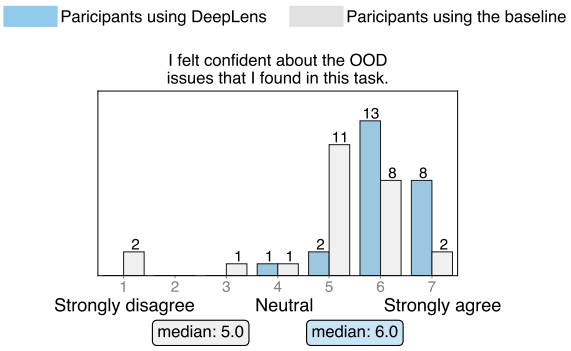

Figure 7: Users' self-ratings about their task performance.

contrast, P11 commented, "*It is not easy to read the whole text [when using the baseline tool].*"

We have also further analyzed the impact of the OOD threshold adjustment feature on user performance. We found that only two users (P6 and P14) had tried to adjust the threshold, and they eventually reset it to the pre-computed one. A plausible explanation is that the default threshold has already provided a good starting point for users to investigate OOD issues. Thus, we believe the OOD threshold in the distribution view may have little impact on user performance. Since this threshold adjustment feature is present in both conditions, the better user performance of DeepLens comes directly from the cluster and highlighting views.

7.2 User Confidence and Cognitive Overhead

In the post-task survey, participants self-reported their confidence about OOD issues they identified with help of the assigned tool in two different 7-point Likert scale questions. Figure 7 shows participants' assessments when using DeepLens and the baseline tool. We found that participants using DeepLens were more confident about the OOD issues they found, where the median confidence ratings are 6 vs. 5. The mean difference is 1.12 (6.17 vs. 5.04), which is statistically significant (Welch's t-test: p-value = 0.002). This confidence improvement was largely attributed to DeepLens's ***Cluster View***. P5 commented, "*The automatic clustering function works quite well, and the keyword summary is quite useful to have an overview.*" P16 said, "*By clustering the data, DeepLens makes keywords in the word cloud better indicators when identifying OOD issues.*

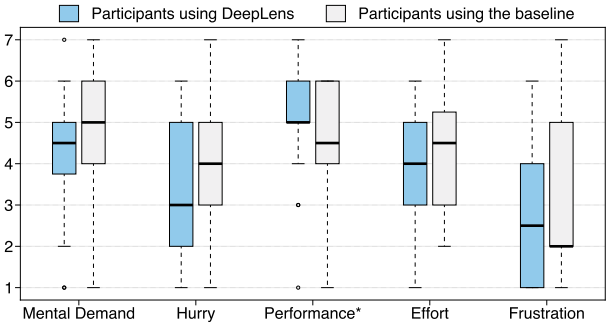

Figure 8: NASA Task Load Index Ratings. Entries with a star (*) mean the mean difference is statistically significant.

Figure 8 shows participants' ratings on the five cognitive factors of the NASA TLX questionnaire. Though DeepLens has more features and renders more information, we find that there was no significant difference when using DeepLens and the baseline tool in terms of mental demand, hurry, effort, and frustration (Welch's t-test: $p = 0.22$, $p = 0.24$, $p = 0.26$, $p = 0.75$). However, participants using DeepLens felt they have better performance compared with participants using the baseline tool (mean difference: 0.88, Welch's t-test: $p = 0.049$). This indicates that DeepLens is more effective and useful when helping users inspect and identify OOD issues in an ML model compared with the baseline tool.

7.3 User Ratings of Individual Features

In the post-task survey, participants rated the key features of DeepLens. Among 24 participants, 23 participants indicated that they would like to use DeepLens when solving OOD problems in their own ML models, while 1 participant stayed neutral. The median rating is 6 on a 7-point Likert scale (1—I don't want to use it at all, 7—I will definitely use it). As shown in Fig. 9, participants felt DeepLens's interface and interactive features intuitive and helpful. The ***Cluster View*** is most appreciated by participants. 22 out of 24 participants agreed that "*it was helpful to see the clustering results.*" The median rating of the cluster view is 6. P17 commented, "*I like that it [DeepLens] had clustering, for instance, it was super quick to find "covid" and "entertainment" OOD categories using this feature.*" P20 commented, "*DeepLens has the cluster and keyword visualization which can help me identify a type of OOD quickly.*" Besides, 18 out of 24 participants agreed that "*seeing highlighted keywords was helpful.*" The median rating of the ***Highlighting View*** is 6. P24 commented in the post-task survey, "*[DeepLens] brings me less reading and easy to focus on the details*" 17 out of 24 participants also found comparing ID and OOD data helpful (median rating: 6).

7.4 User Preference and Feedback

In the final survey, participants self-reported their preference between DeepLens and the baseline tool (Figure 10). 23 out of 24 participants reported that DeepLens was more helpful (median rating: 6) and they preferred to use it in practice (median rating: 6). We coded participants' responses to this question and identified 2 different themes. First, 17 participants mentioned that the ***Cluster View*** makes it easier when identifying and analyzing OOD issues. P23 commented, "*keywords [in the cluster view] are useful when facing a large dataset.*" P13 said, "*[the cluster view] gives a visual representation of the data, which makes it easier to identify the OOD data.*" Second, 5 participants credited their success to the ***Highlighting View***. P9 said, "*highlighted keywords reduce my time consumption and make it easy to tell whether a data instance is OOD.*"

Participants also pointed out some limitations in the current form of DeepLens. 2 participants commented that it would be better if the keywords summarization in DeepLens could be improved. P21 said, "*there are some meaning-less high-frequency words which might disturb.*" 4 participants suggested improving the usability of filters in the instance view, e.g., by allowing users to add multiple filter conditions at the same time. 1 participant mentioned that a semantic word search function (i.e., matching semantically similar words instead of identical ones) would assist their exploration process.

8 DISCUSSION

8.1 Design Implications

The user study results suggest that, with the help of DeepLens, users are able to find more types of OOD data with more confidence compared with using the baseline tool. Though addressing OOD issues is an urgent topic for deploying safe and reliable AI services [52], most efforts have been devoted to improving the algorithm accuracy of OOD detection. Our work indicates that only detecting OOD samples is not sufficient for improving model developers' productivity, especially when the dataset is large and the types of OOD are diverse. It is equally important to facilitate developers to understand and explore different types of OOD data in large text corpora. Once developers have gained deep insights of the OOD data in their datasets, they can further make strategic decisions to improve the model, e.g., data augmentation or selection for model retraining.

During the continuous delivery of machine learning models, the number of newly collected data can be massive. To reduce the cognitive effort of exploring different types of OOD instances in the new data, it is essential to summarize a small number of potential OOD types for developers. DeepLens addresses this by leveraging a text clustering algorithm. Furthermore, the interactive cluster exploration support in DeepLens preserves the user's control over verifying each type of OOD data. This is aligned with one of the human-AI interaction guidelines—*providing several suggestions instead of fully automating the process* [2]. Based on the user study results, we find that such a semi-automated process of exploring OOD data improves participants' performance and confidence.

While clustering results do not directly tell users what exactly an OOD type is, the summarized keywords from each cluster serve as the starting point for exploration. In the final-study survey of our user study, 7 out of 23 participants who preferred to use DeepLens in practice explicitly mentioned how summarized keywords had assisted them. Previous work [9] has shown that one important requirement when designing an interactive system for OOD detection in image data is *examining OOD samples in the context of normal samples*. DeepLens supports this by allowing users to compare ID text and OOD text side by side. Furthermore, compared with image

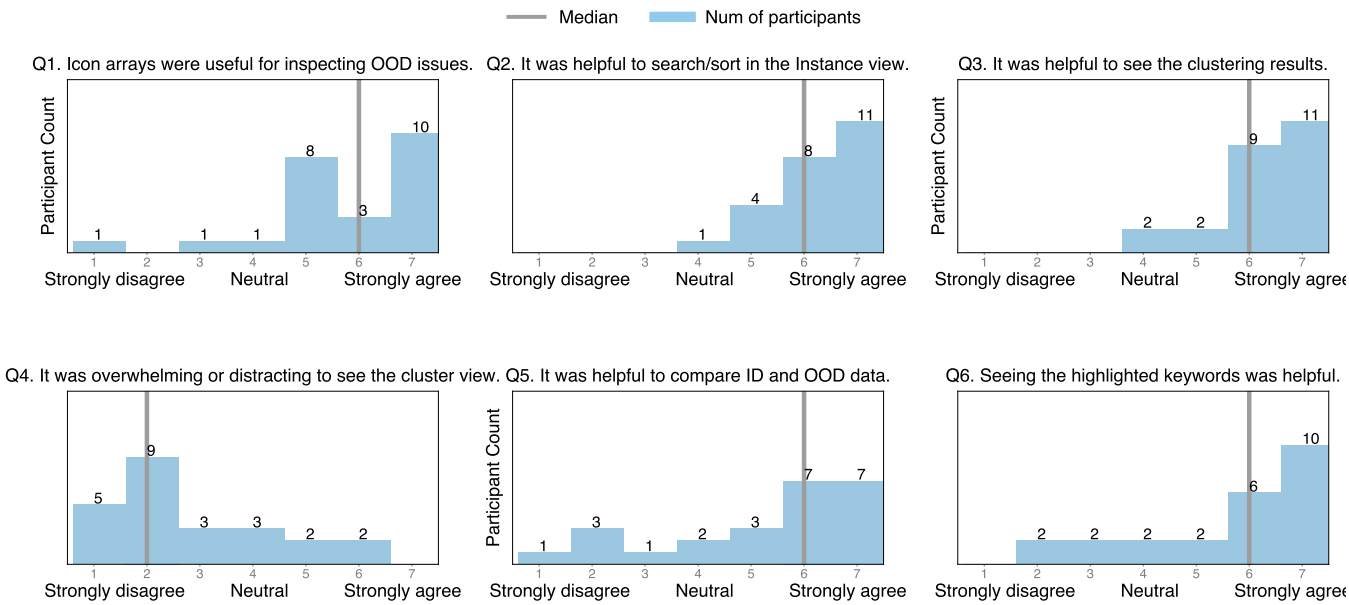

Figure 9: Users' ratings on individual features of DeepLens.

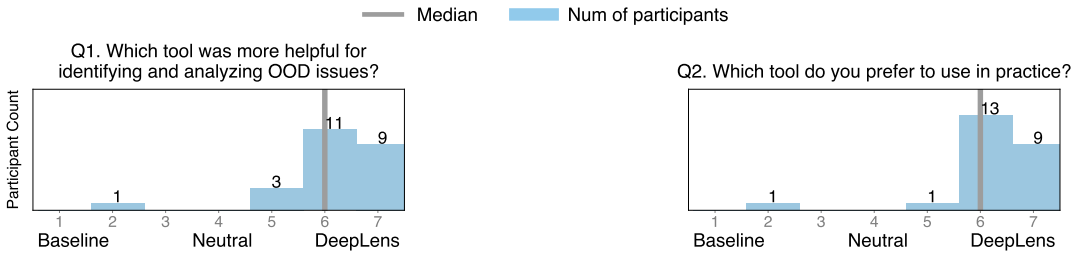

Figure 10: Users' preference on tool choice for OOD detection.

data, text documents are less glanceable. Therefore, DeepLens highlights the salient words in each text document to help developers quickly grasp the gist of each document.

8.2 Target User Groups and Use Cases

DeepLens is designed for users who know ML but are not familiar with OOD issues. In our user study, most participants have heard about OOD issues but have not worked on OOD issues before. 4 out of 24 participants even reported that they had never heard about OOD before. Our user study results suggest that these participants performed better when using DeepLens compared with using the baseline tool (mean number of OOD types found per participant: 3.5 vs. 1.3, Welch's t-test: $p < 0.0001$). Furthermore, they also felt more confident with the OOD issues they identified (median rating of confidence: 6 vs. 5). While experts may be more likely to identify OOD types by reading raw text data, they still appreciated DeepLens since it automates some of their work. For instance, the clustering in DeepLens automates the process of categorizing similar texts for them. P17 wrote, "*by going through these clusters, I can find trends faster than by going through individual data points.*" P5 commented, "*DeepLens automated some of the manual work, and I found that my productivity is improved. I can get more work done within the same amount of time.*"

DeepLens is specifically designed for debugging OOD issues for NLP models. Therefore, our findings and design implications cannot be generalized to other kinds of ML issues, such as gradient vanishing. In addition, DeepLens can also be deployed as an online tool to continuously monitor potential data distribution shifts for deployed models.

8.3 Limitations and Future Work

In addition to the limitations and suggestions pointed out by our user study participants (Sec. 7.4), there are several other limitations to our user study design and system.

User Study Baseline. In our current form of user study, a variant of DeepLens was created as the baseline method by disabling the cluster view and the highlighting view. However, this cannot distinguish the contribution of individual features to user performance improvement. One can consider creating more variants of

DeepLens by disabling individual features as the comparison baselines. One can also consider instrumenting the tool and measuring the utility rate of each feature during user study sessions.

Limited NLP Tasks. Our user study cannot confirm whether DeepLens works for all types of NLP tasks. To comprehensively evaluate the usefulness of DeepLens, one can consider using DeepLens to identify OOD issues in more diverse NLP tasks, e.g., question answering and natural language inference.

Accuracy of OOD Detection. Currently, DeepLens leverages *MSP* [4] as the OOD detection algorithm. Though *MSP* has been proven effective in several NLP tasks [4], it may not always be applicable to other kinds of NLP tasks or models. Since the design of DeepLens is not limited to a specific type of OOD detection algorithm, one future direction could be to integrate more OOD detection algorithms to DeepLens and allow users to switch between different algorithms.

Scalability Issue. Based on our user study results, DeepLens can handle 1,000-4,500 data points. However, once the data is scaled up (e.g., millions of data points), nodes in the cluster view may overlap with each other. To address this issue, one can leverage more advanced visualization techniques such as Bubble Treemaps [18] to visualize clustering results hierarchically.

Alternative Algorithms and Design. Our cluster view can be further improved by using more advanced dimension reduction and clustering algorithms. In the current version of DeepLens, we choose PCA for dimension reduction and K-Means for text clustering since they are classical and common choices. However, more advanced dimension reduction methods, e.g., t-SNE [24], Isomap [56] could potentially lead to better dimension reduction results. Besides, our text clustering can also be improved with methods that are specialized for topic modeling, e.g. ConceptScope [63] and TopicNets [16]. Finally, our highlighting view can potentially be improved by replacing neuron activation analysis with other interactive tools for selecting and visualizing salient words in text data, e.g., exBERT [25].

9 CONCLUSION

In this paper, we present a novel interactive system, DeepLens, to help ML developers detect, explore, and understand potential OOD (out-of-distribution) issues in NLP models. DeepLens leverages a text clustering algorithm to help users efficiently identify and explore potential types of OOD in large-scale text data. Furthermore, DeepLens integrates a neuron activation analysis-based algorithm to highlight salient words in an individual data instance to help users quickly understand a text without reading it in detail. We implemented DeepLens as a web application and conducted a within-subjects user study with 24 ML developers on four different NLP tasks. The results show that with the help of DeepLens, developers were able to have a better understanding of OOD issues in ML models and identify more types of OOD data confidently compared with using the baseline tool. In the end, we discuss the design implications from DeepLens and propose several promising future directions.

ACKNOWLEDGMENTS

We would like to thank all anonymous participants in the user study and anonymous reviewers for their valuable feedback. This work was supported in part by Amii RAP program, Canada CIFAR AI Chairs Program, the Natural Sciences and Engineering Research Council of Canada (NSERC No.RGPIN-2021-02549, No.RGPAS-2021-00034, No.DGECR-2021-00019), as well as JSPS KAKENHI Grant No.JP20H04168, JST-Mirai Program Grant No.JPMJMI20B8.

REFERENCES

[1] J Alammar. 2021. Ecco: An Open Source Library for the Explainability of Transformer Language Models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*. Association for Computational Linguistics.

[2] Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N Bennett, Kori Inkpen, et al. 2019. Guidelines for human-AI interaction. In *Proceedings of the 2019 chi conference on human factors in computing systems*. 1–13.

[3] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. 2016. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565* (2016).

[4] Udit Arora, William Huang, and He He. 2021. Types of Out-of-Distribution Texts and How to Detect Them. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 10687–10701.

[5] Koby Bibas, Meir Feder, and Tal Hassner. 2021. Single Layer Predictive Normalized Maximum Likelihood for Out-of-Distribution Detection. *Advances in Neural Information Processing Systems* 34 (2021), 1179–1191.

[6] Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.".

[7] Julian Bitterwolf, Alexander Meinke, and Matthias Hein. 2020. Certifiably adversarially robust detection of out-of-distribution data. *Advances in Neural Information Processing Systems* 33 (2020), 16085–16095.

[8] Tom Bocklisch, Joey Faulkner, Nick Pawlowski, and Alan Nichol. 2017. Rasa: Open source language understanding and dialogue management. *arXiv preprint arXiv:1712.05181* (2017).

[9] Changjian Chen, Jun Yuan, Yafeng Lu, Yang Liu, Hang Su, Songtao Yuan, and Shixia Liu. 2020. Oodanalyzer: Interactive analysis of out-of-distribution samples. *IEEE transactions on visualization and computer graphics* 27, 7 (2020), 3335–3349.

[10] Sungik Choi and Sae-Young Chung. 2020. Novelty detection via blurring. *Proceedings of the International Conference on Learning Representations (ICLR)* (2020).

[11] Sourya Dipta Das, Ayan Basak, and Saikat Dutta. 2021. A heuristic-driven ensemble framework for COVID-19 fake news detection. In *International Workshop on Combating Online Hostile Posts in Regional Languages during Emergency Situation*. Springer, 164–176.

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[13] Mirta Galesic, Rocio Garcia-Retamero, and Gerd Gigerenzer. 2009. Using icon arrays to communicate medical risks: overcoming low numeracy. *Health Psychology* 28, 2 (2009), 210.

[14] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. In *International Conference on Learning Representations*. http://arxiv.org/abs/1412.6572

[15] Cheryl L Grady, Anthony R McIntosh, M Natasha Rajah, and Fergus IM Craik. 1998. Neural correlates of the episodic encoding of pictures and words. *Proceedings of the National Academy of Sciences* 95, 5 (1998), 2703–2708.

[16] Brynjar Gretarsson, John O'donovan, Svetlin Bostandjiev, Tobias Höllerer, Arthur Asuncion, David Newman, and Padhraic Smyth. 2012. Topicnets: Visual analysis of large text corpora with topic modeling. *ACM Transactions on Intelligent Systems and Technology (TIST)* 3, 2 (2012), 1–26.

[17] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International conference on machine learning*. PMLR, 1321–1330.

[18] Jochen Görtler, Christoph Schulz, Daniel Weiskopf, and Oliver Deussen. 2018. Bubble Treemaps for Uncertainty Visualization. *IEEE Transactions on Visualization and Computer Graphics* 24, 1 (2018), 719–728. https://doi.org/10.1109/TVCG.2017.2743959

[19] 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*. Vol. 52. Elsevier, 139–183.

[20] Matthias Hein, Maksym Andriushchenko, and Julian Bitterwolf. 2019. Why relu networks yield high-confidence predictions far away from the training data

and how to mitigate the problem. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 41–50.
[21] Dan Hendrycks and Kevin Gimpel. 2017. A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net. https://openreview.net/forum?id=Hkg4TI9xl
[22] Dan Hendrycks, Norman Mu, Ekin D. Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. 2020. AugMix: A Simple Data Processing Method to Improve Robustness and Uncertainty. *Proceedings of the International Conference on Learning Representations (ICLR)* (2020).
[23] J Hewitt and P Liang. 2019. Designing and Interpreting Probes with Control Tasks. *Proceedings of the 2019 Con* (2019).
[24] Geoffrey E Hinton and Sam Roweis. 2002. Stochastic neighbor embedding. *Advances in neural information processing systems* 15 (2002).
[25] Benjamin Hoover, Hendrik Strobelt, and Sebastian Gehrmann. 2019. exbert: A visual analysis tool to explore learned representations in transformers models. *arXiv preprint arXiv:1910.05276* (2019).
[26] Yen-Chang Hsu, Yilin Shen, Hongxia Jin, and Zsolt Kira. 2020. Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10951–10960.
[27] Ian T Jolliffe. 2002. *Principal component analysis for special types of data*. Springer.
[28] Matthew Kay, Tara Kola, Jessica R Hullman, and Sean A Munson. 2016. When (ish) is my bus? user-centered visualizations of uncertainty in everyday, mobile predictive systems. In *Proceedings of the 2016 chi conference on human factors in computing systems*. 5092–5103.
[29] Jinhan Kim, Robert Feldt, and Shin Yoo. 2019. Guiding deep learning system testing using surprise adequacy. In *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*. IEEE, 1039–1049.
[30] David Lazer, Ryan Kennedy, Gary King, and Alessandro Vespignani. 2014. The parable of Google Flu: traps in big data analysis. *Science* 343, 6176 (2014), 1203–1205.
[31] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. 2018. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in neural information processing systems* 31 (2018).
[32] Rongjian Li, Wenlu Zhang, Heung-Il Suk, Li Wang, Jiang Li, Dinggang Shen, and Shuiwang Ji. 2014. Deep learning based imaging data completion for improved brain disease diagnosis. In *International conference on medical image computing and computer-assisted intervention*. Springer, 305–312.
[33] Shiyu Liang, Yixuan Li, and R. Srikant. 2018. Enhancing The Reliability of Out-of-distribution Image Detection in Neural Networks. In *International Conference on Learning Representations*. https://openreview.net/forum?id=H1VGkIxRZ
[34] Ziqian Lin, Sreya Dutta Roy, and Yixuan Li. 2021. Mood: Multi-level out-of-distribution detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 15313–15323.
[35] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. 2020. Energy-based out-of-distribution detection. *Advances in Neural Information Processing Systems* 33 (2020), 21464–21475.
[36] Stuart Lloyd. 1982. Least squares quantization in PCM. *IEEE transactions on information theory* 28, 2 (1982), 129–137.
[37] Christos Louizos and Max Welling. 2017. Multiplicative normalizing flows for variational bayesian neural networks. In *International Conference on Machine Learning*. PMLR, 2218–2227.
[38] Jie Lu, Anjin Liu, Fan Dong, Feng Gu, Joao Gama, and Guangquan Zhang. 2018. Learning under concept drift: A review. *IEEE Transactions on Knowledge and Data Engineering* 31, 12 (2018), 2346–2363.
[39] Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*. 142–150.
[40] Alexander Meinke and Matthias Hein. 2020. Towards neural networks that provably know when they don't know. *Proceedings of the International Conference on Learning Representations (ICLR)* (2020).
[41] Tim Miller. 2019. Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence* 267 (2019), 1–38.
[42] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. 2018. *Foundations of machine learning*. MIT press.
[43] Jose G Moreno-Torres, Troy Raeder, Rocío Alaiz-Rodríguez, Nitesh V Chawla, and Francisco Herrera. 2012. A unifying view on dataset shift in classification. *Pattern recognition* 45, 1 (2012), 521–530.
[44] Peyman Morteza and Yixuan Li. 2022. Provable guarantees for understanding out-of-distribution detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 8.
[45] Anh Nguyen, Jason Yosinski, and Jeff Clune. 2015. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 427–436.
[46] Matthew L Olson, Thuy-Vy Nguyen, Gaurav Dixit, Neale Ratzlaff, Weng-Keen Wong, and Minsuk Kahng. 2021. Contrastive identification of covariate shift in image data. In *2021 IEEE Visualization Conference (VIS)*. IEEE, 36–40.
[47] João Palmeiro, Beatriz Malveiro, Rita Costa, David Polido, Ricardo Moreira, and Pedro Bizarro. 2022. Data+ Shift: Supporting visual investigation of data distribution shifts by data scientists. *arXiv preprint arXiv:2204.14025* (2022).
[48] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.
[49] Zi Peng, Jinqiu Yang, Tse-Hsun Chen, and Lei Ma. 2020. A first look at the integration of machine learning models in complex autonomous driving systems: a case study on Apollo. In *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. 1240–1250.
[50] Alise J Ponsero and Bonnie L Hurwitz. 2019. The promises and pitfalls of machine learning for detecting viruses in aquatic metagenomes. *Frontiers in microbiology* 10 (2019), 806.
[51] Stephan Rabanser, Stephan Günnemann, and Zachary Lipton. 2019. Failing loudly: An empirical study of methods for detecting dataset shift. *Advances in Neural Information Processing Systems* 32 (2019).
[52] Jie Ren, Peter J Liu, Emily Fertig, Jasper Snoek, Ryan Poplin, Mark Depristo, Joshua Dillon, and Balaji Lakshminarayanan. 2019. Likelihood ratios for out-of-distribution detection. *Advances in neural information processing systems* 32 (2019).
[53] Peter J Rousseeuw. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics* 20 (1987), 53–65.
[54] Elvis Saravia, Hsien-Chi Toby Liu, Yen-Hao Huang, Junlin Wu, and Yi-Shin Chen. 2018. Carer: Contextualized affect representations for emotion recognition. In *Proceedings of the 2018 conference on empirical methods in natural language processing*. 3687–3697.
[55] Kai Shu, Deepak Mahudeswaran, Suhang Wang, Dongwon Lee, and Huan Liu. 2020. Fakenewsnet: A data repository with news content, social context, and spatiotemporal information for studying fake news on social media. *Big data* 8, 3 (2020), 171–188.
[56] Joshua B Tenenbaum, Vin de Silva, and John C Langford. 2000. A global geometric framework for nonlinear dimensionality reduction. *science* 290, 5500 (2000), 2319–2323.
[57] Sunil Thulasidasan, Gopinath Chennupati, Jeff A Bilmes, Tanmoy Bhattacharya, and Sarah Michalak. 2019. On mixup training: Improved calibration and predictive uncertainty for deep neural networks. *Advances in Neural Information Processing Systems* 32 (2019).
[58] Haoran Wang, Weitang Liu, Alex Bocchieri, and Yixuan Li. 2021. Can multi-label classification networks know what they don't know? *Advances in Neural Information Processing Systems* 34 (2021), 29074–29087.
[59] Xumeng Wang, Wei Chen, Jiazhi Xia, Zexian Chen, Dongshi Xu, Xiangyang Wu, Mingliang Xu, and Tobias Schreck. 2020. ConceptExplorer: Visual analysis of concept drifts in multi-source time-series data. In *2020 IEEE Conference on Visual Analytics Science and Technology (VAST)*. IEEE, 1–11.
[60] Weikai Yang, Zhen Li, Mengchen Liu, Yafeng Lu, Kelei Cao, Ross Maciejewski, and Shixia Liu. 2020. Diagnosing concept drift with visual analytics. In *2020 IEEE Conference on Visual Analytics Science and Technology (VAST)*. IEEE, 12–23.
[61] Anton Yeshchenko, Claudio Di Ciccio, Jan Mendling, and Artem Polyvyanyy. 2021. Visual drift detection for sequence data analysis of business processes. *IEEE Transactions on Visualization and Computer Graphics* (2021).
[62] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. 2019. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*. 6023–6032.
[63] Xiaoyu Zhang, Senthil Chandrasegaran, and Kwan-Liu Ma. 2021. Conceptscope: Organizing and visualizing knowledge in documents based on domain ontology. In *Proceedings of the 2021 chi conference on human factors in computing systems*. 1–13.
[64] Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems* 28 (2015).

A TASKS FOR USER STUDY

A.1 NLP Task 1: Topic Classification

DBPedia dataset extracts structured content from the information created in the Wikipedia project. In this task, we use DBPedia-14, collected by picking 14 non-overlapping topics from Wikipedia in 2014. We use examples from the first 4 classes as ID data and the

rest as OOD data. To simulate real-world data with semantic shift, we sample 1000 instances from the test splits of 14 topics.
In-distribution Data ID data contains 300 samples and 4 topics: Company, Educational Institution, Artist, and Athlete.
Out-of-distribution Data OOD data contains 700 samples and 10 topics: Office Holder, Mean of Transportation, Building, Natural Place, Village, Animal, Plant, Album, Film, and Written Work.
Model We fine-tune the BERT model for 1 epoch on the training split of ID data with a learning rate of 5×10^{-5} and a batch size of 16. It achieves 98.5% accuracy on the validation splits of ID data and 28% accuracy on the test data.

A.2 NLP Task 2: Sentiment Analysis

In this task, we use IMDB datasets as ID data and Yelp Polarity binary sentiment classification datasets as OOD data. The IMDB dataset contains movie reviews. The Yelp polarity dataset is formed by reviews for different businesses. Both IMDB and Yelp datasets have two labels Positive and Negative to predict the sentiment of the reviews. To create online data with background shift, we sample 1000 instances from both IMDB and Yelp test splits.
In-distribution Data 495 samples of movie reviews from IMDB dataset.
Out-of-distribution Data 505 samples of business reviews from Yelp dataset.
Model We fine-tune the BERT model for 1 epoch on the training split of ID data with a learning rate of 5×10^{-5} and a batch size of 16. It achieves 93.5% accuracy on the validation splits of ID data and 89% accuracy on the test data.

A.3 NLP Task 3: Emotion Recognition

The six basic emotions included in the Emotion dataset are Sadness, Fear, Joy, Anger, Surprise, and Love. The source of the dataset is English Twitter Messages. There are 2 columns in the dataset, mapping to emotion index (0 to 5) and text. We use examples from the Sadness and Fear classes as ID data and the rest as OOD data. To create online data with semantic shift, we sample 1000 instances from 6 topics' test splits.
In-distribution Data ID data contains 644 samples and 2 emotions: Sadness and Fear.
Out-of-distribution Data OOD data contains 356 samples and 4 emotions: Joy, Anger, Surprise, and Love.
Model We fine-tune the BERT model for 4 epochs on the training split of ID data with a learning rate of 2×10^{-5} and a batch size of 32. It achieves 99.2% accuracy on the validation splits of ID data and 31% accuracy on the test data.

A.4 NLP Task 4: Fake News Detection

In this task, we design a background shift scenario that involves fakeness detection on different types of news. FakeNewsNet is a dataset collected from two fact-checking websites: GossipCop and PolitiFact. It contains news with labels indicating its validity annotated by professional journalists and experts. PolitiFact contains news related to U.S. politics and GossipCop is formed by entertainment news and gossip news. Besides, we also use the COVID-19 Fake News dataset in this task. It contains COVID-19-related news extracted from social media such as Facebook, Twitter, etc. We use PolitiFact data as ID data. Then we combine GossipCop and COVID-19 Fake News datasets as OOD data. To create online data with background shift, we sample 4500 instances from PolitiFact, GossipCop, and COVID-19 Fake News test splits.
In-distribution Data 2000 samples of news related to U.S. politics.
Out-of-distribution Data 2500 samples of news related to gossip and COVID-19.
Model We fine-tune the BERT model for 3 epochs on the training split of ID data with a learning rate of 5×10^{-5} and a batch size of 16. It achieves 89.5% accuracy on the validation splits of ID data and 67% accuracy on the test data.

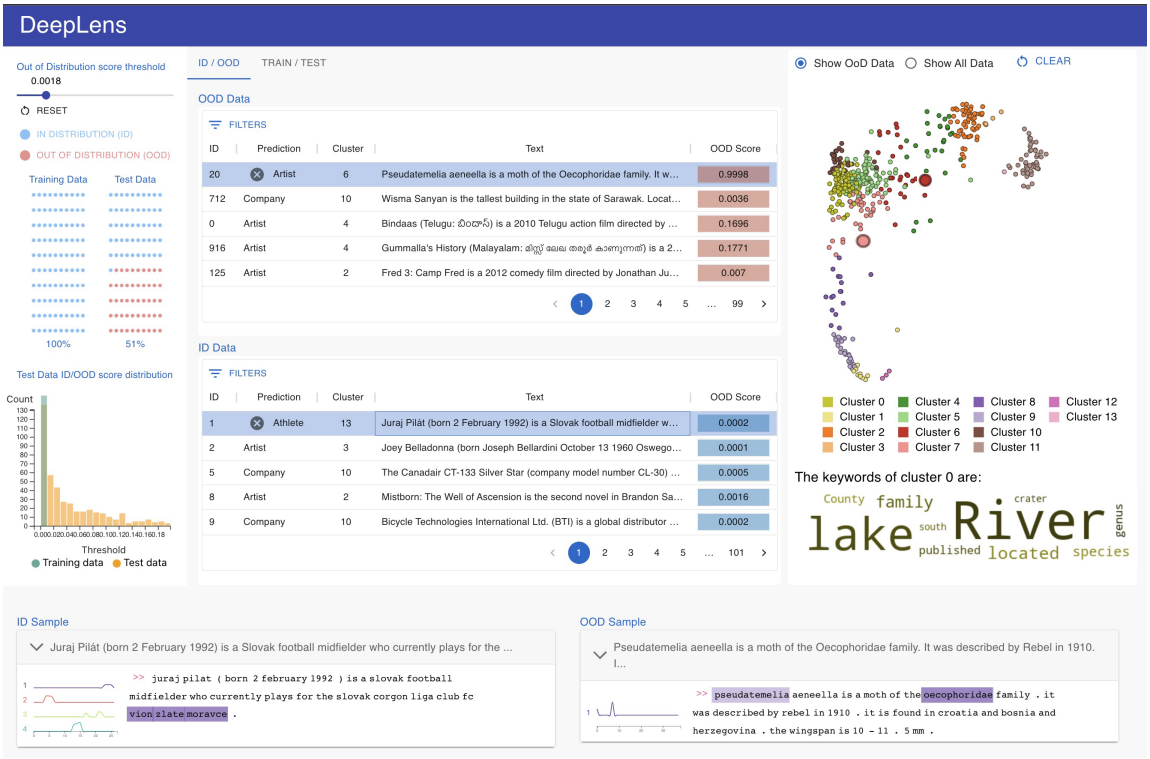

Figure A1: Interface of DeepLens for task 1.

Figure A2: Interface of DeepLens for task 2.

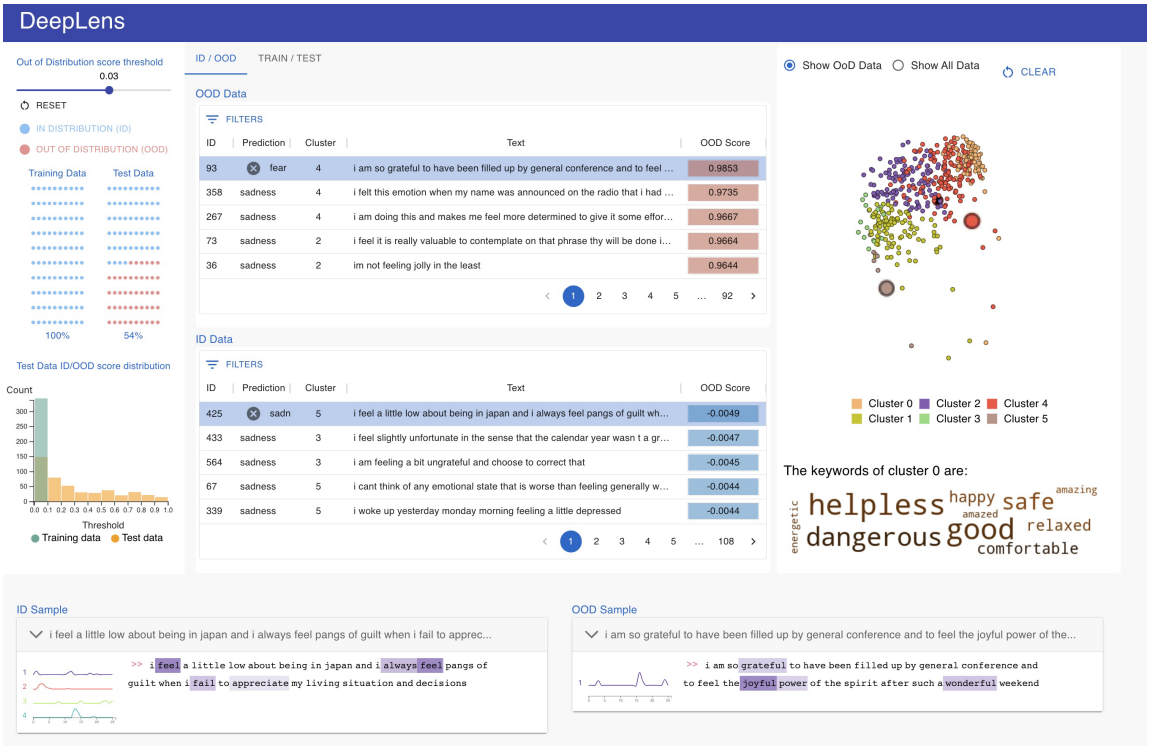

Figure A3: Interface of DeepLens for task 3.

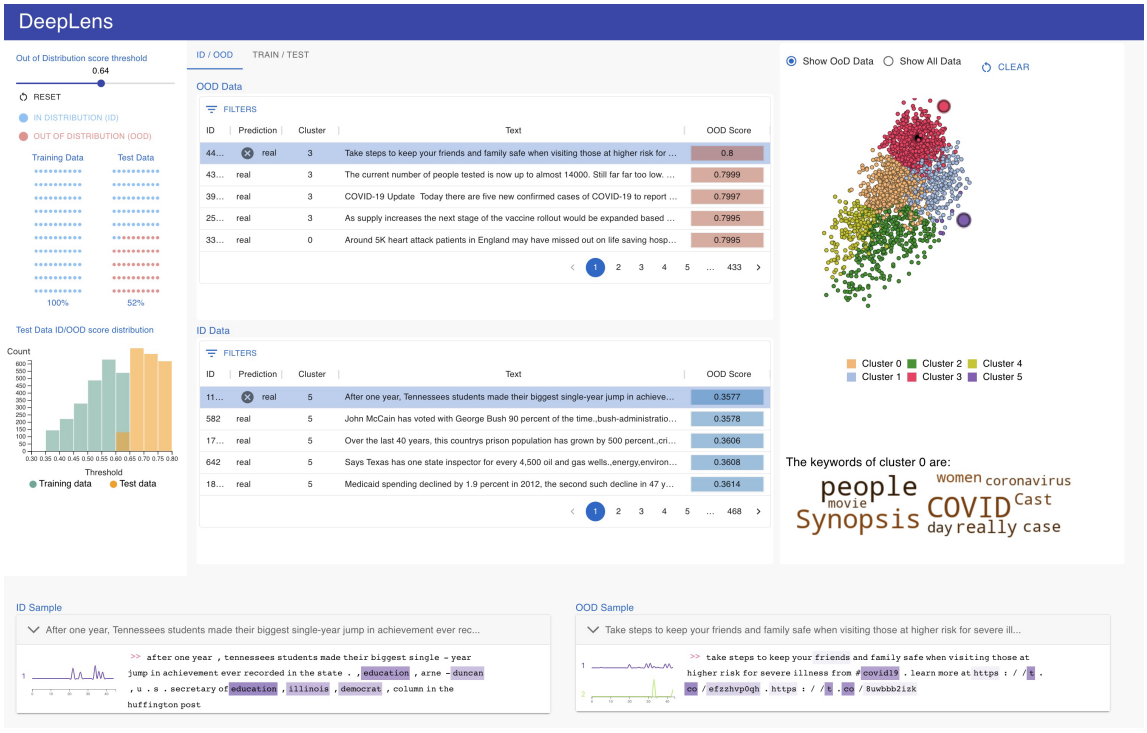

Figure A4: Interface of DeepLens for task 4.