Modeling and Aggregation of Complex Annotations via Annotation Distances

Alexander Braylan
Department of Computer Science
University of Texas at Austin
braylan@cs.utexas.edu

Matthew Lease
School of Information
University of Texas at Austin
ml@utexas.edu

ABSTRACT

Modeling annotators and their labels is valuable for ensuring collected data quality. Though many models have been proposed for binary or categorical labels, prior methods do not generalize to *complex* annotations (e.g., open-ended text, multivariate, or structured responses) without devising new models for each specific task. To obviate the need for task-specific modeling, we propose to model distances between labels, rather than the labels themselves. Our models are largely agnostic to the distance function; we leave it to the *requesters* to specify an appropriate distance function for their given annotation task. We propose three models of annotation quality, including a Bayesian hierarchical extension of *multidimensional scaling* which can be trained in an unsupervised or semi-supervised manner. Results show the generality and effectiveness of our models across diverse complex annotation tasks: sequence labeling, translation, syntactic parsing, and ranking.

1 INTRODUCTION

Annotations (aka labels) provide the basis for supervised learning and evaluation. Given the importance of annotation, many models and measures of annotator behavior and labels have been proposed [1, 8, 22, 42, 53]. The advent of inexpert crowd annotation [54, 55] has stimulated a surge of further modeling for quality assurance with inexpert annotators [63]. However, nearly all existing annotation models, traditional or crowd, assume relatively simple labeling tasks, such as classification or rating.

Not all annotation tasks are so simple. Some tasks involve open-ended answer spaces (e.g., translation, transcription, extraction) or structured responses (e.g., annotating ranked lists, linguistic syntax or co-reference). Lacking general annotation and aggregation models for such tasks, quality assurance is often pursued in other ways: 1) defining custom probabilistic models for each task of interest (e.g., sequence annotation [48] or co-reference [43]); 2) falling back on a second group of annotators to select, verify, or fix responses from the first group (i.e., designing a bespoke annotation workflow) [3]; or 3) measuring worker reliability indirectly, e.g., via attention checks [32] or behavior traces [14, 49]. Limitations of existing approaches helped prompt a 2019 EMNLP workshop calling for modeling of complex annotations [44].

We propose an unified approach for modeling complex annotations that generalizes to a wide variety of annotation tasks. To obviate the need for new probabilistic models for each task, we model distances between annotations, rather than the annotations themselves. Our methods are largely agnostic as to distance function, leaving it to the *requester* to specify an appropriate distance function for their annotation task (via a callback function). In general, any error metric to compare human annotations or model predictions vs. gold labels can serve as the distance function. We can then estimate annotator reliability by the distance from their labels to those of peer annotators or a trusted gold standard. Though we do not evaluate on binary and categorical labeling tasks, our approach to modeling label distances generalizes across both simple and complex labeling tasks.

We describe a method for modeling label distances, *Multidimensional Annotation Scaling* (MAS), a Bayesian hierarchical extension of multidimensional scaling. Beyond inferring the best label for each item, the model also estimates annotator reliability jointly with item difficulty [59]. In addition, We describe configurations of MAS to produce two simpler variants: *Smallest Average Distance* (SAD), a generalization of majority vote to complex annotations, and *Best Available User* (BAU), which selects the annotator with the highest accuracy on average. Requesters can decide how to balance simplicity vs. effectiveness in selecting which method best suits their needs [63]. Our evaluation shows the generality and effectiveness of our distance-based modeling methods across four diverse complex annotation tasks: multiple sequence labeling [37, 48], translation [62], syntactic parsing [31], and item ranking.

Contributions. Complex annotation tasks are abundant and important, yet we are not familiar with any task-independent, general-purpose aggregation models for such tasks. Instead, quality assurance is typically performed for each distinct task by defining custom aggregation models or designing bespoke annotation workflows. By modeling label distances rather than labels, we enable a single, general annotation and aggregation model to support diverse tasks involving complex annotations. This model allows both unsupervised and semi-supervised learning. We also propose two simpler aggregation methods and compare them against baselines in annotation aggregation experiments across four diverse datasets. In addition to conceptual framing and our three proposed methods, we also share our datasets and source code[1]. We thus expect to impact practice, to encourage collection of more such data by easing quality assurance, and to foster related research.

WWW '20, April 20–24, 2020, Taipei, Taiwan

ACM ISBN 978-1-4503-7023-3/20/04.
https://doi.org/10.1145/3366423.3380250

[1] Data and source code: https://github.com/Praznat/annotationmodeling

2 BACKGROUND

The benefits of annotation modeling [42] are well-known for both traditional and crowd annotation settings. In this work, we assume *objective* tasks in which annotation quality is measurable and items to annotate may vary in difficulty. More complex annotation tasks may reveal greater variability in annotator abilities, as well as a wide range of different but satisfactory annotations for the same item. As such, complex annotation tasks may benefit more from modeling than simpler annotation tasks.

As researchers seek to automate ever-more sophisticated tasks, annotation needed to train and evaluate AI models becomes increasingly complex. Examples include structured linguistic syntax [31], ranked lists, sequences [37, 48], open-ended answers to math problems [26], or even drawings [11, 16].

Lacking general annotation and aggregation methods for modeling such complex annotations, quality assurance for complex annotation remains a bit eclectic today. For example, one can include insert non-task, attention-checks (ACs) (e.g., "What is the third word on this page?"). However, such ACs are easily distinguished from actual task-questions, making it easy for an annotator to pass an AC while still performing poorly on the actual task of interest [32]. Similarly, free text responses below a certain word count or with bad grammar may be judged as having bad quality [25, 62]. However, how to assess annotation quality by its content requires specifying task-specific checks, which are often relatively ineffective and can significantly reduce the worker pool size [6].

One of the most popular approaches, *honeypot* questions, evaluates worker responses against pre-defined or *known-gold* answers. This approach can be used to estimate the reliability of workers by how well they score on the honeypot questions. For simple annotation tasks, a small label space permits evaluating responses based on exact match vs. gold labels. As we move toward ordinal rating tasks, we might instead assess partial-credit, penalizing "near misses" less than other errors. With complex annotation, the space of possible labels may be vast and such partial-credit evaluation becomes essential (e.g., there may be several acceptable ways to translate a sentence, and many other ways of variable quality).

Task Complexity. Human computation (HCOMP) tasks span a vast range of complexity, from simple categorization tasks to highly complex work typically involving team coordination and/or highly skilled expertise. With simpler tasks, basic task designs and quality assurance methods suffice, whereas more complex work may require very different strategies, such as multi-stage workflow design and task decomposition strategies, especially when empowering less skilled workers to effectively complete more complex tasks [39]. A multi-stage *design pattern* for crowdsourcing might engage a second-stage crowd to select, verify, or correct responses from the first-stage [3, 45]. A well-known challenge, however, is that each new annotation task often requires designing a new, custom HCOMP workflow. This challenge has provoked much research to design more general workflows across annotation tasks [23, 24].

In general, it is useful to reduce human labor when effective automation exists. Our goal in this work is to advance the frontier of work that can be automated, by enabling general aggregation for more complex annotation tasks than is possible today. Workflow design is complementary, for coordinating hybrid human and AI sub-tasks, and for success on the "last mile" of difficult tasks that will always exist beyond AI's current frontier [15].

2.1 Problem Definition & Goal

Label aggregation is the task of inferring the correct label for a given item from a set of multiple annotations for that item. Typically this task is operationalized as selecting the best label for the item from the set of available annotations, although it can sometimes include label merging as in the averaging of numeric values [63]. *Offline, static* aggregation assumes the annotations are already collected (vs. online, dynamic control of which items to annotate). Research on the offline task has been especially vibrant in the HCOMP field, with many models proposed and benchmarked [19, 51, 63]. Prior work on general-purpose aggregation models has typically assumed simple annotations in which a small label space permits evaluating annotator performance based on exact match vs. gold or peer labels. While this paper continues the tradition of research into offline methods for selecting a best available label per item, it expands from prior work away from the assumptions of simple annotations with a small label space.

Complex label aggregation extends standard label aggregation to annotation types that could not be easily represented as a categorical variable or single-dimensional ordinal variable. Such tasks often involve a very large or infinite answer space, such that annotators are far less likely to produce identical labels for the same item. For example, there can be multiple acceptable ways to translate a sentence (and even more incorrect ways). As such, it appears that methods for assessing and aggregating complex annotations ought to be flexible enough to model relative label similarity between labels, beyond simple exact match.

2.2 Aggregation Methods and Models

As noted above, a variety of general-purpose and task-independent aggregation models exist for simple annotation tasks. We briefly discuss a few representative approaches.

Majority Voting (MV) is the simplest aggregation approach and avoids any modeling of workers or task. When most workers are accurate and have comparable accuracy, it can work quite well, and its being task-agnostic makes it quite versatile across diverse annotation tasks. However, as an unweighted voting method, it can perform poorly when the majority of workers are inaccurate, or worker accuracies are quite varied but not modeled. It also assumes a sufficiently small label space such that some workers will produce the same label, and thus a majority label can be found.

Dawid and Skene [8] proposed the now "classic" approach to simultaneously inferring worker reliability and label quality. Their unsupervised method was based on measuring peer agreement between annotators (i.e., a popularity contest for labels) to infer worker reliability and label quality. Interestingly, it was one of the first applications of the EM algorithm [10]. Despite the relative age and simplicity of this approach, the aforementioned benchmarking studies [19, 51, 63] showed Dawid-Skene (DS) to be remarkably robust across datasets.

Both DS and later Snow et al. [54] assume category-based annotation, modeling each annotator by a confusion matrix for their probability of producing a given categorical label given the gold categorical label. Firstly, note this approach cannot be directly applied to non-classification tasks, such as multiple-choice selection in which there is not a fixed set of categories to model across items. Secondly, for K categories we must estimate a $K \times K$ confusion matrix, which becomes more problematic for space and sparsity as K increases [27], such as with complex annotation tasks. Finally, these approaches assume we can measure peer agreement via exact-match between labels. As K grows, it is less likely annotators will produce the same label for a given item, and so exact-match becomes a harsher 0/1 loss function for testing assessor reliability.

ZenCrowd [9] can be interpreted as a simplified variant of DS (though it was proposed without reference to DS). Rather than representing each worker by a confusion matrix, each worker is instead modeled by a single reliability parameter. Similar unsupervised estimation is performed via EM. This simplified model can be more widely applied to non-categorical annotation tasks and is less prone to sparsity. However, as with DS, labels are compared based on exact-match. ZenCrowd is representative of a larger family of models having a single parameter for each annotator [63].

2.2.1 Semi-supervised Aggregation Models. The models described above are all unsupervised in that they can be trained without knowing gold annotations for any items. However, it is often the case that requesters have access to a number of gold annotations, for example when using honeypot questions. In these cases, these trusted annotations can be used to help aggregation models correctly estimate parameters through *semi-supervised learning*. Aggregation models using known-gold annotations for semi-supervised learning can achieve accuracy improvements on simple annotation tasks, especially when there are relatively few unsupervised annotations per item [17, 56, 58]. Semi-supervised learning is also stipulated to be essential when annotators overall exhibit low reliability or systematic bias in their responses [20].

2.2.2 Aggregation Models for Complex Tasks. Probabilistic models for annotations [42] provide a framework for several useful tools, including parameter inference, semi-supervised learning, and probabilistic task management. The main benefit of such models is their versatility. They can learn point-estimate or probabilistic properties of annotators and items together with inferred truth values. They do not require participation from experts or honeypot items but can benefit from semi-supervised learning. They can be used for decision-theoretic task control and active learning [7, 36].

For complex annotations of different types, formulating new probabilistic models is certainly doable but non-trivial. Some examples are a model based on Hidden Markov Models (HMM) that was developed to aggregate crowd-annotated sequences of text within documents [37] and a Chinese Restaurant Process (CRP) model for short free-response answers [26]. HMMs assume time-dependent data, and the CRP approach works when there are single discrete correct answers but not when there are continuous spaces of similarly correct ones. While it may be theoretically possible to use these models for semi-supervised learning, there is yet no such study of semi-supervised learning on complex annotation tasks.

Designing probabilistic models for complex tasks requires familiarity with the task domain. More generally, formulating models requires advanced mathematics and statistics knowledge, so it would be helpful to the science community if we could provide reusable models which can be more easily adopted by task requesters from diverse backgrounds. A key challenge is formulating the annotation likelihood conditioned on the unknown true value of the item plus any additional parameters that influence the error. If annotator labels L and true value estimators $\hat{L}$ are simple binary, categorical, or one-dimensional continuous variables, it is straightforward to define how they relate to each other according to common conditional probability distributions of the form:

$$P(L|\hat{L}, \theta) \tag{1}$$

with extra parameters θ to model effects like annotator reliability. However, when L and $\hat{L}$ are more complex, it becomes less obvious how to treat them in a model. Hidden variables representing complex concepts, such as $\hat{L}$, and their relationship to the observable L can be very difficult to formulate mathematically. For example, if the annotations are free text responses, the requester would not only have to decide on a latent representation or embedding space for sentences, but also figure out how to translate between that latent space of $\hat{L}$ and the observable space of text L, which is challenging.

Our goal is to provide a more flexible option for complex tasks: a general-purpose and task-independent probabilistic model for aggregating complex annotations.

3 METHODS

Here we describe our approach to modeling complex annotations without needing to define a task-specific probabilistic model for each new annotation task. Section 3.1 describes how we transform complex annotations datasets into distance datasets, allowing a probabilistic model to operate on simpler, task-agnostic continuous values. Our primary model, multidimensional annotation scaling (MAS), is next described in Section 3.2. Next, we introduce two simpler variant methods in Sections 3.3 and 3.4 which are faster but not as complete in the features they model.

3.1 From Annotations to Distances

Our key idea to obviate the need for task-specific models is to model distances between labels, rather than the labels themselves. The models we propose are agnostic to the distance function; we leave it to the *requester* (who defines the annotation task) to specify an appropriate distance function for the given task. Typically such distance functions already exist: as long as there is an *evaluation function* to quantify error in comparing predicted labels vs. gold labels, it can used to construct a distance function for our model.

Formally, a distance function should satisfy the following:

Non-negativity: $f(x,y) \geq 0$
Symmetry: $f(x,y) = f(y,x)$
Triangle Inequality: $f(x,y) \leq f(x,z) + f(z,y)$ for any z

In practice, the requester-supplied distance function need not meet all three of these requirements because its output can often be transformed to satisfy them. In particular, *non-negativity* can be satisfied by conversion to quantiles or exponentiation, and *symmetry* can be satisfied by adding (or averaging) $f(x,y)$ and $f(y,x)$ [52]. For example, Jensen-Shannon Divergence [13] is a symmetrized version of asymmetric Kullback-Liebler Divergence. We do encourage using distance functions that satisfy *triangle inequality* since we are still investigating the consequences of breaking this rule.

Annotator	Translation
1	Now Hamas and Israel should make peace so that this bloodshed comes to an end.
2	Hamas and Israel should reconcile so that this bloodshed comes to an end.
4	Now that the Hamas and Israel should be made to compromise, so the blood and evil.

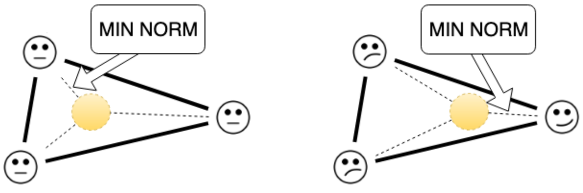
Annotator-1	Annotator-2	Annotation Distance
1	2	0.4333
1	4	0.8586
2	4	0.8758

Table 1: Example of input complex annotation dataset (top) converted to annotation distances (bottom). Section 4.1 describes the Urdo-to-English translation dataset shown here and the specific distance function used.

After selecting a distance function f, we induce a *distance dataset* D from the set of annotations by computing the distance between all pairs of labels for each example. **Table 1** shows a simple example of input annotations and output distances. This produces a symmetric matrices of distances $D_{iuv} = f(L_{iu}, L_{iv})$ between annotations by users (annotators) $u, v \in U$ for each item $i \in I$. In the extreme case of all users annotating all items, the total size of this distance dataset would be $\|U\|^2\|I\|$.

3.2 Multidimensional Annotation Scaling

Once a dataset of annotation distances is produced, we use it to train a distance-based annotation model. Such a model should infer true values for each item and might also infer helpful parameters describing user error and item difficulty. Whereas Equation (1) models annotations, we now instead model annotation distances with the conditional likelihood $P(D|\theta)$. This key transformation of the data allows our methods to work entirely on simple continuous space, both for observed and inferred variables, rather than on complex objects. This way, we avoid the main difficulty in designing probabilistic models for complex annotations.



(a) Equal-γ annotators **(b) Different-γ annotators**

Figure 1: Illustration of an item modeled by *multidimensional annotation scaling* (MAS). The emoji faces represent annotator labels, bold lines are observed distances between annotations, the golden circles are inferred true values, and dotted lines show the inferred ε for each annotation. When equal *user error* γ (Equation (5)) are learned for all annotators, the inferred true value is the geometric center. When different γ values are learned, the inferred true value is pulled closer to the more trusted annotators' labels.

Our proposed method to model $P(D|\theta)$ is inspired by Dawid and Skene [8] and intended to generalize a wide variety of aggregation models. The idea is to model a K-dimensional representation space in which the central point is taken as the estimated true item value, and annotation embeddings are estimated around that central point at norms regularized by expected user error. Much like word and sentence embeddings serve useful purposes in NLP [35], the annotation embeddings and other parameters produced by our model will be useful for our purposes.

In order to compute annotation embeddings, we devise a probabilistic model based on *multidimensional scaling* [34]. Multidimensional scaling is a method for estimating coordinates $\mathbf{x}$ of points given only a matrix of distances between those points by minimizing an objective function, generally $\sum(\|\mathbf{x}_i - \mathbf{x}_j\| - D_{ij})^2$. The estimated coordinate vectors carry meaning not in their absolute direction or magnitude, but rather in their position relative to each other. Multidimensional scaling can be seen as a generalization of kernel PCA when the kernel function is isotropic [60], and it is often used for dimensionality reduction and data visualization.

Our model, *multidimensional annotation scaling* (MAS), is a hierarchical Bayesian probabilistic model with a multidimensional scaling likelihood function, in which the estimated coordinates serve as annotation embeddings. Instead of the data populating a single distance matrix, each item has a separate annotation distance matrix. Additionally, because each user may annotate several items, we leverage the full dataset to compute *global* parameters representing annotator reliability, which serve as priors for the *local* parameters of each item's multidimensional scaling likelihood.

3.2.1 The MAS Model. We define the MAS model as follows in Equations (2)-(6) and illustrate the basic premise in Figure 1.

$$\hat{L}_i = L_{iu'_i}, \quad u'_i = \text{argmin}_{u \in U(i)} \varepsilon_{iu}, \tag{2}$$

$$\varepsilon^{\text{MAS}}_{iu} = \|\mathbf{x}_{iu}\| \tag{3}$$

For each item i, the model may select the "best" annotation as the true value estimator $\hat{L}_i$. This selected annotation is the one with the smallest inferred error ε_{iu} out of all annotations made by users $U(i)$ that worked on item i. Annotations may also be graded and ranked according to this ε_{iu}, which represents the model's predicted distance from annotation L_{iu} to the best possible annotation. In the MAS model, the origin in the space of embeddings $\mathbf{x}$ is taken to represent the true value for an item, so the norm of $\mathbf{x}_{iu}$ is understood as that annotation's distance from the truth, or ε_{iu}. This interpretation differs from standard multidimensional scaling, where the magnitude of the coordinates need not carry any meaning. In order to interpret ε_{iu} in this way, MAS assumes the annotation embedding space is *isotropic* because direction carries no meaning and *unimodal* because there is a single optimal point. While the concept of a single optimal point may seem inappropriate for some complex annotations tasks, it can be useful to model this way even for such tasks.

$$D_{iuv} \sim \mathcal{N}(\|\mathbf{x}_{iu} - \mathbf{x}_{iv}\|, \sigma)\ ,\ D_{iuv} \in \mathbb{R}_+ \tag{4}$$

Equation 4 is the generalized multidimensional scaling objective function expressed as a probabilistic likelihood. Maximizing the normal likelihood with free scale parameter σ minimizes the square error between observed distances in the data and learned distances in the embedding space.

$$\mathbf{x}_{iu} = \gamma_u \delta_i \frac{\tilde{\mathbf{x}}_{iu}}{\|\tilde{\mathbf{x}}_{iu}\|} \ , \ \gamma_u, \delta_i \in \mathbb{R}_+, \mathbf{x}_{iu}, \tilde{\mathbf{x}}_{iu} \in \mathbb{R}^K \tag{5}$$

The annotation embeddings $\mathbf{x}$ comprise normalized raw coordinates $\tilde{\mathbf{x}}$ as well as scale parameters γ representing user error and δ representing item difficulty. Normalizing the raw coordinates forces the scale parameters to entirely determine the embeddings' magnitudes. The model prefers to fit larger values of the scale parameters when those users and items are associated with larger distances in the data. When many annotations have small distances between each other, the model favors placing them closer to the origin compared to isolated annotations with higher distances from the others, thereby rewarding consensus. The model also favors placing annotations made by smaller-γ users closer to the center, thereby rewarding annotator reliability.

$$\log \gamma_u \sim \mathcal{N}(\log \bar{\gamma}, \Phi) \ , \ \log \delta_i \sim \mathcal{N}(\log \bar{\delta}, \Psi) \tag{6}$$

The parameters γ and δ are given hierarchical Bayesian priors with global location parameters $\bar{\gamma}$ and $\bar{\delta}$ and with configurable scales Φ and Ψ, respectively, which are set to 1 by default. The use of hierarchical Bayesian modeling reduces arbitrary choices of hyperparameters by allowing global parameters to be learned empirically, and it has been adopted in much of the recent work in label aggregation [4, 27, 47]. The only free hyperparameter left is the dimensionality K of the embedding space. We arbitrarily set $K = 8$ (untuned), slightly more than a typical five annotations per item. On simulated development data, varying the value of K did not have a major effect on results as long as $K > 2$.

3.2.2 Parameter estimation. We estimate MAS by maximizing the joint probability of Equations (4)-(6). We specify the model in the *Stan* probabilistic programming language [5]. Stan supports maximum a posteriori (MAP) estimation, variational inference (VI), and Markov chain Monte Carlo (MCMC). Our experiments run the fastest method, MAP, using Stan's default default L-BFGS optimization, until convergence or a maximum of 1500 iterations.

Free variables $\tilde{\mathbf{x}}$, γ, δ, $\bar{\gamma}$, and $\bar{\delta}$ are initialized randomly according to Stan's default settings, except for γ parameters, which are set to the average annotation distance of each user (i.e., the BAU score from Section 3.4 below). Because L-BFGS performs local optimization, it is helpful to initialize parameters with informed prior estimates when possible. In this case, BAU scores provide easy initial estimates for γ parameters. While δ parameters could be initialized to item-average distance, one depends on the other, so we use default estimates for δ parameters, as stated above.

3.2.3 Semi-supervised learning. Section 2.2.1 discussed the value of utilizing any available gold labels for semi-supervised estimation of an aggregation model. This is expected to be broadly useful in cases where peer-agreement alone cannot be replied upon as a dependable measure of annotator quality.

To achieve semi-supervised learning with MAS, we assign any gold labels in training the same user index $u = G$. We add these gold labels to the input annoation dataset and induce distances between annotations as usual. Next, during training a very informative prior is applied on the gold annotator's error parameter γ_G to impose a soft constraint on the model to recognize these annotations as near-optimal. Rather than setting the value of γ_G to a fixed small number, it is defined to be very small relative to the distribution of other users' γ values. In particular, we define

$$\log \gamma_G = \log \bar{\gamma} - 4\Phi \tag{7}$$

with the effect that users who tend to make annotations similar to known gold are placed near the origin in the embedding space, thereby reducing γ for those annotators.

3.3 Smallest Average Distance (SAD)

One simple variant of MAS, *Smallest Average Distance* (SAD), can be interpreted as a generalization of majority voting for complex annotations, operating entirely locally to each individual item. SAD assigns a score ε_{iu} to each annotation for item i by annotator $u \in U(i)$, equal to that annotation's average distance to all other annotations for the same item i. More formally, we calculate ε_{iu} as follows:

$$\varepsilon_{iu}^{\text{SAD}} = \frac{1}{\|S_{iu}\|} \sum S_{iu} \quad , \quad S_{iu} = \{D_{iuv} | v \in U, v \neq u\} \tag{8}$$

where S_{iu} denotes the set of all annotation distances for item i between annotator u and any other annotator $v \in U(i)$. SAD predicts this most central annotation, having the smallest average distance to all other annotations for item i, to be deemed the best consensus annotation for that item.

Relation to MAS. SAD is fastest to compute in its local formulation defined above, but MAS can also be configured to replicate SAD by setting hyper-parameters $\Psi = 0$, and $\Phi = 1$. Setting $\Psi = 0$ effectively treats all annotators as equal and therefore relies solely on distances within an item to estimate annotation quality.

Semi-supervised learning. Like majority vote, SAD does not model or utilize annotator reliability. SAD therefore cannot exploit semi-supervised learning as MAS can.

3.4 Best Available User (BAU)

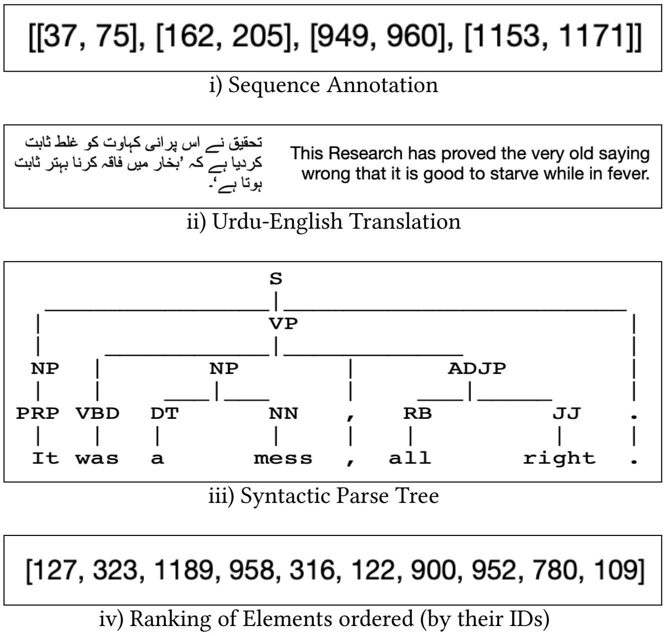
Whereas SAD operates entirely *local* to each item, BAU passes over the annotation distance dataset to estimate *global* annotator error across items. BAU assigns a ε_{iu} score to each annotation according to the annotator's estimated error ε_u, which is calculated as follows:

$$\varepsilon_{iu}^{\text{BAU}} = \varepsilon_u = \frac{1}{\|S_u\|} \sum S_u \quad , \quad S_u = \{D_{iuv} | i \in I, v \in U, v \neq u\} \tag{9}$$

This means that labels are scored entirely by their annotator's global reliability, regardless of the annotator's label for the particular item. BAU thus predicts the best label for each item to be whichever label came from the best available user (annotator) for that item. SAD and BAU thus present as contrasting extremes in exploiting local vs. global information in modeling.

Relation to MAS. Under the configuration of $\Psi = 1$ and $\Phi = 0$, MAS can approximate BAU if the learned values of γ have not strayed too far from their initialization. In practice, this may be useful for diagnostics or interpretation of results.

Semi-supervised learning. BAU can benefit from semi-supervised learning because annotators whose annotations are closer to known gold will be deemed more reliable.



[[37, 75], [162, 205], [949, 960], [1153, 1171]]

i) Sequence Annotation

تحقیق نے اس پرانی کہاوت کو غلط ثابت
کردیا ہے کہ 'بخار میں فاقہ کرنا بہتر ثابت
ہوتا ہے'۔

This Research has proved the very old saying wrong that it is good to starve while in fever.

ii) Urdu-English Translation

iii) Syntactic Parse Tree

[127, 323, 1189, 958, 316, 122, 900, 952, 780, 109]

iv) Ranking of Elements ordered (by their IDs)

Figure 2: Examples of complex annotations used in experiments: i) a list of ranges representing token sequences in a medical abstract; ii) a translation between two human languages; iii) a syntactic parse tree; and iv) ranking elements for an item (ordered by their ID numbers).

4 TASKS & DATASETS

In order to evaluate aggregation models for complex annotations, we need complex annotation datasets that include: i) multiple annotations per item, ii) associated annotator identifiers for each label, and iii) gold labels for evaluation. There are few public complex annotation datasets available that meet all three of the above requirements, for a variety of reasons.

Firstly, annotation was traditionally performed only by trusted annotators, with quality assurance performed by careful management and design of the annotation workflow. While inter-annotator agreement [1] was often quantified, using reliability models of trusted annotators has been less common [8, 22]. Secondly, while the advent of crowdsourcing simple labeling tasks has motivated aggregation modeling [54], complex tasks have lacked task-agnostic annotation models and been more difficult to crowdsource. A third issue is that datasets are typically released for public use only after internal quality assurance has been performed, reducing any multiple labels per item to single consensus labels. When multiple labels per item are included, annotator identifiers are often redacted rather than replaced by identifiers protecting the real identities of annotators.

Below we describe two real datasets for translation (Section 4.1) and sequence annotation (Section 4.2) which satisfy all three of the requirements above. In addition, we also generate synthetic datasets for two additional tasks, parsing (Section 4.3) and ranking (Section 4.4). Simulated experiments allow stress-testing under controlled conditions, whereas translations and sequences show empirical results on actual annotations. **Figure 2** shows illustrative examples of each task and **Table 2** summarizes key properties of each dataset.

4.1 Real Annotations: Translations

This dataset is a collection of Urdu-to-English translations made by non-professional translators [62]. We use the 293 sentences that have more than one translation. Gold translations come from professional translators. The final dataset we use contains 561 translations by 25 unique workers over these sentences. As seen in Table 2, the average number of annotations per item in the translation dataset is less than two. In fact, only a very small minority of the items in this dataset have more than two annotations available, making this a relatively challenging dataset to study consensus effects.

4.2 Real Annotations: Sequences

Nye et al. [40] and Nguyen et al. [37] share a dataset of 5,000 medical paper abstracts describing randomized control trials. Each abstract is annotated by roughly 5 Amazon Mechanical Turk workers. For each abstract, workers were asked to mark all text spans which identify the population enrolled in the clinical trial. Each token thus has a binary label: inside or outside of a span. They also collect gold annotations by medical students for a subset of 200 abstracts.

Out of the 200 abstracts with available gold, Nguyen et al. [37] share outputs from their Crowd-HMM model for 191 of the abstracts. We thus reduce the dataset used to only these 191 abstracts, with 1165 sequence annotations by 91 unique workers.

4.3 Synthetic Annotations: Parse Trees

Syntactic parsing represents a challenging annotation task which has traditionally required trained linguists. We selected this task for several reasons. Syntactic parsing has attracted great attention in the NLP community, and syntax trees clearly represent complex annotations. Such a difficult annotation task could reveal varying abilities with even trusted annotators which might be usefully modeled. Finally, given aggregation modeling support, we can envision ambitious crowdsourcing task designers pushing the envelope to engage the crowd in more complex tasks like this.

We focus specifically on constituency parsing, as embodied in the Penn Treebank (PTB) [31]. Given lack of a known dataset meeting Section 4's criteria (i-iii), we generate a synthetic dataset as follows. We randomly sample sentences of length 10 or more from PTB's Brown corpus [12]. We employ a diverse set of automatic parsing models included in NLTK [28]: the Charniak parser [33], MaltParser [38], and the Stanford Parser [30]. From each parser we generate a k-best set of candidate parses per sentence. Next, we evaluate the quality of each candidate parse vs. PTB's gold parse by the *EVALB* metric [50]. Finally, we merge all model output parses into a single ranking, ordered by decreasing EVALB score.

We simulate varying annotator accuracy by assigning each annotator $u \in U$ an overall accuracy parameter a_u from a beta distribution. We define two configurations for setting this parameter. In the "basic" setting, most annotators will be fairly accurate, sampling $\forall_u\, a_u \sim \text{beta}(4, 1)$. In contrast, the "noisy" configuration assumes low accuracy workers are more prevalent, sampling $\forall_u\, a_u \sim \text{beta}(3, 2)$. **Figures 4a** and **4b** show histograms for these two distributions.

Dataset	Type	Annotators	Items	Labels	Identical #(%)	Labels/User	Labels/Item	Section
Translations	Real	25	293	561	9 (1.6%)	22.4±20.5	1.9±0.5	4.1
Sequences	Real	91	191	1165	87 (7.5%)	12.8±18.4	6.1±1.2	4.2
Parses (basic)	Synthetic	30	100	600	59 (9.8%)	20	6±2.39	4.3
Parses (noisy)	Synthetic	30	100	600	84 (14%)	20	6±2.25	4.3
Rankings	Synthetic	30	100	600	2 (0.3%)	20	6±2.34	4.4

Table 2: Datasets used and summary statistics. The number of annotators $||U||$, number of items $||I||$, and number of annotations $||L||$ vary by dataset. The number of identical labels, i.e. those occurring more than once, is typically very small for complex annotation tasks, but with some variation between datasets. The number of labels per user and labels per item suggests how much information the model can learn from the data about user and item-level parameters, respectively.

We randomly assign 20% of the sentences to each annotator. For each sentence i and annotator u, we generate an item-level error parameter e_{iu} from a Normal distribution $e_{iu} \sim N(0, 0.1)$. The annotator's adjusted accuracy for the given sentence will then be $(a_u + e_{iu})$. Finally, the simulator "generates" annotator u's parse for sentence i by selecting the output model parse having EVALB score s best matching the adjusted accuracy, i.e., minimizing $|s - (a_u + e_{iu})|$.

4.4 Synthetic Annotations: Element Rankings

This task involves ranking a set of top elements for each question (item). For example, users might be asked to name and sort the ten largest countries by population, the five best-selling fiction books of 2018 by sales volume, or the three richest politicians in Europe.

We assume 100 such items, each having 50 elements, and respondents needing to identify and rank the top 10 elements for each item. Our simulator generates a "true score" g_e for each element e in an item from a standard normal distribution, and a gold ranking over elements for each item is induced from these scores.

Top-10 rankings for each worker are simulated by sorting the top ten elements by the worker's "perceived score". The perceived score is drawn from a normal distribution with location = g_e and scale = $\sigma_u \sigma_i$. These σ parameters simulate worker skill and item difficulty. To simulate variation over users and items, each σ_u is drawn from a Uniform(0,1) distribution.

5 EXPERIMENTS

Validation vs. testing. Model development was performed entirely on the Rankings task, leading to bug fixes and modeling improvements but no parameter tuning. The other three tasks (translations, sequence annotation, and parsing) were reserved for final testing of model generalization.

5.1 Methods

5.1.1 General Baselines. We are not aware of any task-independent models for aggregating complex annotations to compare our models against. Instead, we compare to a variety of other baselines.

Random User (RU). The simplest baseline is to choose a random user's annotation for each item. This represents the scenario as if only a single label were collected per item. We report the RU performance as an average over five trials, sampling with replacement from available labels.

ZenCrowd (ZC) (Section 2.2) models each worker's probability of providing correct labels and effectively performs weighted voting. ZC thus represents an improvement over (unweighted) majority voting, provided worker accuracy estimates are reasonable.

5.1.2 Sequence task baselines. **Token-wise Majority Vote (TMV)**. In addition to comparisons against prior work, Nguyen et al. [37] report a simple baseline which breaks sequence annotations into individual tokens and then performs a token-wise majority vote.

Crowd-HMM (CHMM). Nguyen et al. [37] proposed a novel bespoke Crowd-HMM probabilistic model for sequence labeling. Using the same dataset that we also adopt in this study, they show empirical improvement of Crowd-HMM over prior work [18, 48].

Note: Nguyen et al. [37] use 4,800 abstracts without gold for unsupervised training, whereas we limit ourselves to the 191 abstracts shared by the authors with Crowd-HMM model outputs. Our empirical results thus potentially underestimate the relative performance of our distance-based models in comparison to Crowd-HMM.

5.1.3 Oracle. Finally, we compare to an upper-bound "Oracle", which selects the best annotation per item by cheating, using the gold to pick the best annotation according to the evaluation metric.

5.2 Evaluation Metrics & Annotation Distances

For each complex annotation task (Section 4), a different evaluation metric is often warranted. We summarize below the metrics used, then discuss how each is used to induce a distance function for our distance-based aggregation models (Section 3).

5.2.1 Evaluation Metrics. We report a single metric for each task, each measuring annotation quality (larger is better). All of the metrics we adopt are defined in the range [0, 1], though any arbitrary evaluation metric could be used in practice.

Translation. For evaluating worker translations against gold translations we use the GLEU score [61], which is a variant of the BLEU [41] score but specialized for comparisons between individual sentences. We use the NLTK [28]'s implementation.

Sequence Annotation. We adopt the same F1 evaluation metric reported by Nguyen et al. [37]. They first define textual span-based

Experiment		Baselines				Our Work					Upperbound
Task	Evaluation Metric	RU	ZC	TMV	CHMM	SAD	BAU	MAS	S-BAU	S-MAS	Oracle
Translations	GLEU	0.185	0.188	-	-	0.198	0.216	**0.217**	0.220	**0.227**	0.246
Sequences	F1	0.561	0.569	0.652	0.702	0.663	0.669	**0.709**	0.677	**0.709**	0.827
Parses (basic)	EVALB	0.812	0.819	-	-	0.850	0.877	**0.932**	0.902	**0.933**	0.939
Parses (noisy)	EVALB	**0.705**	0.702	-	-	0.675	0.640	0.655	0.641	**0.756**	0.830
Rankings	Kendall τ	0.491	0.495	-	-	0.680	0.697	**0.710**	0.697	**0.711**	0.724

Table 3: Results of 1-best evaluation. Evaluation metrics vary by task, but larger is better. The best unsupervised result in each row is bolded. Lesser results whose difference is not statistically significant at the 0.05 level are underlined. Semi-supervised results S-BAU and S-MAS are similarly bolded or underlined.

precision and recall metrics as follows:

$$\text{Precision } P = \frac{\text{\# true positive tokens}}{\text{\# tokens in labeled span}} \tag{10}$$

$$\text{Recall } R = \frac{\text{\# true positive tokens}}{\text{\# tokens in gold span}} \tag{11}$$

They they average these span-based P and R metrics over all spans, and report F1 as the harmonic mean of these P and R averages.

Parsing. We use EVALB for evaluating annotations against gold.

Ranking. We report Kendall [21]'s τ correlation to evaluate the position of elements in annotated ranked lists against their positions in the gold ranked lists. While the gold list contains an exhaustive ranking over all elements, the annotation task is only to rank the top 10 elements. In evaluating annotator rankings, we assume any element not in the top 10 of an annotator list is considered to be "tied for last place" and assigned the maximum position.

5.2.2 From Metrics to Distances. As discussed earlier, the models we propose are agnostic to the distance function. In general, such distance functions already exist: as long as we can quantify error in comparing predicted labels vs. gold labels (i.e., an evaluation metric), the same error measure can used as a convenient distance function for our model. We adopt this approach in our study and leave for future work investigation of how alternative distance functions interact with choice of evaluation metric.

All evaluation metrics we report quantify the annotation quality q. Since all are conveniently defined over $[0, 1]$, we can induce an error measure e by simply taking $e = 1 - q$. In general, non-bounded metrics would require empirically identifying the maximum score quality in order to convert from quality to error.

Translation is the only task we consider in which the the evaluation metric (GLEU) violates the symmetry requirement for distance functions (Section 3.1). As described there, we apply the general approach of symmetrizing the metric by computing in both directions and averaging, then proceeding as with other metrics.

$$f(x, y) = 1 - 0.5(\mathbb{GLEU}(x, y) + \mathbb{GLEU}(y, x)) \tag{12}$$

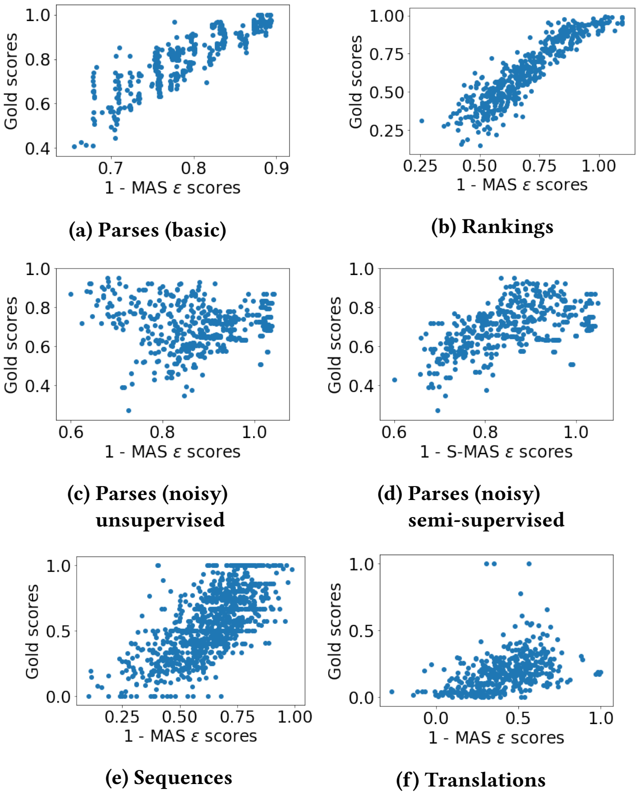
Sequence Annotation. We use $f(x, y) = 1 - \text{F1}(x, y)$.

Parsing. The distance function is $f(x, y) = 1 - \text{EVALB}(x, y)$.

Ranking. The distance function is $f(x, y) = 1 - \tau(x, y)$.

5.3 Evaluation: *1-Best*

Canonical *1-best evaluation* evaluates how well aggregation models choose a single best *consensus* annotation for each item from the set of available annotations for that item. As with the above Oracle upper-bound, evaluation assumes a reference gold standard for each task against which other annotations can be compared. The evaluation metric for each task is specified along with its corresponding dataset in Section 4. We report mean performance of each method across the items in each dataset. A two-tailed paired t-test is also conducted to measure statistical significance of difference in mean performance across items between methods for each dataset. Results for 1-best evaluation are shown in **Table 3**.



(a) Parses (basic)

(b) Rankings

(c) Parses (noisy) unsupervised

(d) Parses (noisy) semi-supervised

(e) Sequences

(f) Translations

Figure 3: Correlation between MAS model scores for annotations vs. *gold scores*: actual annotation quality (as measured by the evaluation metric with reference to the gold annotation). Specifically, we scatterplot (1 - MAS score) versus gold scores over all annotations in each dataset.

Real datasets. In the translation task, MAS outperforms other distance-based models, which themselves outperform RU and ZC. For the sequence annotation task, all distance-based methods outperform TMV, which itself is still much better than RU. MAS is the best performing distance-based method, remarkably achieving performance on par with the CHMM probabilistic model specially designed for this task.

Synthetic datasets. Parse tree and ranking task results show that our distance-based models (BAU, SAD, and MAS) outperform baselines under normal conditions, with MAS generally strongest. The exception is with the "noisy" parsing task, in which low accuracy workers are more prevalent. Here is more consensus among erroneous annotators than among reliable ones, so consensus-based methods even underperform RU. This situation is a compelling reason for using semi-supervised learning (Section 5.5).

ZenCrowd. Across tasks, ZC performs nearly identically to RU. The likely culprit for ZC's lackluster performance is the large label space of complex annotation (Section 2.1), leading to poor annotator accuracy estimates for weighted voting. With a large label space, annotators will rarely produce identical labels (see Table 2), and thus any model estimating annotator reliability based on exact match of labels may struggle to learn meaningful reliability weights. Our results for ZC are likely indicative of a larger family of existing, similar annotation models which estimate annotator reliability based on exact match between labels [63].

5.4 Evaluation: *Score-all*

Whereas *1-best evaluation* above only evaluates an aggregation model's ability to select the best annotation, one might want to identify the top-k annotations to keep, the bottom-k annotations to discard, or to discard entire items if no annotations of sufficient quality are given. In short, it can be useful to evaluate how reliably the model scores all annotations.

To evaluate this, we measure how closely model scores for annotations correlate with actual annotation quality (as measured by the evaluation metric with reference to the gold annotation). We refer to the latter as *gold scores*, and we measure Pearson correlation between model vs. gold scores over all annotations. This correlation measure serves as a rough estimate of how accurately each annotation's distance to gold could be predicted, if the requester were interested in selecting annotations by threshold rather than just choosing the best.

Table 4 displays results for distance-based methods. Results are similar to the 1-best evaluation, with MAS typically performing best. **Figure 3** displays scatterplots between annotation quality predicted by MAS and actual gold scores for each of the five datasets. These scatterplots provide a more visual diagnostic than the Pearson correlations used to summarize performance.

The score-all results demonstrate the relative difficulty of aggregating these different datasets. The basic parses and rankings simulations are the easiest, with very good fit between predicted and actual annotation quality. Translations and the noisy parses are the most difficult, but for different reasons. For noisy parses, the high degree of user error causes unsupervised consensus estimates to be of low quality, but this is overcome by introducing a small amount of supervision. For translations, the problem is likely due in part to having too few annotations per item.

Task	SAD	BAU	MAS	S-BAU	S-MAS
Translations	0.18	0.20	**0.22**	0.24	**0.38**
Sequences	0.63	0.51	**0.65**	0.52	**0.69**
Parses (basic)	0.48	0.72	**0.83**	0.78	**0.85**
Parses (noisy)	**0.19**	-0.64	0.06	-0.63	**0.50**
Rankings	0.80	0.83	**0.85**	0.83	**0.86**

Table 4: Experimental results of score-all evaluation, measuring Pearson correlation between distance models' predicted annotation quality and their gold scores.

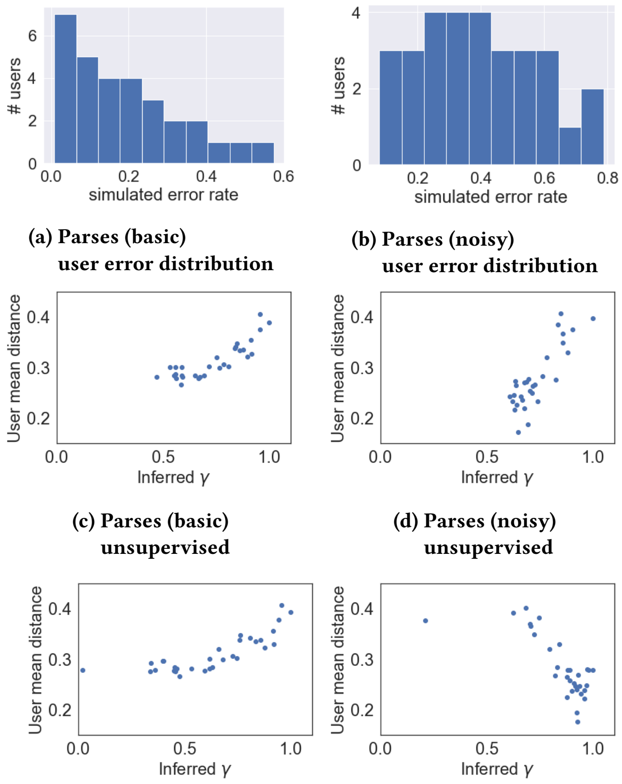
5.5 Evaluation: Semi-Supervised Learning

Section 5.3's 1-best results suggest that when low accuracy annotators are more prevalent (i.e., in the "noisy" parsing task), consensus-based methods perform worse than simply selecting an annotation at random (RU). This result for complex annotations is consistent with prior findings for simple annotations (Section 2.2.1): purely unsupervised aggregation models struggle when peer-agreement does not provide a reliable measure of label quality.

Section 3.2.3 described how this might be addressed in our distance-based models by exploiting semi-supervised learning (when gold annotations are available). To test this approach empirically, we reserve 10% of items as known-gold and remove them from testing data. Note that for all experiments (unsupervised and semi-supervised), these same known-gold items were removed from the test set to ensure fair comparison of unsupervised vs. semi-supervised settings for model estimation.

For the most part, semi-supervised results are the same, matching or slightly exceeding unsupervised results. See 1-best results (Table 3) and score-all results (Table 4). For example, on the translations task, semi-supervised S-MAS achieves 1-best 0.227 GLEU vs. unsupervised 0.217. In contrast, semi-supervised estimation yields drastic S-MAS improvement for the "noisy" parsing case. For 1-best, the change is from 0.655 to 0.756 1-best EVALB score (15% improvement, statistically significant). For score-all, change is from 0.06 Pearson correlation to 0.50. However, semi-supervised BAU, which only updates the scores for annotators that worked on known-gold, performs the same as unsupervised BAU. MAS benefits far more than BAU by using known-gold to update inferred reliability for all annotators, through iterative parameter inference.

How MAS uses semi-supervised learning to succeed in the "noisy" parsing task is depicted in **Figure 4**. Unsupervised MAS learns user error γ parameters that correlate closely with BAU's calculated user average distances, as seen in Figure 4d. In this unsupervised case both MAS and BAU suffer because their most trusted annotators are actually the most erroneous. When using semi-supervised learning, however, MAS is able to rearrange its inferences for annotator reliability in the reverse direction, as seen in Figure 4f. The smallest-γ point in this scatterplot represents the gold user, and MAS iteratively shrinks the γ of all users whose annotations are generally small in distance to other small-γ users such as the gold user. The process repeats until even the γ values of users who did not annotate a known-gold item are affected.



(a) Parses (basic) user error distribution

(b) Parses (noisy) user error distribution

(c) Parses (basic) unsupervised

(d) Parses (noisy) unsupervised

(e) Parses (basic) semi-supervised

(f) Parses (noisy) semi-supervised

Figure 4: Insights from varying annotator accuracies in the parsing task. Histograms of simulated user error and scatterplots comparing MAS γ values per user against average user distance (BAU's ε score), for the basic and noisy configurations. In the semi-supervised cases, inference of the γ parameters by MAS is assisted by the "gold" user represented by the left-most point in each plot. In the noisy simulator configuration, this reordering against consensus is what allows semi-supervised MAS to outperform even as other user-weighted methods underperform against RU.

Ultimately, results show that findings of prior semi-supervised work for simple labeling tasks [56, 58] seem to carry over to complex labeling tasks, with our task-agnostic distance-based model able to similarly exploit and benefit from semi-supervised training.

6 DISCUSSION & CONCLUSION

Training data is the fuel of modern AI. As we seek to grow AI capabilities to accomplish more complex prediction tasks, we will need quality annotations for those tasks. We can expect more varied performance across annotators as annotation task difficulty increases, and as diverse crowdsourced annotators are engaged to tackle ever-more challenging annotation tasks.

We have proposed distance-based aggregation models for complex annotations beyond binary, categorical, or ordinal variables. Our Multidimensional Annotation Scaling (MAS) method and its variants bypass the challenge of having to define task-specific probabilistic models for each new type of complex annotation by instead modeling annotation distances, which can often be easily induced from existing evaluation metrics. Consequently, these distance-based methods can be used with little alteration across a wide variety of tasks. Results on four types of tasks producing complex annotations – sequence labeling, translation, syntactic parsing, and ranking – show improvement over general baselines and even match performance with a task-specific probabilistic model for the sequence task. MAS thus appears to be useful, both for practical adoption and as a baseline against which new, bespoke aggregation models for complex annotations can be benchmarked.

In principle, customized models for specific tasks (e.g., Crowd-HMM) can certainly perform better than general-purpose alternatives like MAS, but at the cost of greater complexity and additional time and expertise to design. The question is how much added benefit a custom model may deliver as return-on-investment vs. using an off-the-shelf, task-agnostic model such as MAS? It is also worth framing MAS in the context of two extremes: the cheapest option – only collecting one annotation per item (i.e., RU) – and the most expensive option – designing a custom probabilistic annotation model or custom human computation workflow for each new annotation task. In this context, MAS offers an alternative for automatically aggregating multiple complex annotations with minimal extra work for the requester.

Many research questions remain regarding effective collection of complex annotations. One idea for future work is to extend MAS to support complex tasks without assuming the annotation space is isotropic and unimodal (Section 3.2.1). This could extend MAS beyond *objective* tasks to also support *subjective* tasks [57], which permit a space of wider and more uneven valid responses.

Whereas aggregation has been traditionally formulated as selecting between available annotator labels for a given item [8], further gains might be had by merging multiple annotator labels. In order to merge labels while keeping the model task-agnostic, we might allow requesters to supply a task-specific merge function, such as some existing method for merging rankings [2, 46]. The task-agnostic model could then exploit this task-specific merge function to potentially achieve superior consensus labels.

Our aggregation model clearly requires having some distance function, and intuition suggests that "better" distance functions should yield better performance (e.g., aligning choice of distance function with the evaluation metric being optimized). It would also be interesting to investigate interactions between choice of distance function and MAS likelihood function. We have used normal likelihood (Equation 4), corresponding to square loss, but many other alternatives could be explored [29]. With transformations of the data or the likelihood function, when appropriate, the MAS model should be capable of improvement akin to other regression models.

ACKNOWLEDGMENTS

We thank the reviewers for their feedback on our submission and the crowd workers for the data they contributed for this research study. This was was supported in part by National Science Foundation grant No. 1253413. Any opinions, findings, and conclusions or recommendations expressed by the authors are entirely their own and do not represent those of the sponsoring agencies.

REFERENCES

[1] Ron Artstein and Massimo Poesio. 2008. Inter-coder agreement for computational linguistics. *Computational Linguistics* 34, 4 (2008), 555–596.

[2] Javed A Aslam and Mark Montague. 2001. Models for metasearch. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 276–284.

[3] Michael S Bernstein, Greg Little, Robert C Miller, Björn Hartmann, Mark S Ackerman, David R Karger, David Crowell, and Katrina Panovich. 2010. Soylent: a word processor with a crowd inside. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*. ACM, 313–322.

[4] Bob Carpenter. 2008. Multilevel bayesian models of categorical data annotation. *Unpublished manuscript* 17, 122 (2008), 45–50.

[5] Bob Carpenter, Andrew Gelman, Matthew D Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. 2017. Stan: A probabilistic programming language. *Journal of statistical software* 76, 1 (2017).

[6] Quanze Chen, Jonathan Bragg, Lydia B Chilton, and Daniel S Weld. 2018. Cicero: Multi-Turn, Contextual Argumentation for Accurate Crowdsourcing. *arXiv preprint arXiv:1810.10733* (2018).

[7] Peng Dai, Daniel Sabey Weld, et al. 2010. Decision-theoretic control of crowd-sourced workflows. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*.

[8] Alexander Philip Dawid and Allan M Skene. 1979. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Applied statistics* (1979), 20–28.

[9] Gianluca Demartini, Djellel Eddine Difallah, and Philippe Cudré-Mauroux. 2012. ZenCrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In *Proceedings of the 21st international conference on World Wide Web*. ACM, 469–478.

[10] Arthur P Dempster, Nan M Laird, and Donald B Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)* 39, 1 (1977), 1–22.

[11] Suyog Dutt Jain and Kristen Grauman. 2013. Predicting sufficient annotation strength for interactive foreground segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*. 1313–1320.

[12] W Nelson Francis and Henry Kucera. 1979. Brown Corpus manual: Manual of information to accompany a standard corpus of present-day edited American English for use with digital computers. *Brown University, Providence, Rhode Island, USA* (1979).

[13] Bent Fuglede and Flemming Topsoe. 2004. Jensen-Shannon divergence and Hilbert space embedding. In *International Symposium onInformation Theory, 2004. ISIT 2004. Proceedings.* IEEE, 31.

[14] Tanya Goyal, Tyler McDonnell, Mucahid Kutlu, Tamer Elsayed, and Matthew Lease. 2018. Your Behavior Signals Your Reliability: Modeling Crowd Behavioral Traces to Ensure Quality Relevance Annotations. In *6th AAAI Conference on Human Computation and Crowdsourcing (HCOMP)*. 41–49.

[15] Mary L Gray and Siddharth Suri. 2019. *Ghost Work: How to Stop Silicon Valley from Building a New Global Underclass.* Eamon Dolan Books.

[16] David Ha and Douglas Eck. 2017. A neural representation of sketch drawings. *arXiv preprint arXiv:1704.03477* (2017).

[17] Dirk Hovy, Taylor Berg-Kirkpatrick, Ashish Vaswani, and Eduard Hovy. 2013. Learning whom to trust with MACE. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 1120–1130.

[18] Ziheng Huang, Jialu Zhong, and Rebecca J. Passonneau. 2015. Estimation of Discourse Segmentation Labels from Crowd Data. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, 2190–2200. http://aclweb.org/anthology/D15-1261

[19] Nguyen Quoc Viet Hung, Nguyen Thanh Tam, Lam Ngoc Tran, and Karl Aberer. 2013. An evaluation of aggregation techniques in crowdsourcing. In *International Conference on Web Information Systems Engineering*. Springer, 1–15.

[20] Panos Ipeirotis. 2010. A Computer Scientist in a Business School. https://www.behind-the-enemy-lines.com/2010/09/worker-evaluation-in-crowdsourcing-gold.html September 15.

[21] Maurice G Kendall. 1938. A new measure of rank correlation. *Biometrika* 30, 1/2 (1938), 81–93.

[22] Won Kim and W John Wilbur. 2010. Improving a gold standard: Treating human relevance judgments of MEDLINE document pairs. In *2010 Ninth International Conference on Machine Learning and Applications*. IEEE, 491–498.

[23] Aniket Kittur, Boris Smus, Susheel Khamkar, and Robert E Kraut. 2011. Crowdforge: Crowdsourcing complex work. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*. ACM, 43–52.

[24] Anand Kulkarni, Matthew Can, and Björn Hartmann. 2012. Collaboratively crowdsourcing workflows with turkomatic. In *Proceedings of the acm 2012 conference on computer supported cooperative work*. ACM, 1003–1012.

[25] Yuncheng Li, Yale Song, Liangliang Cao, Joel Tetreault, Larry Goldberg, Alejandro Jaimes, and Jiebo Luo. 2016. TGIF: A new dataset and benchmark on animated GIF description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4641–4650.

[26] Christopher H Lin, Mausam Mausam, and Daniel S Weld. 2012. Crowdsourcing control: Moving beyond multiple choice. In *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence*.

[27] Chao Liu and Yi-Min Wang. 2012. TrueLabel+ confusions: a spectrum of probabilistic models in analyzing multiple ratings. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*. Omnipress, 17–24.

[28] Edward Loper and Steven Bird. 2002. NLTK: the natural language toolkit. *arXiv preprint cs/0205028* (2002).

[29] Yuanhua Lv and ChengXiang Zhai. 2009. Positional language models for information retrieval. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*. ACM, 299–306.

[30] Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*. 55–60.

[31] Mitchell Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. (1993).

[32] Catherine C Marshall and Frank M Shipman. 2013. Experiences surveying the crowd: Reflections on methods, participation, and reliability. In *Proceedings of the 5th Annual ACM Web Science Conference*. ACM, 234–243.

[33] David McClosky, Eugene Charniak, and Mark Johnson. 2006. Effective self-training for parsing. In *Proceedings of the main conference on human language technology conference of the North American Chapter of the Association of Computational Linguistics*. Association for Computational Linguistics, 152–159.

[34] Al Mead. 1992. Review of the development of multidimensional scaling methods. *Journal of the Royal Statistical Society: Series D (The Statistician)* 41, 1 (1992), 27–39.

[35] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[36] An Thanh Nguyen, Byron C Wallace, and Matthew Lease. 2015. Combining crowd and expert labels using decision theoretic active learning. In *Third AAAI conference on human computation and crowdsourcing*.

[37] An T Nguyen, Byron C Wallace, Junyi Jessy Li, Ani Nenkova, and Matthew Lease. 2017. Aggregating and predicting sequence labels from crowd annotations. In *Proceedings of the conference. Association for Computational Linguistics. Meeting*, Vol. 2017. NIH Public Access, 299.

[38] Joakim Nivre, Johan Hall, Jens Nilsson, Atanas Chanev, Gülşen Eryigit, Sandra Kübler, Svetoslav Marinov, and Erwin Marsi. 2007. MaltParser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering* 13, 2 (2007), 95–135.

[39] Jon Noronha, Eric Hysen, Haoqi Zhang, and Krzysztof Z Gajos. 2011. Platemate: crowdsourcing nutritional analysis from food photographs. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*. 1–12.

[40] Benjamin Nye, Junyi Jessy Li, Roma Patel, Yinfei Yang, Iain J Marshall, Ani Nenkova, and Byron C Wallace. 2018. A corpus with multi-level annotations of patients, interventions and outcomes to support language processing for medical literature. In *Proceedings of the conference. Association for Computational Linguistics. Meeting*, Vol. 2018. NIH Public Access, 197.

[41] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics, 311–318.

[42] Rebecca J Passonneau and Bob Carpenter. 2014. The benefits of a model of annotation. *Transactions of the Association for Computational Linguistics* 2 (2014), 311–326.

[43] Silviu Paun, Jon Chamberlain, Udo Kruschwitz, Juntao Yu, and Massimo Poesio. 2018. A probabilistic annotation model for crowdsourcing coreference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 1926–1937.

[44] Silviu Paun and Dirk Hovy. 2019. EMNLP Workshop on Aggregating and Analysing Crowdsourced Annotations for NLP. Association for Computational Linguistics, Hong Kong. http://dali.eecs.qmul.ac.uk/annonlp

[45] Matt Post, Chris Callison-Burch, and Miles Osborne. 2012. Constructing parallel corpora for six indian languages via crowdsourcing. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*. Association for Computational Linguistics, 401–409.

[46] Tao Qin, Xiubo Geng, and Tie-Yan Liu. 2010. A new probabilistic model for rank aggregation. In *Advances in neural information processing systems*. 1948–1956.

[47] Vikas C Raykar, Shipeng Yu, Linda H Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy. 2010. Learning from crowds. *Journal of Machine Learning Research* 11, Apr (2010), 1297–1322.

[48] Filipe Rodrigues, Francisco Pereira, and Bernardete Ribeiro. 2014. Sequence labeling with multiple annotators. *Machine learning* 95, 2 (2014), 165–181.

[49] Jeffrey M Rzeszotarski and Aniket Kittur. 2011. Instrumenting the crowd: using implicit behavioral measures to predict task performance. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*. ACM, 13–22.

[50] Satoshi Sekine and Michael Collins. 1997. *EvalB: a bracket scoring program*. http://nlp.cs.nyu.edu/evalb/

[51] Aashish Sheshadri and Matthew Lease. 2013. SQUARE: A Benchmark for Research on Computing Crowd Consensus. In *Proceedings of the 1st AAAI Conference on Human Computation (HCOMP)*. 156–164.

[52] Yu A Shreider. 1974. What Is Distance? Popular Lectures in Mathematics. (1974).

[53] Padhraic Smyth, Usama M Fayyad, Michael C Burl, Pietro Perona, and Pierre Baldi. 1995. Inferring ground truth from subjective labelling of venus images. In *Advances in neural information processing systems*. 1085–1092.

[54] Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y Ng. 2008. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *Proceedings of the conference on empirical methods in natural language processing*. Association for Computational Linguistics, 254–263.

[55] Qi Su, Dmitry Pavlov, Jyh-Herng Chow, and Wendell C Baker. 2007. Internet-scale collection of human-reviewed data. In *Proceedings of the 16th international conference on World Wide Web*. ACM, 231–240.

[56] Wei Tang and Matthew Lease. 2011. Semi-supervised consensus labeling for crowdsourcing. In *SIGIR 2011 workshop on crowdsourcing for information retrieval (CIR)*. 1–6.

[57] Yuandong Tian and Jun Zhu. 2012. Learning from crowds in the presence of schools of thought. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 226–234.

[58] Jing Wang, Panagiotis G Ipeirotis, and Foster Provost. 2011. Managing crowdsourcing workers. In *The 2011 winter conference on business intelligence*. Citeseer, 10–12.

[59] Jacob Whitehill, Ting-fan Wu, Jacob Bergsma, Javier R Movellan, and Paul L Ruvolo. 2009. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *Advances in neural information processing systems*. 2035–2043.

[60] Christopher KI Williams. 2001. On a connection between kernel PCA and metric multidimensional scaling. In *Advances in neural information processing systems*. 675–681.

[61] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144* (2016).

[62] Omar F Zaidan and Chris Callison-Burch. 2011. Crowdsourcing translation: Professional quality from non-professionals. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, 1220–1229.

[63] Yudian Zheng, Guoliang Li, Yuanbing Li, Caihua Shan, and Reynold Cheng. 2017. Truth inference in crowdsourcing: Is the problem solved? *Proceedings of the VLDB Endowment* 10, 5 (2017), 541–552.