On-demand open-world evaluation for knowledge base population

Anonymous EMNLP submission

Abstract

Large-scale information extraction systems for knowledge base population (KBP) predict true relations from an unenumerable set of candidate relations from a large document corpus. Due to the prohibitive cost of annotating the entire corpus, KBP evaluation has been traditionally closed-world, wherein only a subset of candidates are annotated, and candidates outside this subset are automatically marked negative. In the annual TAC-KBP challenge, the subset is constructed from the pooled predictions of previous systems. We show that this closed-world evaluation significantly penalizes new system improvements. To address this bias, we introduce a new on-demand evaluation in which a system's predictions are immediately evaluated through crowdsourcing, thus lazily simulating an open-world. With additional careful sampling and reweighting, we are able to produce scores at a fraction of the cost on a mock evaluation of the 2016 TAC-KBP challenge.

1 Introduction

Harnessing the wealth of information present in unstructured text online has been and continues to be 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, for example, "CARRIE FISHER'S MOTHER is DEBBIE REYNOLDS", from a document corpus. These knowledge bases have found many potential applications including question answering (Berant et al., 2013; Fader et al., 2014; Reddy et al., 2014), automated reasoning (Kalyanpur et al., 2012) and dialogue (Lee et al., 2015; Han et al., 2015).

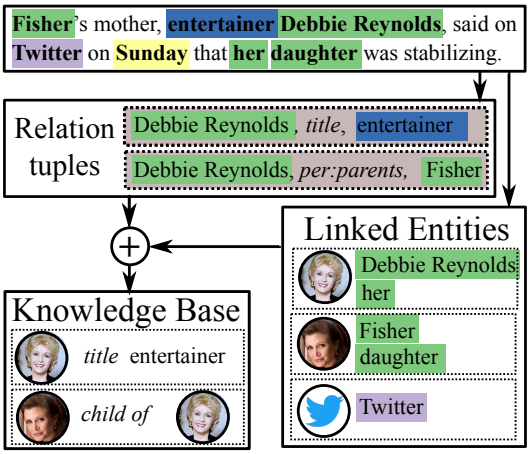


Figure 1: An example describing entities and relations in knowledge base population.

However, evaluating such large-scale information extraction systems remains a challenge: it is simply not economically feasible to exhaustively annotate every possible candidate relation from a sufficiently large corpus. As a result, the community has adopted a pooling-based methodology to construct datasets, similar to the Cranfield evaluation methodology in information retrieval. Under this approach, relations predicted by participating teams at the annual TAC-KBP competition are pooled together, labeled and released as a dataset for researchers to develop and evaluate their systems. However, if a newly developed system predicts a relation that wasn't seen by any of the competing teams, it can not be evaluated and considered to be wrong by default.

The key finding of this paper is that the pooling methodology results in a dataset that is significantly biased against novel systems: despite the fact that significant improvements increase scores

by less than 1% F_1, the F_1 scores of systems that did not participate in the pool are on average 2% points lower than they would be had they participated in the pool. Worse yet, if a system predicts qualitatively different relations that were systematically missed before, the stronger the bias against it will be, potentially even causing scores to drop. While designers of rule-based systems can still look to their intuition for guidance, those who rely on empirical tuning and machine learning are, at best, left in the dark, and more likely to be directed towards only predicting what others have predicted and no more.

Of course, relations predicted by systems participating in the competition are fairly evaluated and do not suffer from the studied bias, but this requires researchers to wait a year to get credible feedback on new ideas. These observations may explain why rule-based systems still play such a significant role in the top submissions at competition, and why, even after 8 years of running the competition, top automated systems achieve scores of only about 35% F_1 while human annotators score above 60% F_1 on the same task.

Instead, we propose a new evaluation methodology we call on-demand open-world evaluation. Under this paradigm, we continuously expand our evaluation corpus through crowdsourcing to actively correct for pooling bias. As researchers submit their new system's predictions to our evaluation platform,[1] we correct for bias in precision by carefully sampling from the relations predicted by newly submitted systems and for bias in recall by exhaustively annotating some documents. In doing so, the key challenges we address are reducing cost while decreasing variance and ensuring coverage of entities and relations.

We simulate our framework on evaluation data released through TAC-KBP Slot Validation tracks and show that we are able to obtain unbiased estimates while only labeling a fraction of the data. We also run a mock evaluation of three distinct systems on the 2016 TAC-KBP corpus and find that we are able to estimate precision and recall within 2% F_1 (half as much as the official scores) and within a budget of $2000 (a fraction of the cost of the official evaluation). We hope that the immediate, unbiased evaluation will help accelerate the development of better information extraction systems.

[1] An open-source implementation of the online evaluation platform is available for submissions at http://anonymo.us.

2 Setup

In knowledge base population, we are given a document corpus and must identify entities and relations mentioned to construct a knowledge base (Figure **??**). Each relation is a triple (SUBJECT, PREDICATE, OBJECT) where SUBJECT and OBJECT are some globally unique entity identifiers (e.g. Wikipedia page titles), PREDICATE is a relation belonging to a specified schema. A KBP system returns an output in the form of *relational tuples* (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 respectively identified as the subject and object of the predicate CHILD OF, and the whole sentence is provided as provenance. The provenance also includes information about entity linking by specifying that the subject CARRIE FISHER is referenced by **Fisher** within the sentence. Note that the same relation can be expressed in different parts of the document corpus; each of these instances is described by a unique relational tuple.

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

Pooled evaluation. The primary source of evaluation data for KBP comes from the annual TAC-KBP competition organized by NIST (). Each year, participating teams submit their predicted knowledge bases on a new document corpus, in the form the relational tuples described above. The organizers pool together relational tuples from all the participating systems and pick all tuples with subject entities that belong to a held-out set of evaluation entities to be assessed by trained annotators. The annotators judge whether or not the predicted relations are true given the specified provenance. A relational tuple for the relation (CARRIE FISHER, CHILD OF, DEBBIE REYNOLDS) that specifies an ambiguous provenance like "**Carrie Fisher** and **Debbie Reynolds**

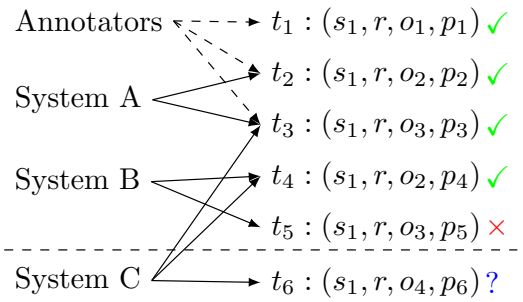


Figure 2: In pooled evaluation, a evaluation dataset is constructed by labeling the relational tuples collected from the pooled systems (A and B) and those identified by a team of human annotators (Annotators). However, when a new system (C) is evaluated on this dataset, some of its predictions may not be part of the dataset and these predictions can not be fairly evaluated, leading to a *pooling bias* that unfairly scores new systems.

arrived together at the awards show" is considered to be incorrect. Separately, a team of annotators identify relations for the same evaluation entities by manually searching the document corpus within a time budget. These two sets of labeled relational tuples 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 and o_3.

A system is evaluated by first selecting the predicted relational tuples for the evaluation entities and then measuring precision, recall and F_1 using the evaluation dataset. Precision is simply the fraction of predicted tuples that are correct: system A would get a precision of $\frac{2}{2}$ for predicting t_2 and t_3 while system B would get a precision of $\frac{1}{2}$ because t_5 specified an incorrect provenance. Recall is the fraction of true *relations* that the system identifies, irrespective of which provenance it uses as long as the provenance is correct. Thus, system A would get a recall of $\frac{2}{3}$ while system B would only get a recall of $\frac{1}{3}$. Finally, F_1 is the harmonic mean of precision and recall.

It is common practice within the field to use a more lenient scoring metric called the `anydoc` heuristic that ignores the provenance when checking if an relational tuple is true. Under this metric, t_5 would be considered to be correct and system B would get a precision and recall of $\frac{2}{2}$ and $\frac{2}{3}$ respectively.

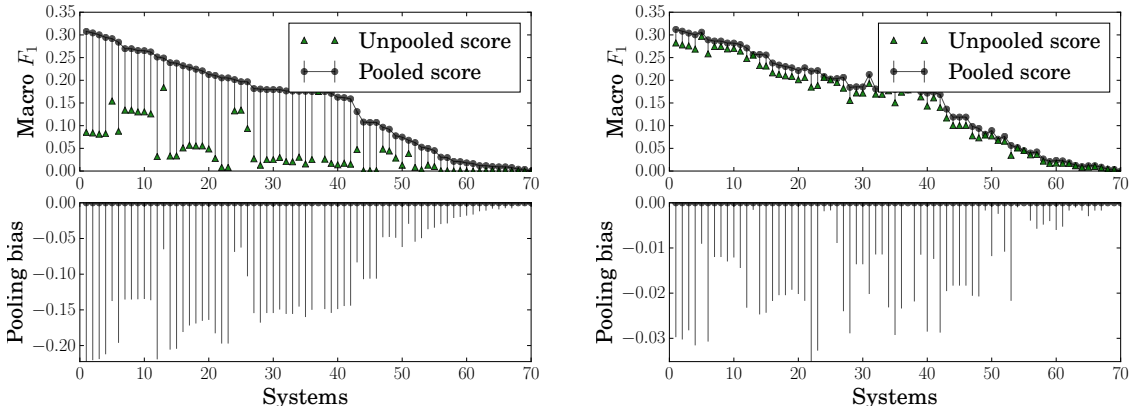
Pooling bias. As part of the competition, every relation predicted for the evaluation entities by the participating systems is labeled by annotators. However, when this dataset is used during system development, a new system (e.g. system C) may predict relational tuples that are not part of the dataset (e.g. t_6). These tuples can not be assessed without labeling the predictions and are considered to be wrong by default. As a result, system C would be evaluated to have a precision and recall of $\frac{2}{3}$ and $\frac{2}{3}$. On the other hand, if system C had participated in the pooling process, and t_6 were indeed correct, system C get higher precision and recall scores of $\frac{3}{3}$ and $\frac{3}{4}$ and would rank higher than system A.[2] The discrepancy between the scores of a system because it did not participate in the pooling process is called pooling bias.

3 Measuring pooling bias

While it is apparent from the example in Figure 2 that the pooling evaluation methodology could introduce bias, the argument for its use has been that this bias would be insignificant if enough systems were pooled. While this was the conclusion drawn during early studies of the pooling methodology in information retrieval (Zobel, 1998; Voorhees and Harman, 1999), more recent work has called the pooling argument into question (Buckley and Voorhees, 2004; Buckley et al., 2007). In this section, we'll measure pooling bias in KBP evaluation and show that the pooling argument does not presently hold.

Measuring bias. To measure pooling bias, we'll need to recreate the conditions shown in Figure 2: we must be able to measure evaluation metrics for a system before and after its inclusion to the pooled dataset. We simulate this condition using the "leave-out-unique" experiment proposed by Zobel (1998) to study pooling bias in information retrieval by using relational tuples from the participating systems in the TAC-KBP competition and their labels released as part of the evaluation dataset. While the original evaluation dataset contains labels for the relational tuples from all the participating systems, but we can remove any tuples that are unique to a particular system to simulate what the evaluation dataset would be if that system did not participate in the pooling process. Finally, we can then measure the difference

[2] Note that the recall scores of system A and B would also decrease because a new true relation had been found.



(a) Pooling bias for macro F_1 scores without the `anydoc` heuristic.

(b) Pooling bias for macro F_1 scores with the `anydoc` heuristic.

Figure 3: Pooling bias measures how much a system's evaluation score is changed by not having its predictions labeled as part of the pooled evaluation. Here, pooling bias is measured for the participating systems at the TAC-KBP 2015 competition. The median bias on the top 40 teams against an unpooled team is 15.51% points for the standard macro F_1 metric and 2.05% when evaluating using the `anydoc` heuristic. Note that the median difference between the `anydoc` scores and standard scores is 0.88% points; this difference is caused because the `anydoc` heuristic erroneously considers relational tuples with incorrect provenances to be correct.

in evaluation scores between the original evaluation dataset and the hypothetical one we've just created.

Results. Figure 3 shows the results of the leave-out-unique experiment on the TAC-KBP 2015 competition on the F_1 metric with and without the `anydoc` heuristic. In total, there are 70 system submissions from 18 teams for 317 query entities and the evaluation set consists of 11,008 assessed relational tuples.[3] The effect of pooling bias is understated on systems that performed very poorly. Consequently, we report median results on the top 40 systems (after which there is a sharp drop in performance). The outlier in the graph at rank 36 corresponds to a submission from University of Texas, Austin that only filtered predictions from other systems: this submission does not suffer any pooling bias because it has no unique predictions.

Without the `anydoc` heuristic, the median pooling bias is 15.51% F_1 points and with the heuristic it is 2.05% F_1 points. It should be noted that the `anydoc` heuristic also overestimates scores by about 0.88% F_1 points. The bias affects precision and recall almost equally: the median bias the without `anydoc` heuristic being 17.93% and 17.00% respectively, and the median bias the with `anydoc` heuristic being 2.34% and 1.93% respectively. At the same time, the median difference between two adjacent submissions is only 0.3%F_1 with largest difference being 1.5% F_1 (between the 26th and 27th ranked submissions).

We also considered an alternate scoring model of ignoring relational tuples outside the evaluation set during scoring and find that under this model F_1 scores remain biased by 14.99% and 0.73% points on average, with and without the `anydoc` heuristic respectively. While this scoring model is less biased than assuming unassessed relational tuples to be wrong, combined with the discrepancy of the `anydoc` score it remains too large a difference to be ignored.

The analysis confirms the necessity of the `anydoc` heuristic during development evaluations, but at the same time, even with the heuristic, the bias is much larger than the typical system difference. Thus, we conclude that this evaluation is biased against improvements that lead to novel predictions!

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

4 On-demand open-world evaluation

Pooling bias is essentially a sampling bias problem where relational tuples from pooled systems are overrepresented and those from new systems are underrepresented in the evaluation dataset. We could of course completely eliminate the bias by exhaustively annotating the entire document corpus, but that would be a laborious and prohibitively expensive task: using the interfaces we've developed (described in detail in Section 6), it costs about $20 to annotate a single document by non-expert crowdworkers, leading to an estimated cost of at least $200,000 for a reasonably large corpus of 10,000 documents. The annotation effort would cost significantly more with expert annotators.

In constrast, we propose a new paradigm called on-demand open-world 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 open world evaluation independent of KBP and describe our solution that allows us to accurately estimate evaluation metrics without bias in a cost-effective manner. We'll then instantiate the framework for KBP in Section 5.

4.1 Problem statement

Let $\mathcal{X}$ be a universe of possible system outputs (e.g. relational tuples), $\mathcal{Y} \subseteq \mathcal{X}$ be an unknown subset of this universe corresponding to the correct elements in $\mathcal{X}$, $X_1, \ldots X_m \subseteq \mathcal{X}$ be known subsets that correspond to the predicted output from m systems, and let $Y_1, \ldots, Y_m$ be the intersection of $X_1, \ldots, X_m$ with $\mathcal{Y}$. Our goal is estimate the precision, π_i, and recall, ρ_i, of the set of predictions X_i to within a certain confidence interval ϵ. Formally, let $f(x) \stackrel{\text{def}}{=} \mathbb{I}[x \in \mathcal{Y}]$ and $g_i(x) = \mathbb{I}[x \in X_i]$, then:

$$\pi_i \stackrel{\text{def}}{=} \mathbb{E}_{x \sim X_i}[f(x)] \qquad \rho_i \stackrel{\text{def}}{=} \mathbb{E}_{x \sim \mathcal{Y}}[g_i(x)],$$

where x is sampled from X_i and $\mathcal{Y}$ according to distributions $p_i(x)$ and $p'(x)$ respectively. We assume that $p_i(x)$ is known, e.g. the uniform distribution over X_i, and that samples from $p'(x)$ can be obtained, even if it is unknown.

In on-demand open-world evaluation, we are allowed to ask if $x \in \mathcal{Y}$ (e.g. by assessing a system's prediction) or for samples from $\mathcal{Y}$ (e.g. by exhaustively annotating a document) at a certain (monetary) cost. Typically, checking if $x \in \mathcal{Y}$ can be significantly cheaper than asking for samples from $\mathcal{Y}$.

Clearly, π_i and ρ_i can both be estimated by sampling from X_i and $\mathcal{Y}$ respectively. However, simple statistics tell us that we could require at least 10,000 samples each to estimate π and ρ to $\pm 1\%$, which would be quite costly on a per-system basis. To make the system practically viable, we'd like be able to reuse the samples we've collected and only spend money to annotate data when absolutely necessary. In the rest of this section, we'll see how to do this by answering the following three key questions:

1. Suppose we have evaluated $f(x)$ on samples $\hat{X}_1, \ldots, \hat{X}_m$ from $X_1, \ldots, X_m$ respectively. How should we best use all of these samples when estimating π_i?

2. Can we use the samples $\hat{X}_1, \ldots, \hat{X}_m$ when estimating ρ_i in conjunction with samples $\hat{Y}_0$ from $\mathcal{Y}$?

3. Finally, in practice, we only see the sets $X_1, \ldots, X_m$ sequentially, as and when they are submitted to the evaluation platform. How many samples should we draw from X_m, given existing samples $\hat{Y}_0$ and $\hat{X}_1, \ldots, \hat{X}_{m-1}$?

4.2 Amortizing costs when estimating precision

Intuitively, if a set X_j has a significant overlap with X_i, we expect that we should be able to its samples when estimating π_i. However, it might be case that X_j overlaps with X_i only when $p_i(x)$ is relatively small, in which case the sample $\hat{X}_j$ is not representative of X_i and a naive combination could lead to the wrong estimate of π_i. We address this problem by using importance sampling (Owen, 2013).

In particular, the estimator that we propose is:

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

where $n_j = |X_j|$, $q_i(x) = \sum_{j=1}^{m} w_{ij} p_j(x)$ and $w_{ij} \geq 0$ are mixture parameters such that $\sum_{j=1}^{m} w_{ij} = 1$ and $q_i(x) > 0$ wherever $p_i(x) > 0$.

This last condition is easy to guarantee by setting $w_{ii} > 0$.

In Appendix B we prove that $\hat{\pi}_i$ is an unbiased estimator of π_i and also work out an expression for its variance. The variance of course depends on $f(x)$, but the general intuition is that $\hat{\pi}_i$ will have high variance if $q_i(x) \ll p_i(x)$. This motivates the choice $w_{ij} \propto n_j \sum_{x \in \mathcal{X}} p_j(x) p_i(x)$, which assigns 0 weight to any set X_j that has no overlap with X_i and also has the optimal choice of weights if all the sets are identical, i.e. $p_j = p_i$ for all j.

On simulated experiments on the TAC-KBP dataset, we find a 4-fold decrease in variance using the proposed $\hat{\pi}$ when compared to estimating π_i solely using $\hat{X}_i$.

4.3 Amortizing costs when estimating recall

When estimating recall, we ideally would like to compare the performance of the system on samples drawn from $\mathcal{Y}$. Unfortunately, in practice, it is typically much harder to sample $\mathcal{Y}$ than it is to evaluate $f(x)$, because the former needs us to exhaustively annotate a document. In contrast, it is very easy to compare a system's recall relative to *other* systems by using the samples $\hat{X}_i$ we've already collected. This mode of computing recall, called pooled recall, can be biased and typically overestimates recall on the universe of relations. However, we know that if a system represents only a fraction ν_i of the pool of all systems and that the pool represents a fraction θ of $\mathcal{Y}$, ρ_i must equal $\theta\nu_i$.

With this in mind, we use a smaller sample $\hat{Y}_0$ from $\mathcal{Y}$ to estimate θ and then use the rest of $\hat{X}$ to estimate ν_i. In Appendix B we show that the final estimator, $\hat{\rho}_i \stackrel{\text{def}}{=} \hat{\theta}\hat{\nu}_i$, is unbiased and has a variance of,

$$\sigma_\rho^2 = \theta\sigma_\nu^2 + \nu_i\sigma_\theta^2 + \sigma_\nu^2\sigma_\theta^2.$$

With sufficient samples, σ_ν^2 can be made quite small, so that the leading term in the variance is $\nu_i\sigma_\theta^2$ and because typically $\nu_i \ll 1$, we expect the variance of our proposed estimator to be less than that of the estimator constructed by using $\hat{Y}_0$ alone.

On simulated experiments on the TAC-KBP dataset, we find a 2-fold decrease in variance using the proposed estimator $\hat{\rho}$ when compared to estimating ρ_i solely using $\hat{Y}_0$.

4.4 Adaptively drawing samples for new sets

Finally, the desired property for our framework is to annotate new data only when necessary, i.e. a new submission X_m contains sufficiently diverse output. We formalize this statement by requesting for a target variance ϵ. The variance of $\hat{\pi}_m$ is a complex non-convex function, but we know that it is monotonically decreasing in n_m, the number of samples drawn from the new output, X_m. Consequently it is quite easy to solve for the number of samples needed to achieve a target variance of ϵ using a bisection method. Simulated experiment of how many samples are required.

Algorithm 1 The on-demand open-world evaluation methodology

Require: A sequence of predicted output sets $X_1, \ldots X_m \subseteq \mathcal{X}$, a method of evaluating $f(x) = \mathbb{I}[x \in \mathcal{Y}]$ and a method of sampling $x \sim \mathcal{Y}$, desired confidence intervals ϵ_π and ϵ_ρ.
Ensure: Unbiased predictions of precision, $\hat{\pi}_1, \ldots, \hat{\pi}_m$ and recall $\hat{\rho}_1, \ldots \hat{\rho}_m$.
Sample a set $\hat{Y}_0$ from $\mathcal{Y}$ based on ϵ_ρ.
for $i = 0$ **to** $i = m$ **do**
  Compute the minimum number of samples n_i required to estimate π_i within ϵ_π.
  Evaluate $f(x)$ on n_i samples drawn from X_i.
  for $j = 0$ **to** $j = i$ **do**
    Use $\hat{X}_1, \ldots \hat{X}_i$ to update estimates for π_j and ρ_j.
  end for
end for

Algorithm 1 summarizes the whole approach.

5 On-demand open-world evaluation for KBP

Applying the on-demand open-world evaluation framework needs us to define 3 operations:

1. Given some system output X_i, how should we sample its elements, i.e. what is $p_i(x)$?
2. How do we verify system output, i.e. check if $x \in \mathcal{Y}$?
3. How do we sample from the unknown set of true instances $x \sim \mathcal{Y}$?

In this section, we'll see how each of these operations can be practically implemented for knowledge base population.

5.1 Sampling from system predictions

Recall that a KBP system predicts relational tuples of the form (SUBJECT, PREDICATE, OBJECT, PROVENANCE). The choice of distribution over system predictions, $p_i(x)$, allows us to calibrate our precision metric to increase the representation of rare predicates or subject entities in our precision score. This can desirable because, in practice, it is common for a system's predictions to be be dominated by a few common predicates (e.g. professional titles) or subject entities (like "the United States of America"), skewing our evaluation metrics towards a particular relation or entity type.

The traditional solution to this problem is to take the macro average of the metric over predicates or subject entities. We can replicate this behavior with the appropriate choice of a sampling distribution. To estimate the macro-average over predicates, we use a predicate-centric distribution $p_i^{(\text{p})}$ that first uniformly samples a predicate and then uniformly samples an instances with that predicate. Similarly, to estimate the macro-average over subject entities, we use an entity-centric distribution $p_i^{(\text{e})}$ that first uniformly samples over the entities in the system's predicted relational tuples and then uniformly samples instances with that entity as its subject. Figure **??** plots a histogram of different predicates and entities are represented by the uniform, predicate and subject distributions.

5.2 Verifying system predictions

When verifying relational tuples predicted by a system, crowd workers are presented the tuple's provenance and are asked to identify if a relation holds between the identified subject and object mentions (Figure 4a). Crowd workers also link the subject and object mentions to pages on Wikipedia, if possible. On average, we find that crowdworkers are able to perform this task in about 30 seconds, corresponding to about \$0.10 per instance. We requested 5 crowdworkers to annotate a small set of 200 relation instances from the 2015 TAC-KBP corpus and measured an inter-annotator agreement of 0.90 with 3 crowdworkers and 0.95 with 5. Consequently, we take a majority vote over 3 workers in subsequent experiments.

5.3 Sampling true instances

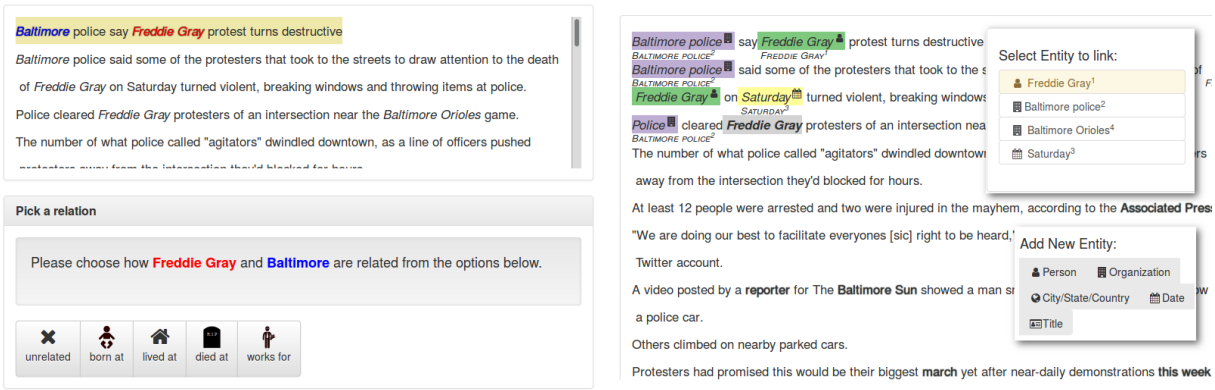
Sampling from the set of true instances $\mathcal{Y}$ is difficult because we can not even enumerate the elements of $\mathcal{Y}$. As a proxy, we assume that relations are identically distributed across documents and have crowd workers annotate a random subset of documents for relations.

To do so, crowd workers begin by identifying every mention span in a document and specifying its type. For each mention, they are also asked to identify the canonical mention within the document and identify links to Wikipedia pages where possible (Figure 4b). Finally, using a separate interface, crowdworkers annotate relations between pair of mentions within a sentence.

These interfaces are far more involved: the entity annotation interface 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.

We compare crowdsourced annotations against those of expert annotators using data from the TAC-KBP 2015 EDL task on 100 documents (). Each document was annotated by at least 7 crowdworkers. We find that 3 crowdworkers are together identify 92% of the entity spans identified by expert annotators, and 7 crowdworkers together identify 96%. When using a token-level majority vote to identify entities, 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 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.

A final issue to discuss is how documents themselves should be sampled to capture diverse entities that span documents. When considering uniformly sampled documents, we found that a majority of the relations extracted correspond to very rare entities and result in very few entities with more than one relation Figure **??**. In contrast, the TAC-KBP query are almost evenly split between rare and semi-frequent entities. As a heuristic, we adopt the following two-stage sampling procedure: First, 20% of our exhaustive document collection is sampled uniformly and annotated. We



(a) Relation extraction.

(b) Entity detection and linking.

Figure 4: Screenshots of the annotation interfaces.

then uniformly sample the entities annotated to create a collection of "query entities". Finally, we construct the remaining 80% of our document collection by searching for documents that contain the query entities according to an exact string match. This process results in far more entities of medium frequency.

6 Evaluation

6.1 Mock evaluation

Let us now see how well our new evaluation framework works in practice.

We conduct a mock evaluation using 15,000 newswire documents from 2016 TAC-KBP competition. We compare the predictions made by three distinct relation extraction systems: a rule-based system, a supervised system and a neural network classifier. Each system uses Stanford CoreNLP () to identify entities and the Illinois Wikifier () to perform entity linking.

In total, 100 documents were exhaustively annotated for about $2,000, and 1000 of each systems submissions were annotated at about $300 each. Table 1 presents the results of these systems on the mock evaluation. Two immediate takeaways are that the precisions of these systems are on par with their precisions in the original 2015 evaluation but the 95% confidence interval is almost a third as large. The recall scores on this evaluation are a bit smaller than on the 2015 evaluation and again the 95% confidence window is significantly smaller. Combining annotation pooling annotations noticeably reduces the variance over the uncombined estimation, while combining the unassessed output makes a smaller impact.

7 Related Work

The subject of pooling bias has been extensively studied in the information retrieval 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 for the TREC AdHoc tasks, as did a later study (Voorhees and Harman, 1999). However, when the topic was revisited after several years, Buckley and Voorhees (2004) found that pooling bias could significantly change system ranking and Buckley et al. (2007) identified that the reason for the small measured bias was because the submissions to the task were very similar; on repeating the experiment using a novel system as part of the TREC Robust track, they identified a 23% point drop in AP scores![4] We adapt the leave-one-out methodology of Zobel (1998) to measure pooling bias in KBP, however, unlike in the information retrieval setting, we find a very strong effect. One explanation for this difference is that the popular information retrieval metrics are rank-weighted, and unassessed documents typically tend to be lower in the ranking and hence contribute less to the evaluation score.

Likewise, many solutions to the pooling bias problem have been proposed in the context of information retrieval, from changing the queries to be more specific (Buckley et al., 2007), 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

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

Scheme	System	$P^e(\pm 95\%)$	$R^e(\pm 95\%)$	$F_1{}^e(\pm 95\%)$
Uncombined	Patterns	$80.4 \pm 3.0\%$	$10.4 \pm 5.0\%$	$18.41 \pm 4.3\%$
	Supervised	$60.4 \pm 3.0\%$	$15.4 \pm 5.0\%$	$24.54 \pm 4.3\%$
	Neural	$20.4 \pm 3.0\%$	$30.4 \pm 5.0\%$	$24.41 \pm 4.3\%$
+ Pooling	Patterns	$80.4 \pm 3.0\%$	$10.4 \pm 3.0\%$	$18.41 \pm 3.0\%$
	Supervised	$60.4 \pm 3.0\%$	$15.4 \pm 3.0\%$	$24.54 \pm 3.0\%$
	Neural	$20.4 \pm 2.6\%$	$30.4 \pm 2.7\%$	$24.41 \pm 2.6\%$

Table 1: Results from a mock evaluation.

Kando, 2008; Aslam et al., 2006). Many of these ideas exploit rank-weighted metrics and the rankings reported by systems, neither of which apply in the KBP setting. Furthermore, the pooling bias persists in KBP evaluations despite the fact that *all answers* reported by systems for a given set of entities are assessed for correctness.While both Aslam et al. (2006) and Yilmaz et al. (2008) estimate evaluation metrics by carefully sampling system output for assessment and using importance reweighing to correct for sampling bias, 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 too.

Crowdsourcing has become common place in the NLP community and there has been prior work in using crowdsourcing for semantic role labeling (He et al., 2015), building semantic ontologies (Vannella et al., 2014) and building a knowledge base for gun-violence related events (Pavlick et al., 2016). The main focus of our work is on *evaluating systems*, not necessarily collecting an exhaustive dataset. As a result, we are able to integrate the systems we are evaluating into our data collection process.

8 Discussion

Over the last ten years of the TAC-KBP competition, the gap between human and system performance has barely narrowed, despite the community's best efforts. In this paper, we've shown that the existing evaluation methodology may be a contributing factor because of its bias against novel system improvements. The new on-demand open-world framework proposed in this work addresses this problem by obtaining human assessments of novel, unseen, system output through crowdsourcing. The framework is made economically feasible by carefully sampling output to be assessed and correcting for sample bias through importance reweighing.

Of course, simply providing a higher fidelity evaluation signal 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[5] allows researchers to incorporate such datasets and while still being able to accurately measure their performance on the knowledge base population task.

Finally, despite the challenges in evaluating knowledge base population we've described in this work, there are many other tasks in NLP that are even harder to evaluate. In particular, we think that the community would be greatly aided by a better evaluation methodology for generation tasks like summarization or dialog. We believe the sampling ideas presented in this work will still be relevant in such a setting, but the challenge to tackle is in being able to reuse assessments obtained on one summary for another.

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

[5]We plan to host this platform publicly at `http://anonymo.us`

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.

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)*.

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.

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)*.

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.

K. Lee, P. H. Seo, J. Choi, S. Koo, and G. G. Lee. 2015. Conversational knowledge teaching agent that uses a knowledge base. *16th Annual Meeting of the Special Interest Group on Discourse and Dialogue* pages 139–143.

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.

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.

E. M. Voorhees and D. Harman. 1999. Overview of the eight text retreival conference (TREC-8). In *TREC-8*.

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)*.

A Annotation interfaces

B Theoretical proofs for the sampling procedures

Let's recall the notation from Section 4.

Let $\mathcal{X}$ be a universe of possible outputs (e.g. relation instances), $\mathcal{Y} \subseteq \mathcal{X}$ be an unknown subset of this universe corresponding to the correct elements in $\mathcal{X}$ and $X_1, \ldots X_m \subseteq \mathcal{X}$ be known subsets that correspond to the predicted output from m systems, and $Y_1, \ldots, Y_m$ be the intersection of $X_1, \ldots, X_m$ with $\mathcal{Y}$. Furthermore, let $\hat{X}_i$ be a set of n_i independent samples drawn from X_i with the distribution p_i, $\hat{Y}_i$ be the intersection of these sets with $\mathcal{Y}$, and $\hat{Y}_0$ be a sample drawn from $\mathcal{Y}$ according to an unknown distribution $p'(x)$.

We would like to evaluate precision, π_i, and recall, ρ_i:

$$\pi_i \stackrel{\text{def}}{=} \mathbb{E}_{x \sim X_i}[f(x)] \qquad\qquad \rho_i \stackrel{\text{def}}{=} \mathbb{E}_{x \sim \mathcal{Y}}[g_i(x)],$$

In this section, we'll provide proofs that show that the estimators proposed are indeed unbiased, and we will characterize their variance.

B.1 Estimating precision

In Section 4, we proposed the following estimator for π_i:

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

where $q_i(x) = \sum_{j=1}^{m} w_{ij} p_j(x)$ and $w_{ij} \geq 0$ are mixture parameters such that $\sum_{j=1}^{m} w_{ij} = 1$ and $q_i(x) > 0$ wherever $p_i(x) > 0$.

Theorem 1 (Statistical properties of $\hat{\pi}_i$)**.** *$\hat{\pi}_i$ is an unbiased estimator of π_i and has a variance of:*

$$\operatorname{Var} \hat{\pi}_i = \sum_{j=1}^{m} \frac{w_j^2}{n_j} \mathbb{E}_{p_j} \left[\frac{p_i(x)^2 f(x)^2 - \pi_{ij} p_i(x) f(x) q_i(x)}{q_i(x)^2} \right],$$

where $\pi_{ij} \stackrel{\text{def}}{=} \mathbb{E}_{p_j} \left[\frac{p_i(x) f(x)}{q_i(x)} \right]$.

Proof. Let $\hat{X} = (\hat{X}_1, \ldots, \hat{X}_m)$ which is drawn from the product distribution of $p_1 \times p_m$. By independence and the linearity of expectation,

$$\mathbb{E}_{\hat{X}} \left[\sum_{j=1}^{m} f(\hat{X}_j) \right] = \sum_{j=1}^{m} \mathbb{E}_{\hat{X}_j}[f(\hat{X}_j)].$$

First, let's show that $\hat{\pi}_i$ is unbiased:

$$\begin{aligned}
\mathbb{E}_{\hat{X}}[\hat{\pi}_i] &= \mathbb{E}_{\hat{X}}\left[\sum_{j=1}^{m}\frac{w_j}{n_j}\sum_{x\in\hat{X}_j}\frac{p_i(x)f(x)}{q_i(x)}\right] \\
&= \sum_{j=1}^{m}\frac{w_j}{n_j}\mathbb{E}_{\hat{X}_j}\left[\sum_{x\in\hat{X}_j}\frac{p_i(x)f(x)}{q_i(x)}\right] \\
&= \sum_{j=1}^{m}\frac{w_j}{n_j}n_j\mathbb{E}_{p_j}\left[\frac{p_i(x)f(x)}{q_i(x)}\right] \\
&= \sum_{j=1}^{m}w_j\sum_{x\in\mathcal{X}}p_j(x)\frac{p_i(x)f(x)}{q_i(x)} \\
&= \sum_{x\in\mathcal{X}}\sum_{j=1}^{m}w_jp_j(x)\frac{p_i(x)f(x)}{q_i(x)} \\
&= \sum_{x\in\mathcal{X}}q_i(x)\frac{p_i(x)f(x)}{q_i(x)} \\
&= \sum_{x\in\mathcal{X}}p_i(x)f(x) \\
&= \pi_i.
\end{aligned}$$

Now let's compute the variance.

$$\begin{aligned}
\operatorname{Var}\hat{\pi}_i &= \sum_{j=1}^{m}\frac{w_j^2}{n_j}\mathbb{E}_{p_j}\left[\frac{p_i(x)^2f(x)^2}{q_i(x)^2}\right] - \sum_{j=1}^{m}\frac{w_j^2}{n_j}\mathbb{E}_{p_j}\left[\frac{p_i(x)f(x)}{q_i(x)}\right]^2 \\
\operatorname{Var}\hat{\pi}_i &= \sum_{j=1}^{m}\frac{w_j^2}{n_j}\mathbb{E}_{p_j}\left[\frac{p_i(x)^2f(x)^2}{q_i(x)^2} - \frac{\pi_{ij}p_i(x)f(x)}{q_i(x)}\right] \\
\operatorname{Var}\hat{\pi}_i &= \sum_{j=1}^{m}\frac{w_j^2}{n_j}\mathbb{E}_{p_j}\left[\frac{p_i(x)^2f(x)^2 - \pi_{ij}p_i(x)f(x)q_i(x)}{q_i(x)^2}\right],
\end{aligned}$$

where $\pi_{ij} \stackrel{\text{def}}{=} \mathbb{E}_{p_j}\left[\frac{p_i(x)f(x)}{q_i(x)}\right]$. □

B.2 Estimating recall

In Section 4, we used the fact that the recall of system i, ρ_i, can be expressed as the recall of i within the pool, ν_i and the recall of the pool itself θ: $\rho_i = \theta\nu_i$:

$$\nu_i = \mathbb{E}_{x\sim\mathcal{Y}|Y}[g_i(x)] \qquad\qquad \theta = \mathbb{E}_{x\sim\mathcal{Y}}[g(x)],$$

where x is sampled under the distribution $p'(x \mid x \in Y)$ and $p'(x)$ respectively and $g(x) \stackrel{\text{def}}{=} \mathbb{I}[x \in \bigcup_{i=1}^{m} X_i] = \max_{j\in[1,m]} g_j(x)$ is the indicator function for x belonging to the pool.

Ideally, to estimate the pooled recall, ν_i, we need to take expectations with respect to $x \sim Y$, but have samples drawn from individual X_i. To correct for this bias, we'll use a self-normalizing estimator for ν_i:

$$\hat{\nu}_i \stackrel{\text{def}}{=} \frac{\sum_{j=1}^{m}\frac{w_j}{n_j}\sum_{x\in\hat{Y}_j}\frac{u(x)g_i(x)}{q(x)}}{\sum_{j=1}^{m}\frac{w_j}{n_j}\sum_{x\in\hat{Y}_j}\frac{u(x)}{q(x)}},$$

where $p'(x) \propto u(x)$, $q(x) = \sum_{j=1}^{m} w_j p_j(x)$ and $w_j \geq 0$ are mixture parameters such that $\sum_{j=1}^{m} w_j = 1$.

The pool recall θ can be estimated as follows:

$$\hat{\theta} \stackrel{\text{def}}{=} \sum_{x \in \hat{Y}_0} g(x),$$

where $g(x) \stackrel{\text{def}}{=} \mathbb{I}\left[x \in \bigcup_{i=1}^{m} X_i\right] = \max_{j \in [1,m]} g_j(x)$.

Finally, we proposed the following estimator for recall ρ_i:

$$\hat{\rho}_i \stackrel{\text{def}}{=} \hat{\theta}\hat{\nu}_i.$$

Let's start by showing that ν_i is unbiased.

Theorem 2 (Statistical properties of $\hat{\nu}_i$)**.** *$\hat{\nu}_i$ is a consistent estimator of ν_i.*

Proof. We have that $p'_Y(x) = \frac{w(x)}{Z_Y}$. While we do not know the value of Z_Y, we can divide both the numerator and denominator of $\hat{\nu}_i$ by this quantity:

$$\begin{aligned}\hat{\nu}_i &= \frac{\sum_{j=1}^{m} \frac{w_j}{n_j} \sum_{x \in \hat{Y}_j} \frac{u(x) g_i(x)}{Z_Y q(x)}}{\sum_{j=1}^{m} \frac{w_j}{n_j} \sum_{x \in \hat{Y}_j} \frac{u(x)}{Z_Y q(x)}} \\ &= \frac{\sum_{j=1}^{m} \frac{w_j}{n_j} \sum_{x \in \hat{Y}_j} \frac{p'_Y(x) g_i(x)}{q(x)}}{\sum_{j=1}^{m} \frac{w_j}{n_j} \sum_{x \in \hat{Y}_j} \frac{p'_Y(x)}{q(x)}}.\end{aligned}$$

As the number of samples $n_i \to \infty$,

$$\begin{aligned}\mathbb{E}_X[\hat{\nu}_i] &= \mathbb{E}_X\left[\frac{\sum_{j=1}^{m} \frac{w_j}{n_j} \sum_{x \in \hat{Y}_j} \frac{p'_Y(x) g_i(x)}{q(x)}}{\sum_{j=1}^{m} \frac{w_j}{n_j} \sum_{x \in \hat{Y}_j} \frac{p'_Y(x)}{q(x)}}\right] \\ &= \frac{\mathbb{E}_X\left[\sum_{j=1}^{m} \frac{w_j}{n_j} \sum_{x \in \hat{Y}_j} \frac{p'_Y(x) g_i(x)}{q(x)}\right]}{\mathbb{E}_X\left[\sum_{j=1}^{m} \frac{w_j}{n_j} \sum_{x \in \hat{Y}_j} \frac{p'_Y(x)}{q(x)}\right]}.\end{aligned}$$

Following similar arguments as in the proof of Theorem 1, the numerator and denominator are unbiased estimators of $\mathbb{E}_{x \sim \mathcal{Y}|Y}[g_i(x)]$ and $\mathbb{E}_{x \sim \mathcal{Y}|Y}[1] = 1$ respectively. Thus,

$$\mathbb{E}_X[\hat{\nu}_i] = \mathbb{E}_{x \sim \mathcal{Y}|Y}[g_i(x)]. \qquad = \nu_i.$$

$\hat{\nu}_i$ is an unbiased estimator of ν_i.

□

Finally, we turn to studying $\hat{\rho}$:

Theorem 3 (Statistical properties of $\hat{\rho}_i$)**.** *$\hat{\rho}_i$ is an unbiased estimator of ρ_i with variance*

$$\operatorname{Var} \hat{\rho}_i = \theta \operatorname{Var} \hat{\nu}_i + \nu_i \operatorname{Var} \hat{\theta} + \operatorname{Var} \hat{\theta} \operatorname{Var} \hat{\nu}_i.$$

Proof. First, let's show that $\rho_i = \theta\nu_i$:

$$\begin{aligned}\rho_i &\stackrel{\text{def}}{=} \mathbb{E}_{x \sim \mathcal{Y}}[g_i(x)] \\ &= p'(Y_i) \\ &= p'(Y \wedge Y_i) \\ &= p'(Y) p'(Y_i \mid Y) \\ &= \mathbb{E}_{x \sim \mathcal{Y}}[g(x)] \mathbb{E}_{x \sim \mathcal{Y}|Y}[g_i(x)] \\ &= \theta\nu_i.\end{aligned}$$

From Theorem 2, we have that $\hat{\nu}_i$ is an unbiased estimator of ν_i. It is evident that $\hat{\theta}$ is an unbiased estimator of θ. $\hat{\nu}_i$ and $\hat{\theta}$ are estimated using independent samples ($\hat{Y}$ and $\hat{Y}_0$ respectively), and hence

$$\begin{aligned}\mathbb{E}_{Y_0,Y}[\hat{\rho}] &= \mathbb{E}_{Y_0,Y}[\hat{\theta}\hat{\nu}_i] \\ &= \mathbb{E}_{Y_0}[\hat{\theta}]\mathbb{E}_{Y}[\hat{\nu}_i] \\ &= \theta\nu_i \\ &= \hat{\rho}.\end{aligned}$$

By Lemma 2,

$$\operatorname{Var}\hat{\rho}_i = \theta \operatorname{Var}\hat{\nu}_i + \nu_i \operatorname{Var}\hat{\theta} + \operatorname{Var}\hat{\theta}\operatorname{Var}\hat{\nu}_i.$$

□

C Basic probability lemmas

Lemma 1 (Mean and variance of the sum of two random variables)**.** *Let x and y be two random variables with mean 0, variances σ_x^2 and σ_y^2 and a correlation coefficient of ρ. Then, the estimator $z = \alpha x + (1-\alpha)y$, where $0 \le \alpha \le 1$ also has mean 0 and has minimum variance σ_z^2 when*

$$\alpha = \begin{cases} 0 & \rho > \frac{\sigma_x}{\sigma_y} \\ 1 & \rho > \frac{\sigma_x}{\sigma_y} \\ \frac{\sigma_y(\sigma_y - \rho\sigma_x)}{\sigma_x^2+\sigma_y^2-2\rho\sigma_x\sigma_y} & \textit{otherwise}, \end{cases} \qquad \sigma_z^2 = \begin{cases} \sigma_y^2 & \rho > \frac{\sigma_x}{\sigma_y} \\ \sigma_x^2 & \rho > \frac{\sigma_x}{\sigma_y} \\ \frac{\sigma_x^2\sigma_y^2(1-\rho^2)}{\sigma_x^2+\sigma_y^2-2\rho\sigma_x\sigma_y} & \textit{otherwise}. \end{cases}$$

In general, if x_i are uncorrelated random variables, $z = \sum_i \alpha_i x_i$ where $\sum_i \alpha_i = 1$ has mean and optimal variance,

$$\frac{1}{\sigma_z^2} = \sum_i \frac{1}{\sigma_i^2}.$$

Proof. For notational convenience, let $\bar{\alpha} \stackrel{\text{def}}{=} 1 - \alpha$. That z has mean 0 follows directly from the linearity of expectations. The variance of z can be calculated as follows:

$$\begin{aligned}\sigma_z^2 &\stackrel{\text{def}}{=} \operatorname{Var}(z) && = \mathbb{E}[z^2] - \mathbb{E}[z]^2 \\ &= \mathbb{E}[(\alpha x + \bar{\alpha} y)^2] - 0 \\ &= \mathbb{E}[\alpha^2 x^2 + \bar{\alpha}^2 y^2 + 2\alpha\bar{\alpha}xy] \\ &= \alpha^2\sigma_x^2 + \bar{\alpha}^2\sigma_y^2 + 2\alpha\bar{\alpha}\mathbb{E}[xy] \\ &= \alpha^2\sigma_x^2 + \bar{\alpha}^2\sigma_y^2 + 2\alpha\bar{\alpha}\rho\sigma_x\sigma_y,\end{aligned}$$

using the fact that $\rho \stackrel{\text{def}}{=} \frac{\mathbb{E}[xy]}{\sigma_x\sigma_y}$.

We introduce Lagrange multipliers $\lambda_1, \lambda_2 \ge 0$ to handle the constraint that $0 \le \alpha \le 1$,

$$\begin{aligned}\mathcal{L} &= \alpha^2\sigma_x^2 + \bar{\alpha}^2\sigma_y^2 + 2\alpha\bar{\alpha}\rho\sigma_x\sigma_y + \lambda_1\alpha + \lambda_2\bar{\alpha} \\ \frac{d}{d\alpha}\mathcal{L} &= 2\alpha\sigma_x^2 - 2(1-\alpha)\sigma_y^2 + 2(1-2\alpha)\rho\sigma_x\sigma_y + \lambda_1 - \lambda_2 \quad = 2(\alpha(\sigma_x^2+\sigma_y^2-2\rho\sigma_x\sigma_y) - \sigma_y^2 + \rho\sigma_x\sigma_y + \lambda_1' - \lambda_2')\end{aligned}$$

where λ_1' and λ_2' are suitably redefined to absorb the constant.

This quantity is minimized when the gradient with respect α is 0,

$$\begin{aligned}\alpha(\sigma_x^2+\sigma_y^2-2\rho\sigma_x\sigma_y) &= \sigma_y^2 - \rho\sigma_x\sigma_y + \lambda_2' - \lambda_1' \\ \alpha &= \frac{\sigma_y^2 - \rho\sigma_x\sigma_y + \lambda_2' - \lambda_1'}{\sigma_x^2+\sigma_y^2-2\rho\sigma_x\sigma_y} \\ \bar{\alpha} &= \frac{\sigma_x^2 - \rho\sigma_x\sigma_y - \lambda_2' + \lambda_1'}{\sigma_x^2+\sigma_y^2-2\rho\sigma_x\sigma_y}.\end{aligned}$$

The KKT conditions give us that $\lambda'_1\alpha = 0$ and $\lambda'_2(1-\alpha) = 0$, which implies that only one of λ'_1 or λ'_2 are non-zero. We can see that $\alpha = 0$ when $\sigma_y^2 - \rho\sigma_x\sigma_y < 0$, or when $\rho \geq \frac{\sigma_y}{\sigma_x}$. Likewise, $\alpha = 1$ when $\sigma_x^2 - \rho\sigma_x\sigma_y < 0$, or when $\rho \geq \frac{\sigma_x}{\sigma_y}$. This gives us the result on α.

The value of σ_z^2 when $\alpha = 0$ or $\alpha = 1$ is simply σ_y^2 or σ_x^2. When $0 < \alpha < 1$, it is,

$$\begin{aligned}
\sigma_z^2 &= \frac{\sigma_x^2\sigma_y^2(\sigma_y - \rho\sigma_x)^2 + \sigma_x^2\sigma_y^2(\sigma_x - \rho\sigma_y)^2 + 2\rho\sigma_x^2\sigma_y^2(\sigma_y - \rho\sigma_x)(\sigma_x - \rho\sigma_y)}{(\sigma_x^2 + \sigma_y^2 - 2\rho\sigma_x\sigma_y)^2} \\
&= \sigma_x^2\sigma_y^2\frac{\sigma_y^2 + \rho^2\sigma_x^2 - 2\rho\sigma_x\sigma_y + \sigma_x^2 + \rho^2\sigma_y^2 - 2\rho\sigma_x\sigma_y + 2\rho(\sigma_x\sigma_y - \rho\sigma_y^2 - \rho\sigma_x^2 + \rho^2\sigma_x\sigma_y)}{(\sigma_x^2 + \sigma_y^2 - 2\rho\sigma_x\sigma_y)^2} \\
&= \sigma_x^2\sigma_y^2\frac{(1-\rho^2)(\sigma_x^2 + \sigma_y^2 - 2\rho\sigma_x\sigma_y)}{(\sigma_x^2 + \sigma_y^2 - 2\rho\sigma_x\sigma_y)^2} \\
&= \frac{(1-\rho^2)\sigma_x^2\sigma_y^2}{\sigma_x^2 + \sigma_y^2 - 2\rho\sigma_x\sigma_y}
\end{aligned}$$

□

Lemma 2 (Mean and variance of the product of two random variables)**.** *Let x and y be two independent random variables with means μ_x and μ_y, and variances σ_x^2 and σ_y^2. Then, the estimator $z = xy$ has mean $\mu_x\mu_y$ and variance*

$$\sigma_z^2 = \sigma_x^2\sigma_y^2 + \mu_x^2\sigma_y^2 + \sigma_x^2\mu_y^2.$$

Proof. If x and y are independent, $\mathbb{E}[xy] = \mathbb{E}[x]\mathbb{E}[y]$. Thus $\mathbb{E}[z] = \mu_x\mu_y$.

The variance of z can be calculated as follows:

$$\begin{aligned}
\text{Var}(z) &= \mathbb{E}[z^2] - \mathbb{E}[z]^2 \\
&= \mathbb{E}[(xy)^2] - \mathbb{E}[xy]^2 \\
&= \mathbb{E}[x^2]\mathbb{E}[y^2] - \mathbb{E}[x]^2\mathbb{E}[y]^2 \\
&= (\sigma_x^2 + \mu_x^2)(\sigma_y^2 + \mu_y^2) - \mu_x^2\mu_y^2 \\
&= \sigma_x^2\sigma_y^2 + \mu_x^2\sigma_y^2 + \sigma_x^2\mu_y^2 + \mu_x^2\mu_y^2 - \mu_x^2\mu_y^2 \\
&= \sigma_x^2\sigma_y^2 + \mu_x^2\sigma_y^2 + \sigma_x^2\mu_y^2.
\end{aligned}$$

□

Lemma 3 (Mean and variance of the ratio of two random variables)**.** *Let x and y be two random variables with means μ_x and μ_y, variances σ_x^2 and σ_y^2 and correlation ρ. Furthermore, y has positive support (i.e. $y > 0$). Then, $z = x/y$ approximately has mean μ_x/μ_y and variance*

$$\sigma_z^2 \approx \frac{\mu_x^2}{\mu_y^2}\left(\frac{\sigma_x^2}{\mu_x^2} + 2\rho\frac{\sigma_x\sigma_y}{\mu_x\mu_y} + \frac{\sigma_y^2}{\mu_y^2}\right).$$

Proof. Let $f(x, y) = \frac{x}{y}$. Even if x and y are independent, $\mathbb{E}[f(x, y)]$ is not strictly $f(\mathbb{E}[x], \mathbb{E}[y])$. However, taking a first-order Taylor expansion around (μ_x, μ_y), we get

$$\begin{aligned}
\mathbb{E}[f(x, y)] &\approx f(\mu_x, \mu_y) + f'_x(\mu_x, \mu_y)\mathbb{E}[x - \mu_x] + f'_y(\mu_x, \mu_y)\mathbb{E}[y - \mu_y] \\
&= \frac{\mu_x}{\mu_y}.
\end{aligned}$$

Taking a similar approach to calculate variance, we get,

$$\begin{aligned}
\text{Var}(f(x,y)) &\approx \mathbb{E}[(f(x,y) - \mathbb{E}[f(x,y)])^2] \\
&= \mathbb{E}[(f(\mu_x,\mu_y) + f'_x(\mu_x,\mu_y)(x-\mu_x) + f'_y(\mu_x,\mu_y)(y-\mu_y) - f(\mu_x,\mu_y))^2] \\
&= f'_x(\mu_x,\mu_y)^2\mathbb{E}[(x-\mu_x)^2] + f'_y(\mu_x,\mu_y)^2\mathbb{E}[(y-\mu_y)^2] + 2f'_x(\mu_x,\mu_y)f'_y(\mu_x,\mu_y)\mathbb{E}[(x-\mu_x)(y-\mu_y)] \\
&= f'_x(\mu_x,\mu_y)^2\sigma_x^2 + f'_y(\mu_x,\mu_y)^2\sigma_y^2 + 2f'_x(\mu_x,\mu_y)f'_y(\mu_x,\mu_y)\rho\sigma_x\sigma_y.
\end{aligned}$$

Noting that $f'_x(\mu_x,\mu_y) = \frac{1}{\mu_y}$ and that $f'_y(\mu_x,\mu_y) = -\frac{\mu_x}{\mu_y^2}$, we get,

$$\begin{aligned}
\text{Var}(f(x,y)) &\approx \frac{\sigma_x^2}{\mu_y^2} + \frac{\sigma_y^2\mu_x^2}{\mu_y^4} + 2\frac{\mu_x}{\mu_y^3}\rho\sigma_x\sigma_y \\
&= \frac{\mu_x^2}{\mu_y^2}\left(\frac{\sigma_x^2}{\mu_x^2} + 2\rho\frac{\sigma_x\sigma_y}{\mu_x\mu_y} + \frac{\sigma_y^2}{\mu_y^2}\right).
\end{aligned}$$

□

Lemma 4 (Mean and variance of a importance-weighted estimate.)**.** *Let p_i and q_i be two sets of independent random variables with means μ and ξ and variances σ^2 and π^2 respectively. Then, $z = \frac{\sum_{i=1}^n p_i^2 q_i}{\sum_{i=1}^n p_i}$ has mean ξ and variance,*

$$\sigma_z^2 \approx \frac{\mu^2\xi^2}{n}\left(1 + \frac{\sigma^2}{\mu^2}\right)^2\left(9\frac{\sigma^2}{\mu^2} + 4\frac{\pi^2}{\xi^2}\right).$$

Proof. From Lemma 3 we have that

$$\sigma_z^2 \approx \frac{\mu_x^2}{\mu_y^2}\left(\frac{\sigma_x^2}{\mu_x^2} + 2\rho\frac{\sigma_x\sigma_y}{\mu_x\mu_y} + \frac{\sigma_y^2}{\mu_y^2}\right),$$

where $x = \frac{1}{n}\sum_{i=1}^n p_i^2 q_i$ and $y = \frac{1}{n}\sum_{i=1}^n p_i$.

In the following, we will make Gaussian assumptions on any moments > 3 and ignore variance squared terms, e.g. $\sigma^4 \approx 0$. Thus,

$$\begin{aligned}
\mathbb{E}[x^3] &\approx 3\sigma_x^2\mu_x + \mu_x^3 \\
\mathbb{E}[x^4] &\approx 6\sigma_x^2\mu_x^2 + \mu_x^4 \\
\text{Var}[x^2] &= \mathbb{E}[x^4] - \mathbb{E}[x^2]^2 \\
&\approx 6\sigma_x^2\mu_x^2 + \mu_x^4 - (\mu_x^2 + \sigma_x^2)^2 \\
&= 4\sigma_x^2\mu_x^2
\end{aligned}$$

Let us solve for each term independently,

$$\begin{aligned}
\mu_x &= (\sigma^2 + \mu^2)\xi \\
&= \mu^2\xi(1 + \frac{\sigma^2}{\mu^2}) \\
\mu_y &= \mu \\
\sigma_x^2 &\approx \frac{1}{n}(\text{Var}[p_i^2]\xi^2 + (\sigma^2 + \mu^2)^2\pi^2) \\
&< \frac{1}{n}(4\mu^2\sigma^2\xi^2 + 4\mu^4\pi^2) \\
&= \frac{4}{n}\mu^4\xi^2\left(\frac{\sigma^2}{\mu^2} + \frac{\pi^2}{\xi^2}\right) \\
\sigma_y^2 &= \frac{1}{n}\sigma^2.
\end{aligned}$$

Finally, for $\rho\sigma_x\sigma_y = \mathbb{E}[xy] - \mu_x\mu_y$, we get,

$$\begin{aligned}
\rho\sigma_x\sigma_y &\stackrel{\text{def}}{=} \mathbb{E}[(\sum_{i=1}^{n} p_i^2 q_i)(\sum_{j=1}^{n} p_j)] - \mathbb{E}[(\sum_{i=1}^{n} p_i^2 q_i)]\mathbb{E}[(\sum_{j=1}^{n} p_j)] \\
&= \sum_{i=1}^{n} \mathbb{E}[p_i^3]\mathbb{E}[q_i] - \mathbb{E}[p_i^2]\mathbb{E}[q_i]\mathbb{E}[p_i] \\
&= \sum_{i=1}^{n} (\mathbb{E}[p_i^3] - \mathbb{E}[p_i^2]\mathbb{E}[p_i])\mathbb{E}[q_i] \\
&= \frac{1}{n}(3\sigma^2\mu + \mu^3 - (\sigma^2 + \mu^2)\mu)\xi \\
&= \frac{2}{n}\sigma^2\mu\xi.
\end{aligned}$$

noting that all other terms are 0.

Putting all of these together, we get,

$$\begin{aligned}
\sigma_z^2 &< \mu^2\xi^2(1 + \frac{\sigma^2}{\mu^2})^2 \left(\frac{4}{n}(\frac{\sigma^2}{\mu^2} + \frac{\pi^2}{\xi^2}) + \frac{4}{n}\frac{\sigma^2\mu\xi}{\mu^2\xi\mu} + \frac{1}{n}\frac{\sigma^2}{\mu^2}\right) \\
&= \frac{\mu^2\xi^2}{n}\left(1 + \frac{\sigma^2}{\mu^2}\right)^2 \left(9\frac{\sigma^2}{\mu^2} + 4\frac{\pi^2}{\xi^2}\right).
\end{aligned}$$

□