Importance sampling for unbiased on-demand evaluation of knowledge base population

Arun Tejasvi Chaganty* and **Ashwin Pradeep Paranjape*** and **Percy Liang**
Computer Science Department
Stanford University
{chaganty,ashiwnp,pliang}@cs.stanford.edu

Christopher D. Manning
Computer Science Department
Stanford University
{manning}@cs.stanford.edu

Abstract

Knowledge base population (KBP) systems take in a large document corpus and extract entities and their relations. Thus far, KBP evaluation has relied on judgements on the pooled predictions of existing systems. We show that this evaluation is problematic: when a new system predicts a previously unseen relation, it is penalized even if it is correct. This leads to significant bias against new systems, which counterproductively discourages innovation in the field. Our first contribution is a new importance-sampling based evaluation which corrects for this bias by annotating a new system's predictions on-demand via crowdsourcing. We show this eliminates bias and reduces variance using data from the 2015 TAC KBP task. Our second contribution is an implementation of our method made publicly available as an online KBP evaluation service. We pilot the service by testing diverse state-of-the-art systems on the TAC KBP 2016 corpus and obtain accurate scores in a cost effective manner.

1 Introduction

Harnessing the wealth of information present in unstructured text online has been a long standing goal for the natural language processing community. In particular, knowledge base population seeks to automatically construct a knowledge base consisting of relations between entities from a document corpus. Knowledge bases have found many applications including question answering (Berant et al., 2013; Fader et al., 2014; Reddy et al., 2014), automated reasoning (Kalyanpur et al., 2012) and dialogue (Han et al., 2015).

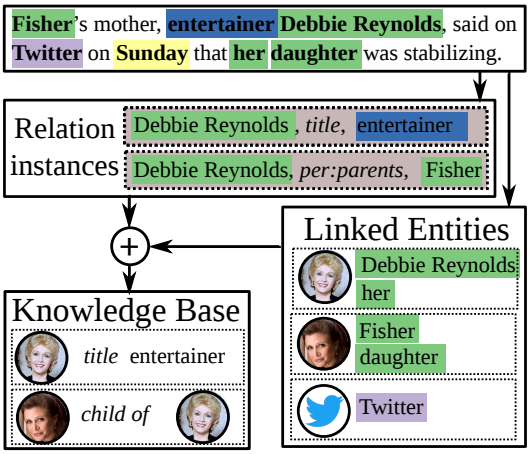


Figure 1: An example describing entities and relations in knowledge base population.

Evaluating these systems remains a challenge as it is not economically feasible to exhaustively annotate every possible candidate relation from a sufficiently large corpus. As a result, a pooling-based methodology is used in practice to construct datasets, similar to them methodology used in information retrieval (Jones and Rijsbergen, 1975; Harman, 1993). For instance, at the annual NIST TAC KBP evaluation, all relations predicted by participating systems are pooled together, annotated and released as a dataset for researchers to develop and evaluate their systems on. However, during development, if a new system predicts a previously unseen relation it is considered to be wrong even if it is correct. The discrepancy between a system's true score and the score on the pooled dataset is called *pooling bias* and is typically assumed to be insignificant in practice (Zobel, 1998).

The key finding of this paper contradicts this assumption and shows that the pooling bias is actu-

* Authors contributed equally.

ally significant, and it penalizes newly developed systems by 2% F_1 on average (Section 3). Novel improvements, which typically increase scores by less than 1% F_1 on existing datasets, are therefore likely to be clouded by pooling bias during development. Worse, the bias is larger for a system which predicts qualitatively different relations systematically missing from the pool. Of course, systems participating in the TAC KBP evaluation do not suffer from pooling bias, but this requires researchers to wait a year to get credible feedback on new ideas.

This bias is particularly counterproductive for machine learning methods as they are trained assuming the pool is the complete set of positives. Predicting unseen relations and learning novel patterns is penalized. The net effect is that researchers are discouraged from developing innovative approaches, in particular from applying machine learning, thereby slowing progress on the task.

Our second contribution, described in Section 4, addresses this bias through a new evaluation methodology, *on-demand evaluation*, which avoids pooling bias by querying crowdworkers, while minimizing cost by leveraging previous systems' predictions when possible. We then compute the new system's score based on the predictions of past systems using importance weighting. As more systems are evaluated, the marginal cost of evaluating a new system decreases. We show how the on-demand evaluation methodology can be applied to knowledge base population in Section 5. Through a simulated experiment on evaluation data released through the TAC KBP 2015 Slot Validation track, we show that we are able to obtain unbiased estimates of a new systems score's while significantly reducing variance.

Finally, our third contribution is an implementation of our framework as a publicly available evaluation service at `https://kbpo.stanford.edu`, where researchers can have their own KBP systems evaluated. The data collected through the evaluation process could even be valuable for relation extraction, entity linking and coreference, and will also be made publicly available through the website. We evaluate three systems on the 2016 TAC KBP corpus for about $150 each (a fraction of the cost of official evaluation). We believe the public availability of this service will speed the pace of progress in developing KBP systems.

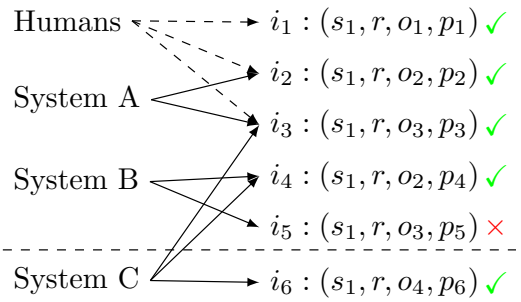


Figure 2: In pooled evaluation, an evaluation dataset is constructed by labeling relation instances collected from the pooled systems (A and B) and from a team of human annotators (Humans). However, when a new system (C) is evaluated on this dataset, some of its predictions (i_6) are missing and can not be fairly evaluated. Here, the precision and recall for C should be $\frac{3}{3}$ and $\frac{3}{4}$ respectively, but its evaluation scores are estimated to be $\frac{2}{3}$ and $\frac{2}{3}$. The discrepancy between these two scores is called *pooling bias*.

2 Background

In knowledge base population, each relation is a triple (SUBJECT, PREDICATE, OBJECT) where SUBJECT and OBJECT are some globally unique entity identifiers (e.g. Wikipedia page titles) and PREDICATE belongm to a specified schema.[1] A KBP system returns an output in the form of *relation instances* (SUBJECT, PREDICATE, OBJECT, PROVENANCE), where PROVENANCE is a description of where exactly in the document corpus the relation was found. In the example shown in Figure 1, CARRIE FISHER and DEBBIE REYNOLDS are identified as the subject and object, respectively, of the predicate CHILD OF, and the whole sentence is provided as provenance. The provenance also identifies that CARRIE FISHER is referenced by **Fisher** within the sentence. Note that the same relation can be expressed in multiple sentences across the document corpus; each of these is a different relation instance.

Pooled evaluation. The primary source of evaluation data for KBP comes from the annual TAC KBP competition organized by NIST (Ji et al.,

[1]The TAC KBP guidelines specify a total of 65 predicates (including inverses) such as `per:title` or `org:founded_on`, etc. Subject entities can be people, organizations, geopolitical entities, while object entities also include dates, numbers and arbitrary string-values like job titles.

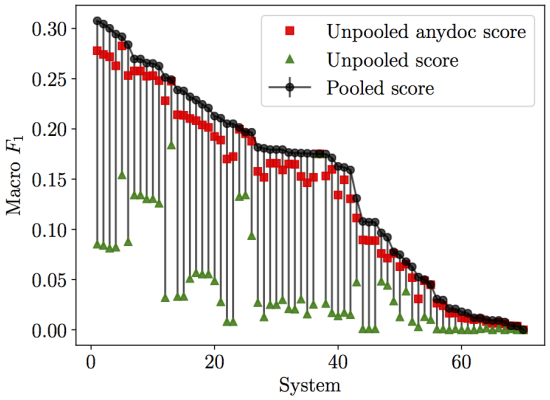
2011). Let E be a held-out set of *evaluation entities*. There are two steps performed in parallel: First, each participating system is run on the document corpus to produce a set of relation instances; those whose subjects are in E are labeled as either positive or negative by annotators. Second, a team of annotators identify and label correct relation instances for the evaluation entities E by manually searching the document corpus within a time budget (Ellis et al., 2012). These labeled relation instances from the two steps are combined and released as the evaluation dataset. In the example in Figure 2, systems A and B were used in constructing the pooling dataset, and there are 3 distinct relations in the dataset, between s_1 and o_1, o_2, o_3.

A system is evaluated on the precision of its predicted relation instances for the evaluation entities E and on the recall of the corresponding predicted *relations* (not instances) for the same entities (see Figure 2 for a worked example). When using the evaluation data during system development, it is common practice to use the more lenient `anydoc` score that ignores the provenance when checking if a relation instance is true. Under this metric, predicting the relation (CARRIE FISHER, CHILD OF, DEBBIE REYNOLDS) from an ambiguous provenance like "**Carrie Fisher** and **Debbie Reynolds** arrived together at the awards show" would be considered correct even though it would be marked wrong under the official metric.

3 Measuring pooling bias

The example in Figure 2 makes it apparent that pooling-based evaluation can introduce a systematic bias against unpooled systems. However, it has been assumed that the bias is insignificant in practice given the large number of systems pooled in the TAC KBP evaluation. We will now show that the assumption is not valid using data from the TAC KBP 2015 evaluation.[2]

Measuring bias. In total, there are 70 system submissions from 18 teams for 317 evaluation entities (E) and the evaluation set consists of 11,008 labeled relation instances.[3] The original evaluation dataset gives us a good measure of the true scores for the participating systems. Similar to Zobel (1998), which studied pooling bias in information retrieval, we simulate the condition of a team not being part of the pooling process by removing any predictions that are unique to its systems from the evaluation dataset. The pooling bias is then the difference between the true and unpooled scores.



	Median bias		
	Precision	Recall	Macro F_1
Official	17.93%	17.00%	15.51%
`anydoc`	2.34%	1.93%	2.05%

Figure 3: Median pooling bias (difference between pooled and unpooled scores) on the top 40 systems of TAC KBP 2015 evaluation using the official and `anydoc` scores. The bias is much smaller for the lenient `anydoc` metric, but even so, it is larger than the largest difference between adjacent systems (1.5% F_1) and typical system improvements (around 1% F_1).

Results. Figure 3 shows the results of measuring pooling bias on the TAC KBP 2015 evaluation on the F_1 metric using the official and `anydoc` scores.[4][5] We observe that even with lenient `anydoc` heuristic, the median bias (2.05% F_1) is much larger than largest difference between adjacently ranked systems (1.5% F_1). This experiment shows that pooling evaluation is significantly and systematically biased against systems that make novel predictions!

[2] Our results are not qualitatively different on data from previous years of the shared task.

[3] The evaluation set is actually constructed from compositional queries like, "what does Carrie Fisher's parents do?": these queries select relation instances that answer the question "who are Carrie Fisher's parents?", and then use those answers (e.g. "Debbie Reynolds") to select relation instances that answer "what does Debbie Reynolds do?". We only consider instances selected in the first part of this process.

[4] We note that `anydoc` scores are on average 0.88% F_1 larger than the official scores.

[5] The outlier at rank 36 corresponds to a University of Texas, Austin system that only filtered predictions from other systems and hence has no unique predictions itself.

4 On-demand evaluation with importance sampling

Pooling bias is fundamentally a sampling bias problem where relation instances from new systems are underrepresented in the evaluation dataset. We could of course sidestep the problem by exhaustively annotating the entire document corpus, by annotating all mentions of entities and checking relations between all pairs of mentions. However, that would be a laborious and prohibitively expensive task: using the interfaces we've developed (Section 6), it costs about $15 to annotate a single document by non-expert crowdworkers, resulting in an estimated cost of at least $1,350,000 for a reasonably large corpus of 90,000 documents (Dang, 2016). The annotation effort would cost significantly more with expert annotators. In contrast, *labeling* relation instances from system predictions can be an order of magnitude cheaper than finding them in documents: using our interfaces, it costs only about $0.18 to verify each relation instance compared to $1.60 per instance extracted through exhaustive annotations.

We propose a new paradigm called on-demand evaluation which takes a lazy approach to dataset construction by annotating predictions from systems *only when they are underrepresented*, thus correcting for pooling bias as it arises. In this section, we'll formalize the problem solved by on-demand evaluation independent of KBP and describe a cost-effective solution that allows us to accurately estimate evaluation scores without bias using importance sampling. We'll then instantiate the framework for KBP in Section 5.

4.1 Problem statement

Let $\mathcal{X}$ be the universe of (relation) instances, $\mathcal{Y} \subseteq \mathcal{X}$ be the unknown subset of correct instances, $X_1, \ldots X_m \subseteq \mathcal{X}$ be the predictions for m systems, and let $Y_i = X_i \cap \mathcal{Y}$. Let $X = \bigcup_{i=1}^m X_i$ and $Y = \bigcup_{i=1}^m Y_i$. Let $f(x) \stackrel{\text{def}}{=} \mathbb{I}[x \in \mathcal{Y}]$ and $g_i(x) = \mathbb{I}[x \in X_i]$, then the precision, π_i, and recall, r_i, of the set of predictions X_i is

$$\pi_i \stackrel{\text{def}}{=} \mathbb{E}_{x \sim p_i}[f(x)] \qquad r_i \stackrel{\text{def}}{=} \mathbb{E}_{x \sim p_0}[g_i(x)],$$

where p_i is a distribution over X_i and p_0 is a distribution over $\mathcal{Y}$. We assume that p_i is known, e.g. the uniform distribution over X_i and that we know p_0 up to normalization constant and can sample from it.

In on-demand evaluation, we can query $f(x)$ (e.g. labeling an instance) or draw a sample from p_0; typically, querying $f(x)$ is significantly cheaper than sampling from p_0. We obtain prediction sets $X_1, \ldots, X_m$ sequentially as the systems are submitted for evaluation. Our goal is to estimate π_i and r_i for each system $i = 1, \ldots, m$.

4.2 Simple estimators

We can estimate each π_i and r_i independently with simple Monte Carlo integration. Let $\hat{X}_1, \ldots, \hat{X}_m$ be multi-sets of $n_1, \ldots, n_j$ i.i.d. samples from $X_1, \ldots, X_m$ respectively, and let $\hat{Y}_0$ be a multi-set of n_0 samples drawn from $\mathcal{Y}$. Then, the simple estimators for precision and recall are:

$$\hat{\pi}_i^{(\text{simple})} = \frac{1}{n_i} \sum_{x \in \hat{X}_i} f(x) \quad \hat{r}_i^{(\text{simple})} = \frac{1}{n_0} \sum_{x \in \hat{Y}_0} g_i(x).$$

4.3 Joint estimators[6]

The simple estimators are unbiased but have wastefully large variance because evaluating a new system does not leverage labels acquired for previous systems.

On-demand evaluation with the joint estimator works as follows: First $\hat{Y}_0$ is randomly sampled from $\mathcal{Y}$ once when the evaluation framework is launched. For every new set of predictions X_m submitted for evaluation, the minimum number of samples n_m required to accurately evaluate X_m is calculated based on the current evaluation data, $\hat{Y}_0$ and $\hat{X}_1, \ldots, \hat{X}_{m-1}$. Then, the set $\hat{X}_m$ is added to the evaluation data by evaluating $f(x)$ on n_m samples drawn from X_m. Finally, estimates π_i and r_i are updated for each system $i = 1, \ldots, m$ using the joint estimators that will be defined next. In the rest of this section, we will answer the following three questions:

1. How can we use all the samples $\hat{X}_1, \ldots \hat{X}_m$ when estimating the precision π_i of system i?
2. How can we use all the samples $\hat{X}_1, \ldots, \hat{X}_m$ with $\hat{Y}_0$ when estimating recall r_i?
3. Finally, to form $\hat{X}_m$, how many samples should we draw from X_m given existing samples and $\hat{X}_1, \ldots, \hat{X}_{m-1}$ and $\hat{Y}_0$?

Estimating precision jointly. Intuitively, if two systems have very similar predictions X_i and X_j,

[6] Proofs for claims made in this section can be found in Appendix B of the supplementary material.

we should be able to use samples from one to estimate precision on the other. However, it might also be the case that X_i and X_j only overlap on a small region, in which case the samples from X_j do not accurately represent instances in X_i and could lead to a biased estimate. We address this problem by using importance sampling (Owen, 2013), a standard statistical technique for estimating properties of one distribution using samples from another distribution.

In importance sampling, if $\hat{X}_i$ is sampled from q_i, then $\frac{1}{n_i}\sum_{x\in\hat{X}_i}\frac{p_i(x)}{q_i(x)}f(x)$ is an unbiased estimate of π_i. We would like the proposal distribution q_i to both leverage samples from all m systems and be tailored towards system i. To this end, we first define a distribution over systems j, represented by probabilities w_{ij}. Then, define q_i as sampling a j and drawing $x \sim p_j$; formally $q_i(x) = \sum_{j=1}^{m} w_{ij}p_j(x)$.

We note that $q_i(x)$ not only significantly differs between systems, but also changes as new systems are added to the evaluation pool. Unfortunately, the standard importance sampling procedure requires us to draw and use samples from each distribution $q_i(x)$ independently and thus can not effectively reuse samples drawn from different distributions. To this end, we introduce a practical refinement to the importance sampling procedure: we independently draw n_j samples according to $p_j(x)$ from each of the m systems independently and then numerically integrate over these samples using the weights w_{ij} to "mix" them appropriately to produce and unbiased estimate of π_i while reducing variance. Formally, we define the *joint precision estimator*:

$$\hat{\pi}_i^{\text{(joint)}} \stackrel{\text{def}}{=} \sum_{j=1}^{m}\frac{w_{ij}}{n_j}\sum_{x\in\hat{X}_j}\frac{p_i(x)f(x)}{q_i(x)},$$

where each $\hat{X}_j$ consists of n_j i.i.d. samples drawn from p_j.

It is a hard problem to determine what the optimal mixing weights w_{ij} should be. However, we can formally verify that if X_i and X_j are disjoint, then $w_{ij} = 0$ minimizes the variance of π_i, and if $X_i = X_j$, then $w_{ij} \propto n_j$ is optimal. This motivates the following heuristic choice which interpolates between these two extremes: $w_{ij} \propto n_j \sum_{x\in\mathcal{X}} p_j(x)p_i(x)$.

Estimating recall jointly. The recall of system i can be expressed can be expressed as a product $r_i = \theta\nu_i$, where θ is the *recall of the pool*, which measures the fraction of all positive instances predicted by the pool (any system), and ν_i is the *pooled recall of system* i, which measures the fraction of the pool's positive instances predicted by system i. Letting $g(x) \stackrel{\text{def}}{=} \mathbb{I}[x \in X]$, we can define these as:

$$\nu_i \stackrel{\text{def}}{=} \mathbb{E}_{x\sim p_0}[g_i(x) \mid x \in X] \quad \theta \stackrel{\text{def}}{=} \mathbb{E}_{x\sim p_0}[g(x)].$$

We can estimate θ analogous to the simple recall estimator $\hat{r}_i$, except we use the pool g instead a system g_i. For ν_i, the key is to leverage the work from estimating precision. We already evaluated $f(x)$ on $\hat{X}_i$, so we can compute $\hat{Y}_i \stackrel{\text{def}}{=} \hat{X}_i \cap \mathcal{Y}$ and form the subset $\hat{Y} = \bigcup_{i=1}^{m}\hat{Y}_i$. $\hat{Y}$ is an approximation of $\mathcal{Y}$ whose bias we can correct through importance reweighting. We then define estimators as follows:

$$\hat{\nu}_i \stackrel{\text{def}}{=} \frac{\sum_{j=1}^{m}\frac{w_{ij}}{n_j}\sum_{x\in\hat{Y}_j}\frac{p_0(x)g_i(x)}{q_i(x)}}{\sum_{j=1}^{m}\frac{w_{ij}}{n_j}\sum_{x\in\hat{Y}_j}\frac{p_0(x)}{q_i(x)}}$$

$$\hat{r}_i^{\text{(joint)}} \stackrel{\text{def}}{=} \hat{\theta}\hat{\nu}_i \quad \hat{\theta} \stackrel{\text{def}}{=} \frac{1}{n_0}\sum_{x\in\hat{Y}_0} g(x).$$

where q_i and w_{ij} are the same as before.

Adaptively choosing the number of samples. Finally, a desired property for on-demand evaluation is to label new instances only when the current evaluation data is insufficient, e.g. when a new set of predictions X_m contains many instances not covered by other systems. We can measure how well the current evaluation set covers the predictions X_m by using a conservative estimate of the variance of $\hat{\pi}_m^{\text{(joint)}}$.[7] In particular, the variance of $\hat{\pi}_m^{\text{(joint)}}$ is a monotonically decreasing function in n_m, the number of samples drawn from X_m. We can easily solve for the minimum number of samples required to estimate $\hat{\pi}_m^{\text{(joint)}}$ within a confidence interval ϵ by using the bisection method (Burden and Faires, 1985).

5 On-demand evaluation for KBP

Applying the on-demand evaluation framework to a task requires us to answer three questions:

1. What is the desired distribution over system predictions p_i?

[7] Further details can be found in Appendix B of the supplementary material.

2. How do we label an instance x, i.e. check if $x \in \mathcal{Y}$?
3. How do we sample from the unknown set of true instances $x \sim p_0$?

In this section, we present practical implementations for knowledge base population.

5.1 Sampling from system predictions

Both the official TAC-KBP evaluation and the on-demand evaluation we propose use micro-averaged precision and recall as metrics. However, in the official evaluation, these metrics are computed over a fixed set of evaluation entities chosen by LDC annotators, resulting in two problems: (a) defining evaluation entities requires human intervention and (b) typically a large source of variability in evaluation scores comes from not having enough evaluation entities (see e.g. (Webber, 2010)). In our methodology, we replace manually chosen evaluation entities by sampling entities from each system's output according p_i. In effect, p_i makes explicit the decision process of the annotator who chooses evaluation entities.

Identifying a reasonable distribution p_i is an important implementation decision that depends on what one wishes to evaluate. Our goal for the on-demand evaluation service we have implemented is to ensure that KBP systems are fairly evaluated on diverse subjects and predicates, while at the same time, ensuring that entities with multiple relations are represented to measure completeness of knowledge base entries. As a result, we propose a distribution that is inversely proportional to the frequency of the subject and predicate and is proportional to the number of unique relations identified for an entity (to measure knowledge base completeness). See Appendix A in the supplementary material for an analysis of this distribution and a study of other potential choices.

5.2 Labeling predicted instances

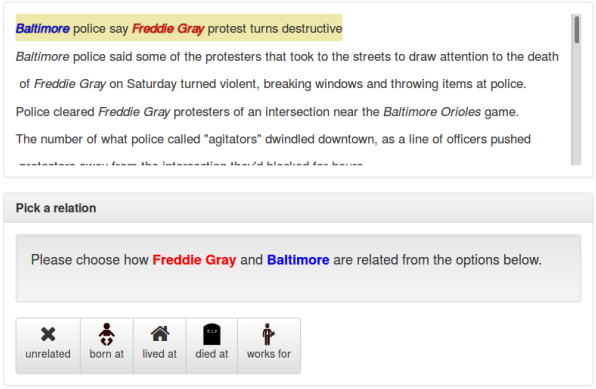
We label predicted relation instances by presenting the instance's provenance to crowdworkers and asking them to identify if a relation holds between the identified subject and object mentions (Figure 4a). Crowdworkers are also asked to link the subject and object mentions to their canonical mentions within the document and to pages on Wikipedia, if possible, for entity linking. On average, we find that crowdworkers are able to perform this task in about 20 seconds, corresponding to about \$0.05 per instance. We requested 5 crowdworkers to annotate a small set of 200 relation instances from the 2015 TAC-KBP corpus and measured a substantial inter-annotator agreement with a Fleiss' kappa of 0.61 with 3 crowdworkers and 0.62 with 5. Consequently, we take a majority vote over 3 workers in subsequent experiments.

5.3 Sampling true instances

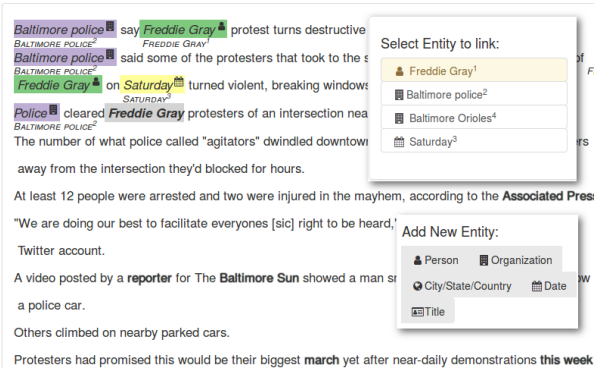
Sampling from the set of true instances $\mathcal{Y}$ is difficult because we can't even enumerate the elements of $\mathcal{Y}$. As a proxy, we assume that relations are identically distributed across documents and have crowdworkers annotate a random subset of documents for relations using an interface we developed (Figure 4b). Crowdworkers begin by identifying every mention span in a document. For each mention, they are asked to identify its type, canonical mention within the document and associated Wikipedia page if possible. They are then presented with a separate interface to label predicates between pairs of mentions within a sentence that were identified earlier.

We compare crowdsourced annotations against those of expert annotators using data from the TAC KBP 2015 EDL task on 10 randomly chosen documents. We find that 3 crowdworkers together identify 92% of the entity spans identified by expert annotators, while 7 crowdworkers together identify 96%. When using a token-level majority vote to identify entities, 3 crowdworkers identify about 78% of the entity spans; this number does not change significantly with additional crowdworkers. We also measure substantial token-level inter-annotator agreement using Fleiss' kappa for identifying typed mention spans ($\kappa = 0.83$), canonical mentions ($\kappa = 0.75$) and entity links ($\kappa = 0.75$) with just three workers. Based on this analysis, we use token-level majority over 3 workers in subsequent experiments.

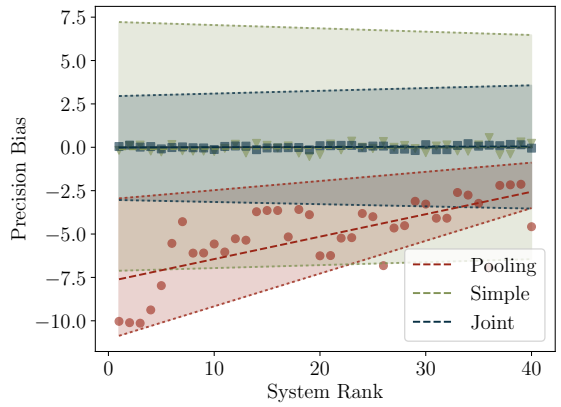
The entity annotation interface is far more involved and takes on average about 13 minutes per document, corresponding to about \$2.60 per document, while the relation annotation interface takes on average about \$2.25 per document. Because documents vary significantly in length and complexity, we set rewards for each document based on the number of tokens (.75c per token) and mention pairs (5c per pair) respectively. With 3 workers per document, we paid about \$15 per document on average. Each document contained an average



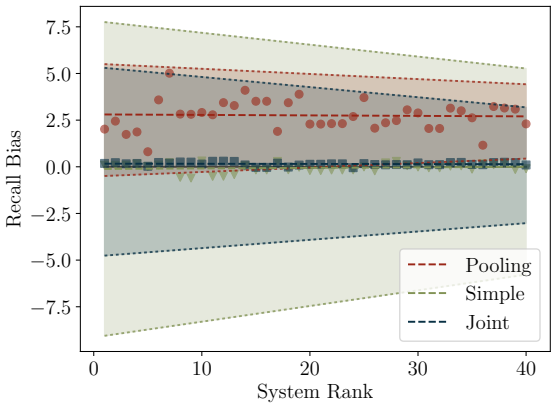
(a)



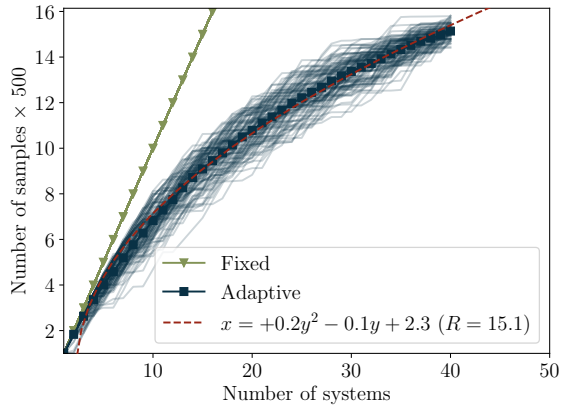
(b)



(c)



(d)



(e)

Sys.	P	R	F_1
	TAC KBP evaluation		
P	47.6%	11.0%	17.9%
P+L	35.5%	18.4%	24.2%
P+L+N	26.3%	27.0%	26.6%
	On-demand evaluation		
P	74.7%	5.8%	10.8%
P+L	54.7%	7.6%	13.3%
P+L+N	34.0%	9.8%	15.2%

(f)

Figure 4: **(a, b):** Interfaces for annotating relations and entities respectively. **(c, d):** A comparison of bias for the pooling, simple and joint estimators on the TAC KBP 2015 challenge. Each point in the figure is a mean of 500 repeated trials; dotted lines show the 90% quartile. Both the simple and joint estimators are unbiased, and the joint estimator is able to significantly reduce variance. **(e):** A comparison of the number of samples used to estimate scores under the fixed and adaptive sample selection scheme. Each faint line shows the number of samples used during a single trial, while solid lines show the mean over 100 trials. The dashed line shows a square-root relationship between the number of systems evaluated and the number of samples required. Thus joint estimation combined with adaptive sample selection can reduce the number of labeled annotations required by an order of magnitude. **(f):** Precision (P), recall (R) and F_1 scores from a pilot run of our evaluation service for ensembles of a rule-based system (R), a logistic classifier (L) and a neural network classifier (N) run on the TAC KBP 2016 document corpus.

9.2 relations, resulting in a cost of about $1.61 per relation instance. We note that this is about ten times as much as labeling a relation instance.

We defer details regarding how documents themselves should be weighted to capture diverse entities that span documents to Appendix A.

6 Evaluation

Let us now see how well on-demand evaluation works in practice. We begin by empirically studying the bias and variance of the joint estimator proposed in Section 4 and find it is able to correct for pooling bias while significantly reducing variance in comparison with the simple estimator. We then demonstrate that on-demand evaluation can serve as a practical replacement for the TAC KBP evaluations by piloting a new evaluation service we have developed to evaluate three distinct systems on TAC KBP 2016 document corpus.

6.1 Bias and variance of the on-demand evaluation.

Once again, we use the labeled system predictions from the TAC KBP 2015 evaluation and treat them as an exhaustively annotated dataset. To evaluate the pooling methodology we construct an evaluation dataset using instances found by human annotators and labeled instances pooled from 9 randomly chosen teams (i.e. half the total number of participating teams), and use this dataset to evaluate the remaining 9 teams. On average, the pooled evaluation dataset contains between 5,000 and 6,000 labeled instances and evaluates 34 different systems (since each team may have submitted multiple systems). Next, we evaluated sets of 9 randomly chosen teams with our proposed simple and joint estimators using a total of 5,000 samples: about 150 of these samples are drawn from $\mathcal{Y}$, i.e. the full TAC KBP 2015 evaluation data, and 150 samples from each of the systems being evaluated.

We repeat the above simulated experiment 500 times and compare the estimated precision and recall with their true values (Figure 4). The simulations once again highlights that the pooled methodology is biased, while the simple and joint estimators are not. Furthermore, the joint estimators significantly reduce variance relative to the simple estimators: the median 90% confidence intervals reduce from 0.14 to 0.06 precision and from 0.14 to 0.08 for recall.

6.2 Number of samples required by on-demand evaluation

Separately, we evaluate the efficacy of the adaptive sample selection method described in Section 4.3 through another simulated experiment. In each trial of this experiment, we evaluate the top 40 systems in random order. As each subsequent system is evaluated, the number of samples to pick from the system is chosen to meet a target variance and added to the current pool of labeled instances. To make the experiment more interpretable, we choose the target variance to correspond with the estimated variance of having 500 samples. Figure 4 plots the results of the experiment. The number of samples required to estimate systems quickly drops off from the benchmark of 500 samples as the pool of labeled instances covers more systems. This experiment shows that on-demand evaluation using joint estimation can scale up to an order of magnitude more submissions than a simple estimator for the same cost.

6.3 A mock evaluation for TAC KBP 2016

We have implemented the on-demand evaluation framework described here as an evaluation service to which researchers can submit their own system predictions. As a pilot of the service, we evaluated three relation extraction systems that also participated in the official 2016 TAC KBP competition. Each system uses Stanford CoreNLP (Manning et al., 2014) to identify entities, the Illinois Wikifier (Ratinov et al., 2011) to perform entity linking and a combination of a rule-based system (P), a logistic classifier (L), and a neural network classifier (N) for relation extraction. We used 15,000 Newswire documents from the 2016 TAC KBP evaluation as our document corpus. In total, 100 documents were exhaustively annotated for about $2,000 and 500 instances from each system were labeled for about $150 each. Evaluating all three system only took about 2 hours.

Figure 4f reports scores obtained through on-demand evaluation of these systems as well as their corresponding official TAC evaluation scores. While the relative ordering of systems between the two evaluations is the same, we note that precision and recall as measured through on-demand evaluation are respectively higher and lower than the official scores. This is to be expected because on-demand evaluation measures precision using each systems output as opposed

to an externally defined set of evaluation entities. Likewise, recall is measured using exhaustive annotations of relations within the corpus instead of annotations from pooled output in the official evaluation.

7 Related work

The subject of pooling bias has been extensively studied in the information retrieval (IR) community starting with Zobel (1998), which examined the effects of pooling bias on the TREC AdHoc task, but concluded that pooling bias was not a significant problem. However, when the topic was later revisited, Buckley et al. (2007) identified that the reason for the small bias was because the submissions to the task were too similar; upon repeating the experiment using a novel system as part of the TREC Robust track, they identified a 23% point drop in average precision scores![8]

Many solutions to the pooling bias problem have been proposed in the context of information retrieval, e.g. adaptively constructing the pool to collect relevant data more cost-effectively (Zobel, 1998; Cormack et al., 1998; Aslam et al., 2006), or modifying the scoring metrics to be less sensitive to unassessed data (Buckley and Voorhees, 2004; Sakai and Kando, 2008; Aslam et al., 2006). Many of these ideas exploit the ranking of documents in IR which does not apply to KBP. While both Aslam et al. (2006) and Yilmaz et al. (2008) estimate evaluation metrics by using importance sampling estimators, the techniques they propose require knowing the set of all submissions beforehand. In contrast, our on-demand methodology can produce unbiased evaluation scores for new development systems as well.

There have been several approaches taken to crowdsource data pertinent to knowledge base population (Vannella et al., 2014; Angeli et al., 2014; He et al., 2015; Liu et al., 2016). The most extensive annotation effort is probably Pavlick et al. (2016), which crowdsources a knowledge base for gun-violence related events. In contrast to previous work, our focus is on *evaluating systems*, not collecting a dataset. Furthermore, our main contribution is not a large dataset, but an evaluation service that allows anyone to use crowdsourcing predictions made by their system.

[8]For the interested reader, Webber (2010) presents an excellent survey of the literature on pooling bias.

8 Discussion

Over the last ten years of the TAC KBP task, the gap between human and system performance has barely narrowed despite the community's best efforts: top automated systems score less than 36% F_1 while human annotators score more than 60%. In this paper, we've shown that the current evaluation methodology may be a contributing factor because of its bias against novel system improvements. The new on-demand framework proposed in this work addresses this problem by obtaining human assessments of new system output through crowdsourcing. The framework is made economically feasible by carefully sampling output to be assessed and correcting for sample bias through importance sampling.

Of course, simply providing better evaluation scores is only part of the solution and it is clear that better datasets are also necessary. However, the very same difficulties in scale that make evaluating KBP difficult also make it hard to collect a high quality dataset for the task. As a result, existing datasets (Angeli et al., 2014; Adel et al., 2016) have relied on the output of existing systems, making it likely that they exhibit the same biases against novel systems that we've discussed in this paper. We believe that providing a fair and standardized evaluation platform as a service allows researchers to exploit such datasets and while still being able to accurately measure their performance on the knowledge base population task.

There are many other tasks in NLP that are even harder to evaluate than KBP. Existing evaluation metrics for tasks with a generation component—such as summarization or dialogue—leave much to be desired. We believe that adapting the ideas of this paper to those tasks is a fruitful direction, as progress of a research community is strongly tied to the fidelity of evaluation.

Acknowledgments

We would like to thank Yuhao Zhang, Hoa Deng, Eduard Hovy, and Jacob Steinhardt for discussions, William E. Webber for his excellent thesis that helped shape this project and the anonymous reviewers for their detailed and pertinent feedback. The first and second authors are supported under DARPA DEFT program under ARFL prime contract no. FA8750-13-2-0040.

References

H. Adel, B. Roth, and H. Schütze. 2016. Comparing convolutional neural networks to traditional models for slot filling. In *Human Language Technology and North American Association for Computational Linguistics (HLT/NAACL)*.

G. Angeli, J. Tibshirani, J. Y. Wu, and C. D. Manning. 2014. Combining distant and partial supervision for relation extraction. In *Empirical Methods in Natural Language Processing (EMNLP)*.

J. A. Aslam, V. Pavlu, and E. Yilmaz. 2006. A statistical method for system evaluation using incomplete judgments. In *ACM Special Interest Group on Information Retreival (SIGIR)*, pages 541–548.

J. Berant, A. Chou, R. Frostig, and P. Liang. 2013. Semantic parsing on Freebase from question-answer pairs. In *Empirical Methods in Natural Language Processing (EMNLP)*.

C. Buckley, D. Dimmick, I. Soboroff, and E. Voorhees. 2007. Bias and the limits of pooling for large collections. In *ACM Special Interest Group on Information Retreival (SIGIR)*.

C. Buckley and E. M. Voorhees. 2004. Retrieval evaluation with incomplete information. In *ACM Special Interest Group on Information Retreival (SIGIR)*, pages 25–32.

R. L. Burden and J. D. Faires. 1985. *Numerical Analysis (3rd ed.)*. PWS Publishers.

G. V. Cormack, C. R. Palmer, and C. L. A. Clarke. 1998. Efficient construction of large test collections. In *ACM Special Interest Group on Information Retreival (SIGIR)*.

H. T. Dang. 2016. Cold start knowledge base population at TAC KBP 2016. *Text Analytics Conference*.

J. Ellis, X. Li, K. Griffitt, and S. M. Strassel. 2012. Linguistic resources for 2012 knowledge base population evaluations. *Text Analytics Conference*.

A. Fader, L. Zettlemoyer, and O. Etzioni. 2014. Open question answering over curated and extracted knowledge bases. In *International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1156–1165.

S. Han, J. Bang, S. Ryu, and G. G. Lee. 2015. Exploiting knowledge base to generate responses for natural language dialog listening agents. *16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 129–133.

D. K. Harman. 1993. The first text retrieval conference (trec-1) rockville, md, u.s.a., 4-6 november, 1992. *Information Processing and Management*, 29:411–414.

L. He, M. Lewis, and L. Zettlemoyer. 2015. Question-answer driven semantic role labeling: Using natural language to annotate natural language. In *Empirical Methods in Natural Language Processing (EMNLP)*.

H. Ji, R. Grishman, and H. Trang Dang. 2011. Overview of the TAC 2011 knowledge base population track. In *Text Analytics Conference*.

K. S. Jones and C. V. Rijsbergen. 1975. Report on the need for and provision of an "ideal test collection. *Information Retrieval Test Collection*.

A. Kalyanpur, B. K. Boguraev, S. Patwardhan, J. W. Murdock, A. Lally, C. A. Welty, J. M. Prager, B. Coppola, A. Fokoue-Nkoutche, L. Zhang, Y. Pan, and Z. M. Qui. 2012. Structured data and inference in deepqa. *IBM Journal of Research and Development*, 56:351–364.

A. Liu, S. Soderland, J. Bragg, C. H. Lin, X. Ling, and D. S. Weld. 2016. Effective crowd annotation for relation extraction. In *North American Association for Computational Linguistics (NAACL)*, pages 897–906.

C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky. 2014. The stanford coreNLP natural language processing toolkit. In *ACL system demonstrations*.

A. B. Owen. 2013. *Monte Carlo theory, methods and examples*.

E. Pavlick, H. Ji, X. Pan, and C. Callison-Burch. 2016. The gun violence database: A new task and data set for NLP. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1018–1024.

L. Ratinov, D. Roth, D. Downey, and M. Anderson. 2011. Local and global algorithms for disambiguation to Wikipedia. In *Association for Computational Linguistics (ACL)*.

S. Reddy, M. Lapata, and M. Steedman. 2014. Large-scale semantic parsing without question-answer pairs. *Transactions of the Association for Computational Linguistics (TACL)*, 2(10):377–392.

T. Sakai and N. Kando. 2008. On information retrieval metrics designed for evaluation with incomplete relevance assessments. In *ACM Special Interest Group on Information Retreival (SIGIR)*, pages 447–470.

D. Vannella, D. Jurgens, D. Scarfini, D. Toscani, and R. Navigli. 2014. Validating and extending semantic knowledge bases using video games with a purpose. In *Association for Computational Linguistics (ACL)*, pages 1294–1304.

W. E. Webber. 2010. *Measurement in Information Retrieval Evaluation*. Ph.D. thesis, University of Melbourne.

E. Yilmaz, E. Kanoulas, and J. A. Aslam. 2008. A simple and efficient sampling method for estimating AP and NDCG. In *ACM Special Interest Group on Information Retreival (SIGIR)*, pages 603–610.

J. Zobel. 1998. How reliable are the results of large-scale information retrieval experiments? In *ACM Special Interest Group on Information Retreival (SIGIR)*.