Visual Supervision in Bootstrapped Information Extraction

Matthew Berger
Vanderbilt University, University of Arizona
matthew.berger@vanderbilt.edu

Ajay Nagesh
University of Arizona
ajaynagesh@email.arizona.edu

Joshua A. Levine
University of Arizona
josh@email.arizona.edu

Mihai Surdeanu
University of Arizona
msurdeanu@email.arizona.edu

Hao Helen Zhang
University of Arizona
hzhang@math.arizona.edu

Abstract

We challenge a common assumption in active learning, that a list-based interface populated by informative samples provides for efficient and effective data annotation. We show how a *2D scatterplot* populated with *diverse* and *representative* samples can yield improved models given the same time budget. We consider this for bootstrapping-based information extraction, in particular named entity classification, where human and machine jointly label data. To enable effective data annotation in a scatterplot, we have developed an embedding-based bootstrapping model that learns the distributional similarity of entities through the patterns that match them in a large data corpus, while being discriminative with respect to human-labeled and machine-promoted entities. We conducted a user study to assess the effectiveness of these different interfaces, and analyze bootstrapping performance in terms of human labeling accuracy, label quantity, and labeling consensus across multiple users. Our results suggest that supervision acquired from the scatterplot interface, despite being noisier, yields improvements in classification performance compared with the list interface, due to a larger quantity of supervision acquired.

1 Introduction

One strategy for mitigating the cost of supervised learning in information extraction (IE) is to bootstrap extractors with light supervision from a few provided examples (or seeds). Most typical bootstrapping methods (Yarowsky, 1995; Collins and Singer, 1999; Abney, 2007; Carlson et al., 2010; Gupta and Manning, 2014, 2015, inter alia) are iterative in nature, and suffer from semantic drift: as the learning advances, the task often drifts semantically into a related but different space, e.g., from learning women names into learning flower names (Komachi et al., 2008; Yangarber, 2003). In such cases, a human-in-the-loop to help guide bootstrapping through active learning (AL) (Settles, 2012) can be highly beneficial.

In this work, we challenge the common assumption made for AL methods in the context of IE: a visual interface that shows a *list* of samples ranked by their *informativeness* to the classifier is effective for building classifiers that minimize human annotator time (Dalvi et al., 2016; He and Grishman, 2015). We argue that this is an inefficient form of acquiring supervision from humans. Instead, we propose a *two-dimensional (2D) scatterplot* interface (rather than the one-dimensional (1D) list), where the examples to be annotated are selected by their capacity to *cluster together* (rather than by their informativeness to the classifier). We demonstrate that our approach leads to more data being annotated, and better overall performance for the model being learned.

In particular, we focus on the task of bootstrapped named entity classification (NEC), where a classifier is trained to label named entities with their corresponding category. For example, such an algorithm starts with a few examples of labeled names, e.g., "Barack Obama" as `PERSON`, from which it learns representative patterns, e.g., the pattern "`@ENTITY , former president`", which are then used to label other names in future iterations. Unlike traditional bootstrapping, our approach receives supervision in two ways: from the seed examples (as shown above), but also from human labels through an active learning step that is inserted after each bootstrapping iteration. To facilitate the clustering of examples in the annotation interface, we propose a semi-supervised NEC approach that learns custom embeddings for the entities being classified (§4). We select entities that are *diverse* and *representative* of the embedding's data distribution, and project them into a 2D visual encoding of the data via a scatterplot using

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 2043–2053
Brussels, Belgium, October 31 - November 4, 2018. ©2018 Association for Computational Linguistics

dimensionality reduction (§5).

The resulting scatterplot interface enables the user to label a larger quantity of entities, at the expense of label noise from mixed-category clusters and a potentially less-informative sampling criterion. To better understand this space, we conducted a user study to compare the effectiveness of the scatterplot interface compared to the traditional list that contains examples selected by informativeness (§7). Through our study, we arrive at the following takeaways:

- We observe that in configurations that used the scatterplot interface, annotators indeed labeled considerably more examples, but at an accuracy slightly lower than in the list interface. Despite the lower accuracy, the scatterplot interface generally yields better classifiers. In other words, the volume of annotations matters just as much as quality for the classifier performance.

- We find that a consensus model of users can mitigate noise, but must preserve a certain quantity of annotated data. A consensus model for the list interface that conservatively estimates labels *reduces* performance despite highly accurate labels due to the small amount of annotations. In contrast, the same model for the scatterplot interface yields higher label noise, but more annotations within the same budget of time and gives the best performance.

2 Related Work

Information extraction (IE) techniques commonly assume that the human supervision comes in the form of knowledge bases of facts disconnected from supporting text, as in the case of distant supervision (Mintz et al., 2009), or provides a light amount of supervision up front, as in the case of bootstrapping (Angeli et al., 2015). Common techniques for bootstrapping are to use rules for incrementally classifying entities (Collins and Singer, 1999) or to use syntactic (He and Grishman, 2015) and semantic (Gupta and Manning, 2014, 2015) contextual features. However, such approaches suffer from semantic drift, as previously discussed.

Considering a human-in-the-loop for IE has the potential to mitigate drift and greatly benefit performance, yet the challenge lies in minimizing human effort. The work of Angeli et al. (2014) show how to use active learning to improve distantly supervised relation extraction techniques (Surdeanu et al., 2012) through humans labeling informative relations. Werling et al. (2015) use Bayesian decision theory to minimize human cost and maximize accuracy for named entity recognition. For certain IE tasks, however, human supervision can be very noisy and thus counterproductive, especially from crowds, thus previous work has shown the importance of how to pose tasks for humans in providing labels (Liu et al., 2016), as well as automatically distinguishing simple labeling tasks from expert tasks in crowd-based task assignments (Wang et al., 2017). Our work shares a similar view of human supervision for IE, yet we instead study the impact of the annotation interface on the overall performance.

The role of the human-in-the-loop for topic modeling has also been extensively explored. For instance Smith et al. (2018) consider the types of modifications that one can provide to a built topic model (Hu et al., 2014) to make the topics more meaningful, while also studying the downstream human factor implications. Furthermore, prior work has also considered how different visual representations of topics impact a human's understanding of topic semantics (Smith et al., 2017). Closely related is the technique of Poursabzi-Sangdeh et al. (2016), where they highlight how different visual representations can have an impact on the effectiveness and efficiency of human labeling for document classification, comparing standard list interfaces with topic-grouped lists. Our method instead considers how 2D scatterplot interfaces, via embedding-based techniques, capture semantics for the purposes of providing labels in an IE task.

The visual analytics community has also investigated the role of visual interfaces and interaction tools for annotating data in supervised learning. For instance Heimerl et al. (2012) enable interactive labeling for document classification by visualizing unlabeled documents based on classifier uncertainty and document diversity. The technique of Höferlin et al. (2012) jointly visualizes unlabeled data and the classifier model, and allows the user to both label data points and directly modify model parameters. Closely related to our method is the work of Bernard et al. (2018) which compares active learning, via list interfaces, with in-

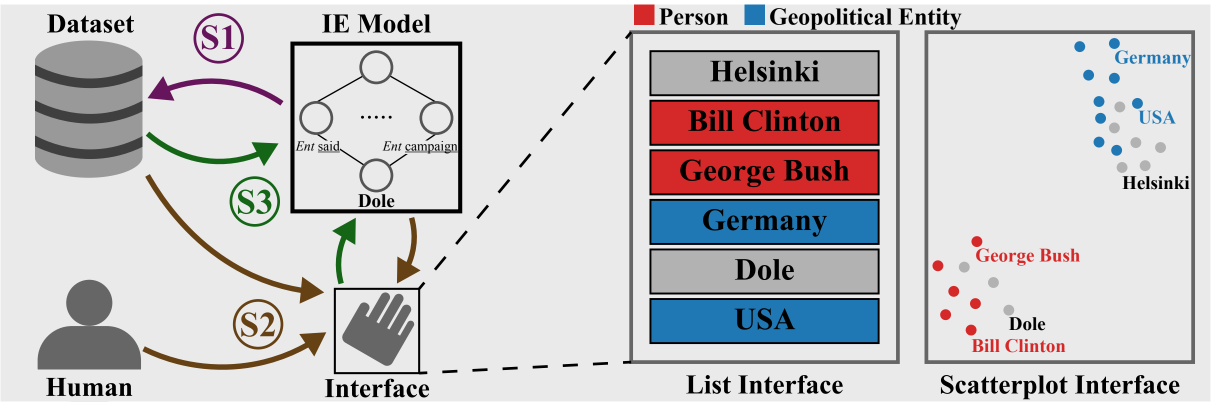


Figure 1: We show the workflow of our study (left): **(S1)** the Information Extraction model first automatically labels entities and updates its parameters, **(S2)** the human then labels entities through a given visual interface, and **(S3)** the model is updated based on the provided set of human labels. We study the effectiveness of two different interfaces (right): a list, and a 2D scatterplot of entities.

teractive visual labeling. However, their method is focused on acquiring labels solely from humans, whereas our method studies the interplay between self-labeling and human labels in a visualization context.

3 Human-Machine Workflow for Bootstrapping

We first describe the general workflow on which our study is based. We consider bootstrapping where both the human and the machine label entities *in tandem*, following the setup described in Fu and Grishman (2013) . More specifically, we consider the following iterative process (c.f. Fig. 1):

- (**S1**) The model automatically classifies entities, adds them as labeled data, and updates its parameters, as detailed in §4.
- (**S2**) The human interacts with a visual interface, driven by the model and the current set of labeled and unlabeled entities. The result of this step is a set of entities labeled by the human. We define this step a **round of labeling**, or just **round**.
- (**S3**) The model then updates its parameters given the human-labeled entities.

In contrast with typical bootstrapping, which interleaves entity promotion with model updates, here the human has the opportunity to label data. The intent is to *guide bootstrapping*: human annotations should steer bootstrapping towards learning the proper concepts, and away from semantic drift.

Yet, in this process we would like to *minimize human effort* while *maximizing bootstrapping performance*, where we define *effort* as the total time spent annotating data. In particular, the primary focus of our study is on step (**S2**), and how the user's interactions with the interface impacts these considerations. We consider two different types of interfaces for this purpose, shown on the right side of Fig. 1: list interfaces, and scatterplot interfaces. In particular, both interfaces utilize the state of the model to decide on what to show to the user. In the case of the list-based interface, we perform *uncertainty sampling* with respect to the model, as a way to maximize the information obtained from the user. For the scatterplot interface, the user interacts with a subset of *diverse* and *representative* entities through a 2D projection. Here we wish to see if more efficient *groupwise labeling* of perceived clusters in the scatterplot results in a more efficient and effective labeling process.

Of course, a critical piece to our study is the bootstrapping model itself. There exists a large body of work in bootstrapping, as previously discussed, and one possibility is to use an existing technique for our work. For our learning scenario, a bootstrapping technique should satisfy several criteria:

- Be efficient to update, in order to minimize user latency with the interface;
- Incorporate user supervision to ensure a discriminative representation;
- Be suitable for visual exploration.

Please note that traditional bootstrapping techniques such as rule-based methods (Collins and Singer, 1999) fail to meet all criteria. In particular, they are not suitable for visual exploration because there is no clear way to represent the semantic proximity of rules or of the concepts being learned. We next describe how to address these challenges through embedding-based bootstrapping.

4 Embedding-Based Bootstrapping

Our bootstrapping technique is based on neural language models (Mikolov et al., 2013), in particular, semi-supervised embedding-based bootstrapping techniques (Valenzuela-Escárcega et al., 2018). Unlike other word embedding algorithms, our approach measures distributional similarity of *entities* (rather than words) with respect to *patterns* (rather than context words). We define a pattern as a small sequence of words that surround an entity, up to ±4 words to the left/right of the entity under consideration. For instance, in the phrases "**John** said on Saturday" and "**John** told reporters" the patterns "said on Saturday" and "told reporters" are suggestive of the category Person for entity **John**.[1] We observe that entities of a given category, e.g. Person, are likely to have common pattern distributions. This observation drives our method for learning entity and pattern embeddings. Furthermore, as we will see, this method permits efficient updates, and it is suitable for visualization due to the geometric representation of entities and patterns.

More specifically, assume that we have extracted a set of entities E and patterns P in a given text corpus C. We associate each entity $e \in E$ and pattern $p \in P$ with *embeddings* $\mathbf{x}_e$ and $\mathbf{x}_p$, respectively, with $\mathbf{x}_e, \mathbf{x}_p \in \mathbb{R}^d$. To satisfy the above form of distributional similarity, we utilize the Skip Gram model (Mikolov et al., 2013) and seek entity embeddings to be close to their embeddings of observed patterns through maximizing:

$$SG = \sum_{(e,p)\in C_p} [\log(\sigma(\mathbf{x}_e^\top \mathbf{x}_p)) + \sum_{n\in N} \log(\sigma(-\mathbf{x}_e^\top \mathbf{x}_n))], \tag{1}$$

where (e, p) corresponds to an entity-pattern occurrence from the corpus set C_p, n represents a *negative* pattern, sampled from the unigram distribution of all patterns N (Levy and Goldberg, 2014) [2], and σ is the sigmoid function. Intuitively, this forces an entity's embedding to be similar to embeddings of its matched patterns from the corpus, and dissimilar to random pattern embeddings.

A disadvantage with the Skip Gram model is that it might fail to be discriminative, since it does not utilize category labels. Thus, we introduce an objective term that seeks to bring entities that belong to the same category to have similar embeddings, and entities that belong to different categories to be far apart in their embeddings. We realize this using large-margin metric learning (Cui et al., 2016; Sohn, 2016), minimizing:

$$LM = \sum_{(a,b,c)\in E_l} \lfloor s(\mathbf{x}_a, \mathbf{x}_c) - s(\mathbf{x}_a, \mathbf{x}_b) + M \rfloor_+, \tag{2}$$

where (a, b, c) represents a triplet of entities, such that a and b belong to the same category, while c belongs to a different category, and the function s is the cosine similarity between entity embeddings. The set E_l is a subset of entities from E that have been assigned categories so far, as provided by the bootstrapper (**S1**) or the human (**S2**). Intuitively, entities from dissimilar categories should be positioned in the embedding space such that their cosine similarity is at least a margin M from any pair of entities of the same category.

We combine the Skip Gram objective with the metric learning objective to obtain:

$$B = LM - SG. \tag{3}$$

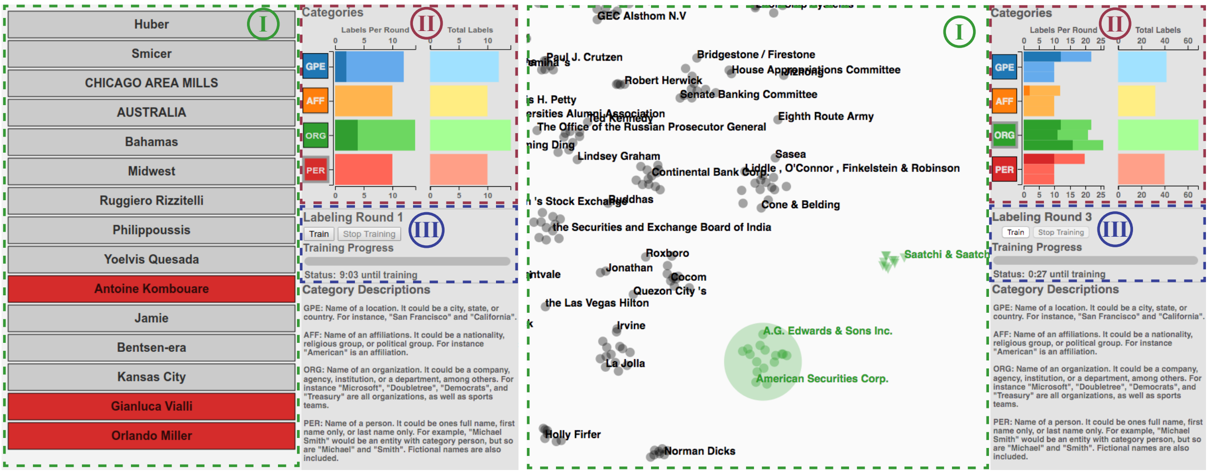
The objective B can be viewed as a form of semi-supervised representation learning, where from a sparse set of labeled entities, we wish to learn entity representations that are similar should they have patterns in common (SG), while simultaneously ensuring that embeddings are discriminative with respect to categories (LM). Step (**S3**) minimizes this objective at every round of bootstrapping via stochastic gradient descent, given the current set of labeled entities E_l.

4.1 Promoting Entities

We use the learned embedding to automatically promote unlabeled entities to categories, as discussed in step (**S1**). We use the normalized entity embeddings as features, and build a multinomial

[1] In this work we use surface patterns, but the proposed algorithm is agnostic to the types of patterns used.

[2] In initial ablation studies, we found that this strategy was about as effective as negative sampling from *all* patterns, but significantly faster, which is necessary for interactivity.



GPE: Name of a location. It could be a city, state, or country. For instance, "San Francisco" and "California".

AFF: Name of an affiliations. It could be a nationality, religious group, or political group. For instance "American" is an affiliation.

ORG: Name of an organization. It could be a company, agency, institution, or a department, among others. For instance "Microsoft", "Doubletree", "Democrats", and "Treasury" are all organizations, as well as sports teams.

PER: Name of a person. It could be ones full name, first name only, or last name only. For example, "Michael Smith" would be an entity with category person, but so are "Michael" and "Smith". Fictional names are also included.

GPE: Name of a location. It could be a city, state, or country. For instance, "San Francisco" and "California".

AFF: Name of an affiliations. It could be a nationality, religious group, or political group. For instance "American" is an affiliation.

ORG: Name of an organization. It could be a company, agency, institution, or a department, among others. For instance "Microsoft", "Doubletree", "Democrats", and "Treasury" are all organizations, as well as sports teams.

PER: Name of a person. It could be ones full name, first name only, or last name only. For example, "Michael Smith" would be an entity with category person, but so are "Michael" and "Smith". Fictional names are also included.

(a) List Interface

(b) Scatterplot Interface

Figure 2: We study two different interfaces for humans to label entities, a list interface (a) and a 2D scatterplot interface (b). In both, the user selects entities in the main view (I), based on an assigned category (II), wherein we also show the total amount of labels the user, and the machine, has thus far labeled. In (III) we show the current status of training/labeling, and the option to initiate/stop training.

logistic regression model over the given set of labeled entities, trained to predict entity categories. For each category, we then promote the most confident entities to the category. To compute confidence, we treat the model's predictions as category probabilities, take the entropy over this distribution as a measure of unconfidence, and promote entities with the lowest entropy.

5 Supervision Interfaces

We now turn to the visual interface through which humans provide supervision (step **S2**). We distinguish the interfaces by how entities are *sampled*, *presented* to the user, and each interface's set of *interactions* for labeling.

5.1 List Interface

Sampling. We sample entities through sorting them by confidence, as defined in §4.1, and sample the most unconfident entities. This form of uncertainty sampling is common in list-based interfaces (Angeli et al., 2014; Poursabzi-Sangdeh et al., 2016), as a measure of informativeness for updating the model (Settles, 2012).

Presentation. We next show the 15 most uncertain entities in a 1D list-based visual interface (Fig. 2a (**I**)).

Interactions. The user labels entities by first selecting their desired category (Fig. 2a (**II**)), followed by clicking on the entity in the main display. We also allow the user to select multiple entities at once, for a given category. As the user labels entities, we repopulate the display with the next set of most unconfident entities.

5.2 Scatterplot Interface

Sampling. For the scatterplot interface, we aim to sample entities that are beneficial for the model, while also ensuring the user can efficiently label entities through groupwise selection. Uncertainty sampling, though potentially informative, can lead to projections that are challenging for the user in performing groupwise labeling, as we experimentally verified that entities with high classification uncertainty are unlikely to group together. To address this, we sample entities based on how they are distributed in the embedding space, to provide us a *diverse* and *representative* sampling (Xu et al., 2007). More specifically, we first perform k-means on the entities' normalized embeddings, with $k = 40$, and sample across clusters to give diversity. To ensure a representative sampling, kernel density estimation is performed within each cluster's entities, and the entity with highest density, along with its nearest neighbors in the embedding space, are selected. The number of neighbors sampled in each cluster is proportional to the cluster size, to ensure balance across clusters.

Presentation. We next perform a 2D projection of 500 sampled entities using t-SNE (Maaten and Hinton, 2008), visually encoding each entity by a filled circle (Fig. 2b (**I**)). To provide context with respect to entities labeled by the human or machine, we jointly project the unlabeled sampled entities, and a subset of the entities previously labeled (drawn as triangles), either by the human annotator or promoted with high-confidence by the machine.

Interactions. The user labels entities by first selecting a category (Fig. 2b **(II)**), followed by using a circular brush to label groups of entities. The user can adjust the brush's radius, as well as change the view through panning and zooming. We dynamically filter the text labels for entities based on the zoom level to reduce clutter, thus the user can observe a high-level overview of the space of entities when zoomed out, and observe more details upon zooming in. This gives the annotator the chance to perform groupwise annotations by jointly considering cluster structure and a sparse set of text labels from the cluster, and labeling all entities at once should the annotator decide the group of entities belongs to a single category.

5.3 Training

Common to both interfaces, user interactions are interleaved with model updates. During training, the list interface is refreshed with the top 15 most informative samples every 3 *training epochs*, or passes over the data, while for the scatterplot the samples and their 2D positions are updated. To ensure temporal coherence for those entities that persist between updates, we employ dynamic t-SNE (Rauber et al., 2016). The user can opt to stop training if they observe little change occurring between snapshots (Fig. 2 **(III)**).

6 User Study

We conducted a user study to investigate the effectiveness of the different interfaces. We recruited 10 participants for our study: the median age was 22, the minimum and maximum respectively 19 and 41, 5 participants self-reported as being “somewhat knowledgeable” of machine learning, 4 with ”no knowledge”, and 1 identifying as an “expert”. We used a within-subject design, where the first interface presented to the participant was selected at random to mitigate potential priming effects. For each interface, a set of instructions was first presented, followed by a brief tutorial where the user must label a small set of seed entities – 10 per category. These seed entities, labeled with ground truth rather than user labels, initialize the bootstrapping model, so that participants start off with identical embeddings. After the instructions, the participant then performs 10 rounds of labeling, where they may label entities for up to 1 minute per round. After each round of labeling, the bootstrapper promotes 10 entities to each category, and then performs 30 epochs of training.

Dataset. We use the Ontonotes dataset (Weischedel et al., 2013), limited to the 4 categories that have the most frequently mentioned entities, in order to make the labeling task manageable, yet still nontrivial, for participants. These categories are people (PER), organizations (ORG), geopolitical entities (GPE), and nationalities as well as religious/political affiliations (AFF), resulting in 6,567, 6,199, 1,617, and 422 entities per category, respectively.

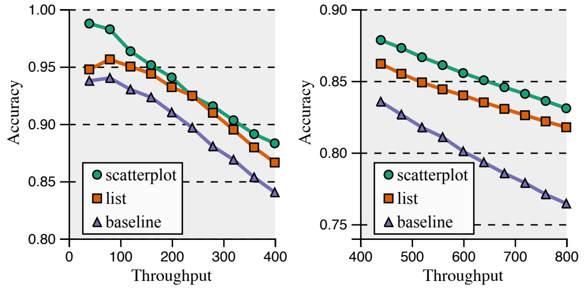
Bootstrapping Details. It is critical to ensure that bootstrapping training minimizes user latency, while not sacrificing performance (c.f. Eq. 3). To strike this balance, we set the number of negative patterns sampled for each entity to 10 (Levy and Goldberg, 2014), the embedding dimension d to 100, perform hard negative triplet mining (Cui et al., 2016) to form the loss on triplets likely to violate the margin, and set the margin M to 0.4. Experimentally, we found these settings allowed training to converge to a good solution after 30 training epochs, and that each epoch took no longer than 1 second on average.

7 Results

We analyze the results in terms of three forms of evaluation: *bootstrapping performance*, as determined by the entities promoted during the course of the user study, *extrapolation*, wherein we let bootstrapping proceed to promote entities after the joint human-machine labeling has completed, and *consensus*, where we combine the set of entities annotated by participants within the different interfaces. We also analyze a typical user's labeling, and corresponding machine performance.

7.1 Bootstrapping Performance

We first look at the effectiveness of promoted entities during the course of each participant's interactions. Fig. 3(a) shows bootstrapping accuracy averaged over all users for the list and scatterplot. We also compare to a baseline of traditional bootstrapping, where no human labels are considered, averaged over 10 trials. Note this is simi-



(a) Bootstrapping (b) Extrapolation

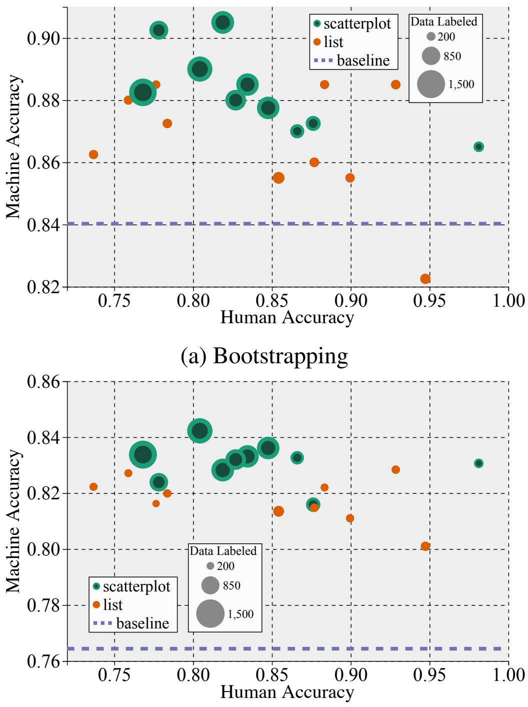
Figure 3: Performance averaged over all users for bootstrapping during user study (a) and extrapolation (b), for both interfaces and a baseline that does not use human labels. The green, orange, and purple plots are scatterplot, list, and baseline, respectively.

	scatterplot	list	baseline
bootstrapping	88.8	87.2	85.1
extrapolation	84.7	83.5	78.6

Table 1: We show performance in terms of recall, computed per-category and averaged across categories, for the baseline, list and scatterplot interfaces. This is shown for the last round of bootstrapping (c.f. Fig. 3a) and extrapolation (c.f. Fig. 3b).

lar to Valenzuela-Escárcega et al. (2018), which demonstrates improved performance across a set of bootstrapping techniques, and thus represents a competitive baseline. We find that the performance of both interfaces outperform this baseline, and that the scatterplot outperforms the list: a two-sample Welch's t-test concludes statistical significance (p=0.05), as well as for a paired t-test measured within participants (p=0.03). In Table 1 we also show recall, computed per-category and averaged over categories, at the last round of bootstrapping and similarly find the scatterplot outperforms the list.

Better insight between the interfaces can be gained by looking at individual user performance. Fig. 4(a) shows a plot of each participant's performance (y-axis) for both interfaces, as a function of their labeling accuracy (x-axis) and the total number of labels provided (size of each circle). Note that the 3 best-performing models come from the scatterplot interface, *even at the expense of a lower labeling accuracy*. This suggests that the *number of labels* can counter the *noise in labeling*, compared to a list where we may have potentially more informative entities labeled more accurately, but much fewer entities annotated.



(a) Bootstrapping

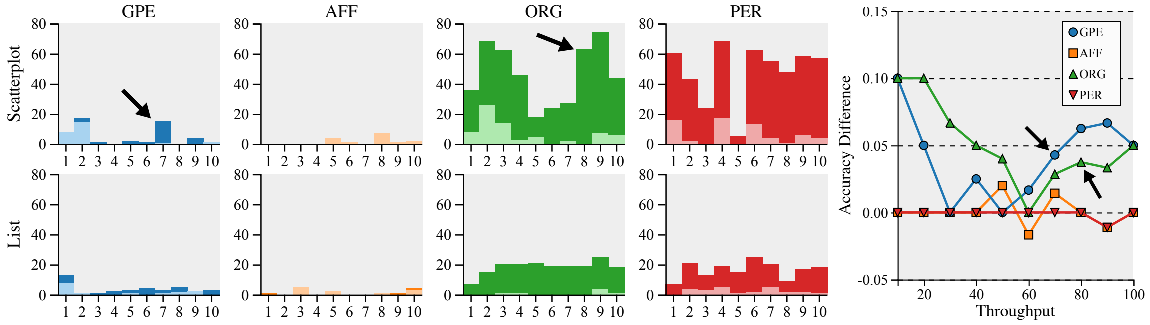
(b) Extrapolation

Figure 4: Bootstrapping performance for user study (a), and extrapolation (b) for individual participants, evaluated at throughputs (number of entities promoted) of 400 and 800, respectively. Each circle is a participant whose color indicates the interface (green, orange, and purple are scatterplot, list, and baseline, respectively); x-axis is participant labeling accuracy; y-axis is machine accuracy; and circle size encodes the number of annotated labels (large circle indicates large number of annotated labels).

7.2 Extrapolation

We next look at how bootstrapping continues to learn, when starting from all of the annotations provided by the corresponding human annotator as well as the entities promoted by the machine. We term this configuration extrapolation. This analysis indirectly measures how much noise crept into the available annotations, by measuring the performance of the classifier trained on this data.

We repeated this experiment over 10 different trials, promoting 10 entities per category in each round, and take the average accuracy, see Fig. 3(b) for the performance averaged over all users for



(a) Human Labeling Accuracy

(b) Interface Accuracy

Figure 5: Labeling accuracy and bootstrapping performance for a typical participant. In (a) we show per-category annotations, where the x-axis is the round number and y-axis is the number of entities labeled. Colors of higher lightness indicate mislabeled entities. For example, approximately 40% of the user's annotations in round 2 for ORG in the scatterplot were incorrect. In (b) we show the difference in bootstrapping performance between the scatterplot and list interfaces. For instance, scatterplot yields a model 5% more accurate than list for GPE in round 10. Note the correlation between human labeling (left) and bootstrapping performance (right), highlighted by the arrows.

the interfaces, as well as the aforementioned baseline. Observe that the baseline performs progressively worse as a function of throughput, indicating that labels provided by the human annotators help to prevent drift. We also find the results to be stronger compared with §7.1; namely, a two-sample Welch's t-test concludes statistical significance (p=0.002) and similarly for a paired t-test within participants (p=0.007). The recall performance in Table 1 also confirms the performance gains of the scatterplot. Supporting these results, Fig. 4(b) shows that the scatterplot interface for users that labeled a large amount of entities generally outperform those where fewer entities were labeled. This experiment suggests that the best strategy to control for semantic drift is to aim for an interface that yields *many* annotations at *reasonable* accuracy, rather than few, higher-quality ones.

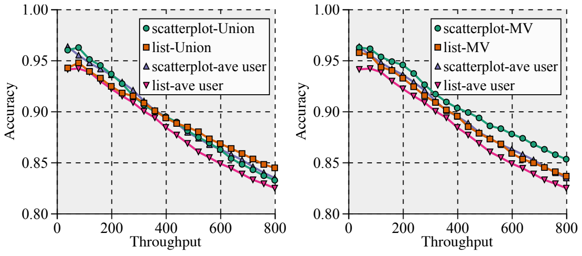
7.3 Individual User Labeling

We show labeling accuracy and bootstrapping accuracy for a participant that has similar performance to the average (c.f. Fig. 3(a)). In Fig. 5(a) we show the number of entities labeled across categories, where bars of higher color lightness encode the number of incorrect labels, and in (b) we plot the difference in bootstrapping performance between the scatterplot and the list. We observe that the class imbalance of the Ontonotes dataset tends to manifest in labeling as well, where ORG and PER are typically labeled the most. We also see a correlation between accurate human labeling and bootstrapping performance (e.g. the arrows pointing at GPE and ORG for rounds 7 and 8, respectively). This highlights an advantage of exploratory visual interfaces for labeling, such as a scatterplot, where the user can search for clusters of a particular category, e.g., in the case of GPE there may not exist many entities of this type in the list interface.

7.4 Consensus

Given the noise inherent in a single user's annotations, we last analyze whether techniques to *combine* labels across a set of users can help reduce label noise and improve performance. To this end, for the scatterplot and list interface we combine all annotations, and consider two different types of consensus methods: 1) **Union:** we take the union of all annotated entities, choosing an entity's label at random during conflicts; and 2) **Majority Vote (MV):** for conflicting entities we select the most voted category across users, discarding entities that have only been annotated once or have ties across users. We then seed bootstrapping with the resulting set of labels, run 20 rounds of promotion, and take the average performance of 10 trials.

Fig. 6(a-b) compares each method across the different interfaces, while Table 2 shows accuracy and total number of labels for the consensus methods. We also include performance, labeling accuracy, and label totals averaged across individual users, where for each user we seed bootstrapping



(a) Union

(b) Majority Vote

Figure 6: Accuracy of different consensus methods, compared across the two interfaces, as well as average performance across individual users.

with their full set of annotations. For **Union**, we observe that despite the large number of labels, the scatterplot performance is comparable to average individual performance, suggesting that in this case, the volume of labels does not counter the noise in the data. For the list, we can see a gain in performance, which is likely attributed to the cleaner annotations from participants. For **MV**, however, we find that the scatterplot performs the best, whereas the list performs worse compared with its **Union** counterpart, now having approximately 3x less labels. This emphasizes the interplay between *accuracy* and *label quantity*, even despite the more informative labels provided through the list interface, and supports our previous observation that balancing the volume of annotations with their quality yields the best annotation strategy.

8 Discussion

We acknowledge some limitations in our technique and user study. A drawback to our sampling scheme for the scatterplot is that it does not utilize classifier informativeness. In initial experiments, we found that uncertainty sampling led to 2D projections that were challenging to groupwise label. This result is intuitive: data points classified with lowest confidence will naturally have low-quality embeddings, yielding poor clusters. Thus we can not perform a consistent comparison of the same sampling criterion used between the interfaces. For future work we will investigate active learning techniques that are relevant to the learning task, while still permitting efficient labeling.

In this work we restricted our study to entity classification, one of many types of IE tasks. Our approach should generalize to other IE tasks, however, provided these tasks exhibit certain structure. Specifically, user annotations in a task should amount to labeling of data instances, and the data instances defined by the task should be able to be perceived in a 2D scatterplot. For instance, coreference resolution and relationship extraction fit both criteria. Assessing the effectiveness of visual supervision for these tasks is outside of the scope of this paper, however, and we will consider these studies in future work.

Interface	Union		MV		Ave. User	
	Acc	Total	Acc	Total	Acc	Total
List	82.7	1463	95.4	543	85.1	217
Scatterplot	77.6	3503	92.5	1959	82.3	788

Table 2: Consensus accuracies and label amounts across the interfaces. The last column is average user accuracy and label amount.

Overall, our user study clearly highlights the importance of visual interfaces in acquiring supervision for semi-supervised information extraction. We demonstrated that, when compared to the traditional list interface, the scatterplot allows a larger volume of annotations to be created at reasonable accuracy, yielding better classifiers. We believe this finding will influence active learning, in terms of sampling criteria and the interplay between AL and visual interfaces.

9 Acknowledgements

This work was supported by funding from an NSF TRIPODS grant (NSF-1749858), and funding from the Bill and Melinda Gates Foundation HBGDki Initiative. Mihai Surdeanu disclose a financial interest in lum.ai. This interest has been disclosed to the University of Arizona Institutional Review Committee and is being managed in accordance with its conflict of interest policies.

References

Steven Abney. 2007. *Semisupervised Learning for Computational Linguistics*, 1st edition. Chapman & Hall/CRC.

Gabor Angeli, Julie Tibshirani, Jean Wu, and Christopher D Manning. 2014. Combining distant and partial supervision for relation extraction. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1556–1567.

Gabor Angeli, Victor Zhong, Danqi Chen, Arun Chaganty, Jason Bolton, Melvin Johnson Premkumar, Panupong Pasupat, Sonal Gupta, and Christopher D Manning. 2015. Bootstrapped self training for knowledge base population. In *Proceedings of the Eighth Text Analysis Conference (TAC2015)*.

Jürgen Bernard, Marco Hutter, Matthias Zeppelzauer, Dieter Fellner, and Michael Sedlmair. 2018. Comparing visual-interactive labeling with active learning: An experimental study. *IEEE transactions on visualization and computer graphics*, 24(1):298–308.

Andrew Carlson, Justin Betteridge, Richard C Wang, Estevam R Hruschka Jr, and Tom M Mitchell. 2010. Coupled semi-supervised learning for information extraction. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 101–110. ACM.

Michael Collins and Yoram Singer. 1999. Unsupervised models for named entity classification. In *1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*.

Yin Cui, Feng Zhou, Yuanqing Lin, and Serge Belongie. 2016. Fine-grained categorization and dataset bootstrapping using deep metric learning with humans in the loop. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1153–1162.

Bhavana Dalvi, Sumithra Bhakthavatsalam, Chris Clark, Peter Clark, Oren Etzioni, Anthony Fader, and Dirk Groeneveld. 2016. IKE - an interactive tool for knowledge extraction. In *Proceedings of the 5th Workshop on Automated Knowledge Base Construction, AKBC@NAACL-HLT 2016, San Diego, CA, USA, June 17, 2016*, pages 12–17.

Lisheng Fu and Ralph Grishman. 2013. An efficient active learning framework for new relation types. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 692–698.

Sonal Gupta and Christopher D Manning. 2014. Improved pattern learning for bootstrapped entity extraction. In *CoNLL*, pages 98–108.

Sonal Gupta and Christopher D. Manning. 2015. Distributed representations of words to guide bootstrapped entity classifiers. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*.

Yifan He and Ralph Grishman. 2015. Ice: Rapid information extraction customization for nlp novices. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 31–35, Denver, Colorado. Association for Computational Linguistics.

Florian Heimerl, Steffen Koch, Harald Bosch, and Thomas Ertl. 2012. Visual classifier training for text document retrieval. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2839–2848.

Benjamin Höferlin, Rudolf Netzel, Markus Höferlin, Daniel Weiskopf, and Gunther Heidemann. 2012. Inter-active learning of ad-hoc classifiers for video visual analytics. In *Visual Analytics Science and Technology (VAST), 2012 IEEE Conference on*, pages 23–32. IEEE.

Yuening Hu, Jordan Boyd-Graber, Brianna Satinoff, and Alison Smith. 2014. Interactive topic modeling. *Machine learning*, 95(3):423–469.

Mamoru Komachi, Taku Kudo, Masashi Shimbo, and Yuji Matsumoto. 2008. Graph-based analysis of semantic drift in espresso-like bootstrapping algorithms. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 1011–1020, Stroudsburg, PA, USA. Association for Computational Linguistics.

Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. In *Advances in neural information processing systems*, pages 2177–2185.

Angli Liu, Stephen Soderland, Jonathan Bragg, Christopher H Lin, Xiao Ling, and Daniel S Weld. 2016. Effective crowd annotation for relation extraction. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 897–906.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 1003–1011. Association for Computational Linguistics.

Forough Poursabzi-Sangdeh, Jordan Boyd-Graber, Leah Findlater, and Kevin Seppi. 2016. Alto: Active learning with topic overviews for speeding label induction and document labeling. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1158–1169.

Paulo E Rauber, Alexandre X Falcão, and Alexandru C Telea. 2016. Visualizing time-dependent data using dynamic t-sne. *Proc. EuroVis Short Papers*, 2(5).

Burr Settles. 2012. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1):1–114.

Alison Smith, Varun Kumar, Jordan Boyd-Graber, Kevin Seppi, and Leah Findlater. 2018. Closing the loop: User-centered design and evaluation of a human-in-the-loop topic modeling system. In *23rd International Conference on Intelligent User Interfaces*, pages 293–304. ACM.

Alison Smith, Tak Yeon Lee, Forough Poursabzi-Sangdeh, Jordan Boyd-Graber, Niklas Elmqvist, and Leah Findlater. 2017. Evaluating visual representations for topic understanding and their effects on manually generated labels. *Transactions of the Association of Computational Linguistics*, 5(1):1–15.

Kihyuk Sohn. 2016. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in Neural Information Processing Systems*, pages 1857–1865.

Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D Manning. 2012. Multi-instance multi-label learning for relation extraction. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, pages 455–465. Association for Computational Linguistics.

Marco A Valenzuela-Escárcega, Ajay Nagesh, and Mihai Surdeanu. 2018. Lightly-supervised representation learning with global interpretability. *arXiv preprint arXiv:1805.11545*.

Chenguang Wang, Alan Akbik, Yunyao Li, Fei Xia, Anbang Xu, et al. 2017. Crowd-in-the-loop: A hybrid approach for annotating semantic roles. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1913–1922.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, et al. 2013. Ontonotes release 5.0 ldc2013t19. *Linguistic Data Consortium, Philadelphia, PA*.

Keenon Werling, Arun Tejasvi Chaganty, Percy S Liang, and Christopher D Manning. 2015. On-the-job learning with bayesian decision theory. In *Advances in Neural Information Processing Systems*, pages 3465–3473.

Zuobing Xu, Ram Akella, and Yi Zhang. 2007. Incorporating diversity and density in active learning for relevance feedback. In *European Conference on Information Retrieval*, pages 246–257. Springer.

Roman Yangarber. 2003. Counter-training in discovery of semantic patterns. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

David Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.