An Interpretable Joint Graphical Model for Fact-Checking from Crowds

An T. Nguyen[1] **Aditya Kharosekar**[1] **Matthew Lease**[1] **Byron C. Wallace**[2]
[1]University of Texas at Austin
[2]Northeastern University
atn@cs.utexas.edu, aditya.kharosekar@gmail.com, ml@utexas.edu, byron@ccs.neu.edu

Abstract

Assessing the veracity of claims made on the Internet is an important, challenging, and timely problem. While automated fact-checking models have potential to help people better assess what they read, we argue such models must be *explainable*, *accurate*, and *fast* to be useful in practice; while prediction accuracy is clearly important, model transparency is critical in order for users to trust the system and integrate their own knowledge with model predictions. To achieve this, we propose a novel probabilistic graphical model (PGM) which combines machine learning with crowd annotations. Nodes in our model correspond to claim veracity, article stance regarding claims, reputation of news sources, and annotator reliabilities. We introduce a fast variational method for parameter estimation. Evaluation across two real-world datasets and three scenarios shows that: (1) joint modeling of sources, claims and crowd annotators in a PGM improves the predictive performance and interpretability for predicting claim veracity; and (2) our variational inference method achieves scalably fast parameter estimation, with only modest degradation in performance compared to Gibbs sampling. Regarding model transparency, we designed and deployed a prototype fact-checker Web tool, including a visual interface for explaining model predictions. Results of a small user study indicate that model explanations improve user satisfaction and trust in model predictions. We share our web demo, model source code, and the 13K crowd labels we collected.[1]

Introduction

Fact-checking, the task of determining the veracity (or correctness) of claims, has attracted significant attention. Websites such as *Snopes, Politifact* and *Emergent*, which rely on professional fact-checkers to carefully analyze claims and sources in order to assess veracity, have emerged as valuable resources for assessing potential misinformation. Unfortunately, the scalability of this manual approach is challenged by the pace and volume of modern online media, which continually generate and report new claims.

Recent research has pursued development of automated fact-checking systems (Popat et al. 2017; Nakashole and Mitchell 2014; Samadi et al. 2016), but these systems do not explicitly model *joint* interactions between key variables: claims, sources and annotators. Moreover, while such models may realize reasonable prediction accuracy, it is not clear how explainable these models are, which we argue is critical for this domain in order for people to be able to trust a model or integrate its predictions with their own knowledge.

We adopt a probabilistic graphical model (PGM) framework for our approach. PGMs afford important benefits over alternative approaches when humans are assumed to be in 'in-the-loop': (i) transparency, (ii) incorporation of users' knowledge; and (iii) uncertainty quantification. Concerning (i), fact-checking is intended to assuage skeptics of claims, who would likely also be skeptical about any automated fact-checking tool. Consequently, the model should be transparent in how it arrived at its prediction. The second consideration — incorporating users' knowledge (ii) — may, first, improve predictive performance. Second, and perhaps more importantly, this allows model predictions to be seen as relative statements with respect to prior knowledge, rather than definitive judgments regarding claims; this is natural for the fact-checking task. Finally, uncertainty quantification (iii) is important for characterizing our confidence (or, from a Bayesian perspective, our beliefs) regarding predictions while accounting for potential sources of errors. PGMs realize all of these desiderata and allow joint reasoning over claim, source and annotator reliabilities. Crucially, PGMs afford clear interpretation, facilitating in-depth inspection and criticism of the system by individuals.

Our approach is further distinguished by its crowd component. Beyond using crowd labels to train our model, we can call upon the crowd at run-time, integrating human intelligence with machine learning to further boost accuracy.

Key contributions: (1) a novel, interpretable PGM for fact-checking, providing a unified framework for modeling source credibility and stance with respect to individual claims, crowd worker reliability, and claim veracity; and (2) an efficient variational inference method for model estimation; (3) a small user study suggesting that model explanations improve user satisfaction and trust in the model.

Methods

Model

We assume that there are n claims, m sources, and a set of news articles in which each article from a source $j \in$

[1] github.com/thanhan/fc-aaai18

$\{1, ..., m\}$ reports a claim $i \in \{1, ..., n\}$. An example:
Claim: *ISIS is harvesting and selling human organs to help fund its operations.* (**Veracity:** *unknown*)
Headline: *Iraqi Official Accuses ISIS Of Harvesting Organs To Finance Operations.* (**Stance:** *observing claim*)
Source: *washington.cbslocal.com*

Let V_i be the **veracity** (correctness) of claim i, and S_{ij} the **stance** of source j w.r.t. claim i. A claim's veracity can be *true*, *false* or *unknown* (i.e., not enough is known about it). An article stance may be *for*, *against* or merely *observing* a claim (i.e., reporting it without assessing its veracity). We first define a multiclass logistic regression (LR) model parameterized by weight matrix W to predict stances S_{ij}:

$$p(S_{ij}|T_{ij}, W) = \text{Cat}(S_{ij} \mid \text{softmax}(T_{ij} \cdot W)) \qquad (1)$$

where Cat is the pmf of the Categorical distribution: $\text{Cat}(X \mid p) = \prod_i p_i^{\text{I}(X=i)}$, and I is an indicator function. T_{ij} encodes text features extracted from claim i and the article that reports this claim from source j. To predict the veracity of claim i, we define another LR model, parameterized by R, that uses all source stances for i as features.

$$p(V_i \mid S_i, R) = \text{Cat}(V_i \mid \text{softmax}(S_i \cdot R)) \qquad (2)$$

where $S_i = [S_{i1}, \dots, S_{im}]$ is a vector of the stances assumed by articles that report on claim i. In this model, we predict claim veracity based on the stances assumed by all articles reporting on it. We use the following coding scheme for stances: *for*$=1$, *observing*$=0$, and *against*$=-1$. Under this scheme, R can be interpreted as learning the reputation scores of the sources. While some previous work (Popat et al. 2017; Nakashole and Mitchell 2014; Wang 2017) has tried to predict claim veracity using textual features, we do not use them in this work because such features may be difficult for users to interpret. For example, one of the top textual features for predicting veracity is the word 'journalist'; but why would the presence of that word increase the probability that the claim is true?

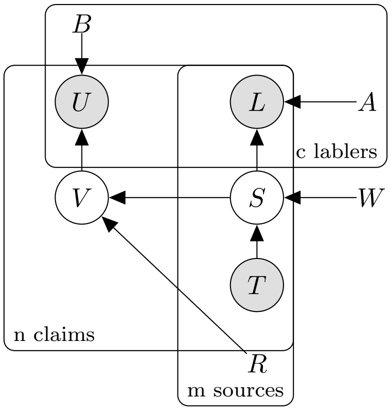
Next we define the worker model. Consider labeler k (here, a crowd worker or a professional journalist). Let L_{ijk} and U_{ik} be the labels for article stance and veracity, respectively, provided by labeler k. That is, labeler k estimates S_{ij} to be L_{ijk} and estimates V_i to be U_{ik}.

$$p(L_{ijk} \mid S_{ij}, A^{(k)}) = \text{Cat}(L_{ijk} \mid A^{(k)}_{S_{ij}}) \qquad (3)$$

$$p(U_{ik} \mid V_i, B^{(k)}) = \text{Cat}(U_{ik} \mid B^{(k)}_{V_i}) \qquad (4)$$

where $A^{(k)}$ is the stance confusion matrix for labeler k: $A^{(k)}_{st}$ is the probability that the labeler provides the stance label t for an article with true stance label s. The claim veracity confusion matrix $B^{(k)}$ specifies the probability of labeler k providing each veracity label, conditioned on the true label. These confusion matrices encode the quality of labels provided by individuals. The confusion matrix for a perfect labeler (who makes no mistakes) would be the identity matrix.

Our model assumes all annotators are fallible. This is particularly appropriate for fact-checking because the task is difficult and can be somewhat subjective. Our model can in principle incorporate users' knowledge, e.g., if users believe that journalists are only sometimes correct (say 90% of the time). However, we are limited in this study by our dataset, which includes only a single journalist 'gold' label per example. For the purpose of evaluation (only), we therefore assume that journalists are perfect labelers. Also, we call on the crowd only to collect stance labels; future work could investigate the more challenging task of asking the crowd to assess claim veracity.



Symbol	Description
V	claim veracity
U	veracity label (by journalists)
R	source reputation
T	text features
S	article headline stance
L	stance label (by crowds and journalists)
W	parameters for predicting stances
A	labeler confusion matrix for stance
B	labeler confusion matrix for veracity

Figure 1: Our graphical model and the symbols we use.

We interpret the veracity V_i and the stance S_{ij} as hidden variables, while W, R, A, B are parameters, which we learn using the EM algorithm (Dempster, Laird, and Rubin 1977).[2] In the E-step (assuming that the parameters are known), we infer the posterior distribution over the hidden variables. In the M-step, we find the expected maximum likelihood estimation (MLE) values of the parameters, where the expectation is over the posterior distribution in the E-step. These steps are repeated until convergence. In the following sections we present two instantiations for estimation: Gibbs sampling and variational inference. Gibbs sampling is simple and produces samples that asymptotically converge to the correct distribution, but it is often slow and not scalable in practice. Variational inference can be much faster, although may yield biased estimates. In a fact-checking system, we envision using variational inference to make near real-time predictions while performing Gibbs sampling in the background to improve these initial estimates.

[2] A Bayesian approach would place priors on parameters and perform inference over them. But this is computationally expensive and in our preliminary experiments did not yield improvements.

Gibbs sampling

Gibbs sampling (Geman and Geman 1984) works by iteratively drawing samples from the conditional distribution of each variable, given all other variables. To derive these conditionals, we first consider the (unnormalized) joint posterior distribution as follows:

$$p(S,V|L,U) \propto \prod_{i,j}\left[p(S_{ij}|T_{ij},W)\prod_{k}p(L_{ijk}|S_{ij},A^{(k)})\right] \prod_{i}\left[p(V_i|S_i,R)\prod_{k'}p(U_{ik'}|V_i,B^{(k')})\right] \tag{5}$$

where i is over all claims; j is over all sources having articles about claim i; k is over all workers who have provided labels for stance S_{ij}; and k' is over all workers who provided labels regarding the veracity of claim V_i. Picking out the terms involving V_i yields the unnormalized conditional:

$$p(V_i \mid \ldots) \propto p(V_i \mid S_i, R)\prod_{k'}p(U_{ik'} \mid V_i, B^{(k')}) \tag{6}$$

Similarly, we have the conditional for S_{ij}:

$$\begin{aligned} p(S_{ij} \mid \ldots) =& p(S_{ij} \mid T_{ij}, W)\cdot \\ & \prod_{k}p(L_{ijk} \mid S_{ij}, A^{(k)})p(V_i \mid S_i, R) \end{aligned} \tag{7}$$

Equation 6 samples claim i veracity from a distribution that is the product of two distributions: (1) the veracity prediction for claim i based on the current stances, and, (2) the likelihood of the veracity labels for this claim, given the labelers' confusion matrices. Equation 7 samples the stance of the article from source j for claim i from the product of three distributions. The first two are similar to the veracity sampling while the third is the likelihood of claim i veracity given corresponding stances. We will see that the update equations for variational inference have a similar structure.

In the M-step, we need to find the (expected) maximum likelihood (ML) solutions for the Logistic Regression (LR) parameters (W and R), and the confusion matrices for the crowd workers. For the former, we fit an LR model for each Gibbs sample in the E-step and average the parameters of these fitted models. For example, let $\{S^{(1)}, \ldots, S^{(g)}\}$ be the g Gibbs samples for S. We fit g sets of LR parameters $\{W^{(1)}, \ldots, W^{(g)}\}$, one for each sample: $W^{(i)} = \text{argmax}_W p(S^{(i)} \mid T, W)$ for $i = 1, \ldots, g$. The final solution is simply the average: $W = \sum_{i=1}^{g} W^{(i)}/g$. Empirically, we found this to provide better results than simply fitting one LR model weighted by the Gibbs samples, suggesting that there are complex interactions between the variables S and V. For the labeler confusion matrices, the (expected) ML solutions are simply the proportions: $A_{st}^{(k)} = E_{st}^{(k)} / \sum_{t'} E_{st'}^{(k)}$, where $E_{st}^{(k)}$ is the expected number of times that labeler k provided labels t for an instance of true label s (where the expectation is taken over the Gibbs samples).

Variational Inference

Variational Inference (Wainwright and Jordan 2008) approximates complex posterior distributions using a simpler distribution. The idea is to perform optimization to make this simple distribution as 'close' to the complex distribution as possible. Here, we want to approximate the posterior distribution $p(S, V)$ (implicitly conditioned on the observed variables and parameters). Using the 'mean field' assumption (Opper and Saad 2001), we first introduce a fully factorized distribution over S and V:

$$q(S,V) = \prod_{i,j} q(S_{ij}) \prod_{i} q(V_i) \tag{8}$$

where we further assume that each factor has univariate categorical distribution with parameter α for claim veracity factors and β for stance factors:

$$q(V_i) = \text{Cat}(V_i \mid \alpha_i) \tag{9}$$

$$q(S_{ij}) = \text{Cat}(S_{ij} \mid \beta_{ij}) \tag{10}$$

Our optimization problem is to minimize the KL divergence $\mathbb{KL}[q(S,V)||p(S,V)]$ with respect to α and β. This is equivalent to maximizing a log likelihood lower bound. Optimization can be performed via coordinate ascent, where each variable is updated while holding all others constant. The update equation has a convenient form: the updated variational distribution for a variable is proportional to the exponentiated expected (unnormalized) log posterior, where the expectation is taken with respect to the current variational distribution over all other variables

$$q^*(X_i) \propto \exp[\mathbb{E}_{q(X_{-i})} \log p(X)] \tag{11}$$

where X_{-i} is the set of all variables except X_i. In our model, we have:

$$\begin{aligned} q^*(V_i) \propto \exp\Big[&\mathbb{E}_{S_i \sim q(S_i)} \log p(V_i \mid S_i, R) \\ &+ \sum_{k'} \log p(U_{ik'} \mid V_i, B^{(k')})\Big] \end{aligned} \tag{12}$$

$$\begin{aligned} q^*(S_{ij}) \propto \exp\Big[&\mathbb{E}_{S_i \sim q(-S_{ij})} \log p(V_i \mid S_i, R) \\ &+ \sum_{k} \log p(L_{ijk} \mid S_{ij}, A^{(k)}) + \log p(S_{ij} \mid T_{ij}, W)\Big] \end{aligned} \tag{13}$$

We now see the similarity to the Gibbs sampling update equations 6 and 7. Two key differences are: (i) instead of sampling, we update the variational distribution; and (ii) instead of conditioning on other variables, we take the expectation over them under the current variational distribution. The main difficulty is computing the following expectation (w.r.t. the high dimensional vector S_i):

$$\mathbb{E}_{S_i \sim q(S_i)} \log\Big(\sum_{k} \exp(S_i \cdot R_k)\Big) \tag{14}$$

Where the sum of exp comes from the softmax function. This is a common problem in variational inference, and many solutions have been proposed, including: a log concavity bound for correlated topic models (Blei and Lafferty 2007), a Bohning bound for factor analysis (Khan et al. 2010), and more general techniques for non-conjugate models with Laplace or delta methods (Wang and Blei 2013). Recent black-box inference algorithms (Ranganath, Gerrish, and Blei 2014) can automatically handle this, but we are interested in a very efficient implementation to power a near real-time system. We thus propose using a Taylor approximation on the $\log$ and $\exp$ functions. For $\log$, a Taylor expansion about a gives the approximation (with error $o(\sum_i X_i - E\sum_i X_i)$):

$$\mathbb{E}\log\Big(\sum_i X_i\Big) \approx \log(a) + \Big[\sum_i \mathbb{E}(X_i) - a\Big]\Big/a \quad (15)$$

We can optimize for a by setting the derivative of the above to zero and solving to yield: $a = \mathbb{E}(\sum_i X_i)$. Plugging this back in, we get $\mathbb{E}\log(\sum_i X_i) \approx \log[(\sum_i \mathbb{E}(X_i)]$. Similarly for $\exp$, we have: $\mathbb{E}\exp(X) \approx \exp[\mathbb{E}(x)]$. We can thus approximate Equation 14 by simply pushing the expectation into $\log$ and $\exp$. Although this does not preserve the likelihood lower bound, it is efficient in practice.

Given a method to approximate Equation 14, we can now approximate the updates in 12 and 13, which are alternately applied until convergence in coordinate ascent. Convergence appears to be very quick, in just a few iterations.

Online and Transfer Scenarios

In the online scenario, we assume no journalist stance labels are available in the training set, but that systems may request test set stance labels from the crowd at run-time. To select articles for which to solicit crowd stance labels, we calculate article scores and select articles stochastically w.r.t. the softmax of the scores. For an article from source j about claim i, we have:

$$\text{score}_{ij} = \lambda_c \cdot H(V_i) + \lambda_s \cdot H(S_{ij}) + \lambda_r \cdot r_j \quad (16)$$

where $H(V_i)$ is the entropy in the prediction of claim i veracity, $H(S_{ij})$ is the same entropy for this article stance and r_j is a measure of source j reputation: $r_j = |R_{j1}| + |R_{j2}| + |R_{j3}|$ — recall that R is the LR parameter for veracity prediction. We normalize the above $H(V_i)$, $H(S_{ij})$ and r_j for them to be on the same scale. λ_c, λ_s and λ_r are weight parameters. We explored setting these to values over $\{0, 1, 10\}$ and we report results for the best configuration $\lambda_c = 10$, $\lambda_s = 0$ and $\lambda_r = 1$ for all methods.[3] This configuration roughly corresponds to selecting the articles reporting the most uncertain claim from the most credible source.

Our methods so far assume the availability of stance labels, which are often limited in practice, while veracity labels are more common. Transfer learning (Pan and Yang 2010; Raina et al. 2007) aims to reuse the knowledge learned in a source domain in a target domain, where few or no labels are available. In the transfer scenario, we want to adapt the trained stance classifier to a new dataset for which we have no stance labels. We achieve this by simply using the parameters of the trained stance classifier to initialize the new stance classifier. We then perform inference and learning on the new dataset as described above.

Evaluation

Data. We report results on two datasets. First, we use the *Emergent* dataset (Ferreira and Vlachos 2016), which consists of 300 claims and over 2,595 news article headlines reporting these claims. The labels are provided by professional journalists. Note that article stance labels available with *Emergent* are not available for many other related datasets, e.g., *Snopes* (Popat et al. 2017) includes only veracity labels. The distribution of journalist stance labels is 47.7% *for*, 15.2% *against* and 37.1% *observing*. For claim veracity, 21% of claims are labeled *true*. 37% as *false*, and notably, 42% as *unknown*. Ferreira and Vlachos (2016) split the dataset into train and test sets of 240 and 60 claims (corresponding to 2,071 and 524 article headlines), respectively. We further split their training set into our own training and validation sets of 180 and 60 claims, respectively. We use the validation set to develop our method and tune the hyperparameters, then report test results. We also use their source code[4] to extract text features from the claims and articles.

For our second dataset, we use *Snopes* (Popat et al. 2017)[5] for our transfer scenario. Lacking stance labels, we transfer the stance classifier trained on *Emergent*. This dataset consists of 4486 claims, with veracity labeled as *true* or *false* (no *unknown* category). Following Popat et al. (2017), we put the claims into Google to retrieve relevant articles (we retrieve 10 articles from the first result page). We split the dataset into 60% train, 20% validation and 20% test.

Crowdsourced labels collection. We use Amazon Mechanical Turk to collect 5 stance labels for each of the 2,595 article headlines in *Emergent*. While this task might seem simple, we find it to be often difficult and somewhat subjective based on both our own examination of the data and crowd worker disagreement (with journalist labels and one another's). Ferreira and Vlachos (2016) do not report multi-annotator agreement statistics for reference-standard journalist annotations. For the (claim, headline) pair shown earlier (regarding ISIS possibly harvesting human organs), the journalist labeled it *observing* while all crowd workers labeled it *for*.

Close reading of the text and interpretation of task guidelines is necessary. In pilot experiments (unreported) that we ran to iteratively refine the design of our data collection interface, we found that we could improve label quality by requiring workers to write a short rationale for each judgment (copying words from the headline), adapting recent work by McDonnell et al. (2016), in which the authors found that this simple technique yields improvement in collecting web search relevance judgments.

Baseline. We aim to explore experimentally whether a joint model (realized via our PGM) improves predictive performance as compared to independent models. Our baseline thus consists of two LR models: the first predicts article

[3] Results were not particularly sensitive to these parameters.

[4] https://github.com/willferreira/mscproject

[5] https://goo.gl/gGzPji

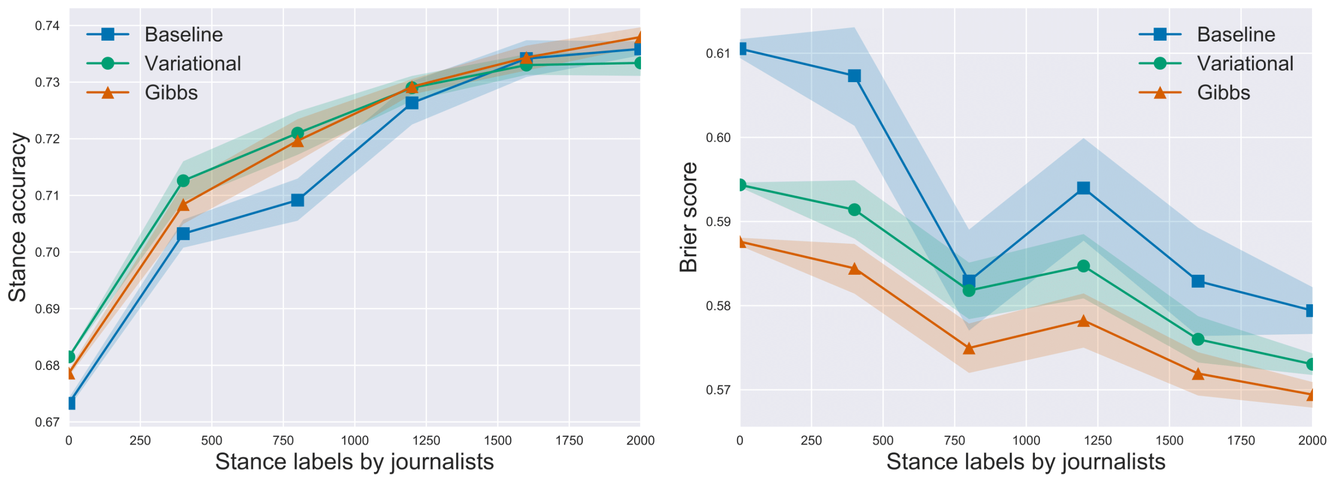


Figure 2: **Left:** offline scenario results for stance accuracy (higher is better). **Right:** Brier score for claim veracity (lower is better). The ratio of crowd vs. journalist stance training labels is varied along the x-axis (from 0 to 2000 journalist labels). Results are averaged over 10 runs; confidence bands of one standard deviation (68%) are shown.

stance from the text, and the second predicts claim veracity from the predicted stances using the same features used by our PGM. Thus for stance classification, our baseline is effectively (Ferreira and Vlachos 2016). We also use Dawid and Skene's (1979) method to aggregate crowd labels; despite its age, this was shown in a recent crowd aggregation benchmarking study (Sheshadri and Lease 2013) to be the most consistently strong method. For the transfer scenario, the baseline uses the stance classifier trained on Emergent to predict stances in the Snopes dataset.

Metrics. For stance prediction, we follow (Ferreira and Vlachos 2016) in using accuracy (given the class distribution reported above, a naive method that always predicted *for* would achieve 47.7%). For predicting claim veracity, we find accuracy to be noisy, possibly due to the small size of the validation and test sets. We thus instead report the Brier score (Brier 1950) score for multiclass forecasting: $\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{C}(P_{ij} - I_{ij})^2$, where we sum over n claims and C categories (we have 3 categories in Emergent and 2 categories in Snopes). P_{ij} is the predicted probability that claim i is of class j and I_{ij} is the indicator that is 1 if claim i is of class j and 0 otherwise. This metric quantifies probability calibration, which is an important property for our application because we believe users will desire estimates of confidence associated with predictions.

Results

We consider three scenarios: offline, online and transfer. In offline and online, we use the Emergent dataset and assume that the training articles are given with their journalist veracity labels and crowd stance labels. In transfer, we adapt the stance classifier trained on Emergent to the Snopes dataset.

Offline scenario (Figure 2). We vary the number of journalist stance labels available in the train set. At the left-most extreme of the Figure, only stance labels from the crowd are available; at the right-most extreme, all training examples have both crowd and journalist stance labels. We thus measure the relative importance of journalist stance labels for training. As expected, prediction improves as more journalist labels are added (also recall that we are evaluating accuracy w.r.t. journalist labels as the reference standard). With regard to the methods being compared, our variational and Gibbs perform better than the baseline, and Gibbs is better at claim veracity. This suggests that the meanfield assumption (that the variational distribution is fully factorized) has caused some mis-calibration in veracity prediction, possibly due to some strong inter-dependence in the variables.

Online scenario (Figure 3). The methods can acquire test set stance labels from the crowd. This experiment explores how on-demand crowd labels can be used at run-time to improve performance. We assume that we do not have access to journalist stance labels because these may be difficult to acquire rapidly in practice; this effectively represents a "worst-case" scenario. As expected, on-demand crowd labels tend to improve both stance and veracity predictions. Gibbs sampling provides further improvement over our variational approach, but at the cost of run-time.

Transfer scenario (Figure 4). We vary the number of (Snopes) training claims. Since Snopes does not has stance labels, we can only evaluate the veracity prediction. Our method and the baseline both utilize the stance classifier trained on Emergent (with all journalist stance labels), but we improve by modeling the uncertainty and interactions between different variables in the presented graphical models. The dataset is much larger, thus we can only use variational inference (Gibbs does not adequately scale). We observe consistent improvement over the baseline.

Discussion. Stance and veracity prediction are both difficult tasks. Overall, the benefit of getting more labels is small and the differences between methods are modest, although significant (as seen in the confidence band). As for runtime, in the Emergent dataset, Variational takes about 5 minutes to

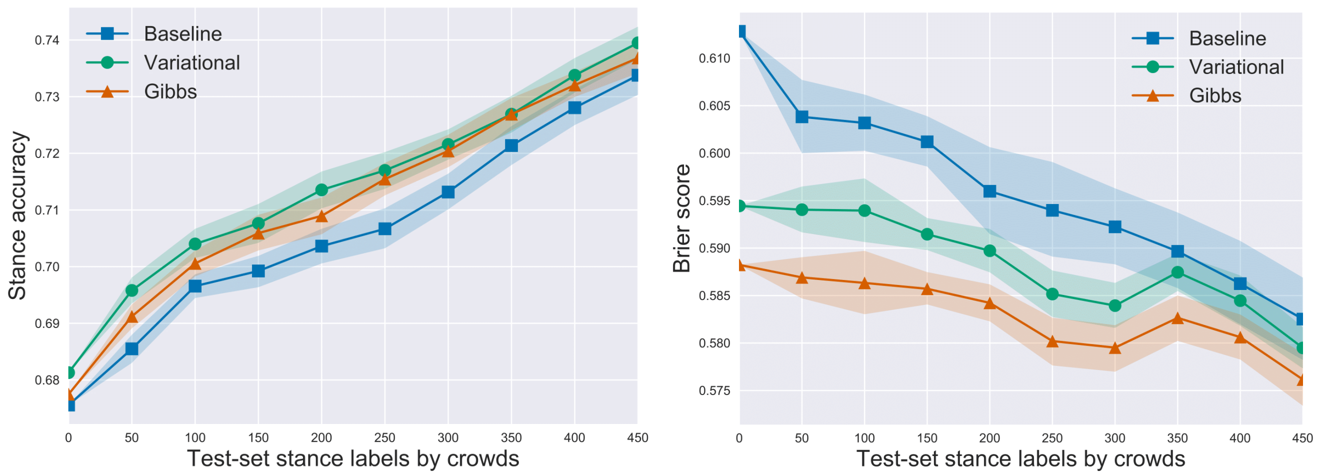


Figure 3: Online scenario results when on-demand crowd (only) labels for test-set articles are utilized at run-time. The number of (aggregated) crowd stance labels is varied on the x-axis.

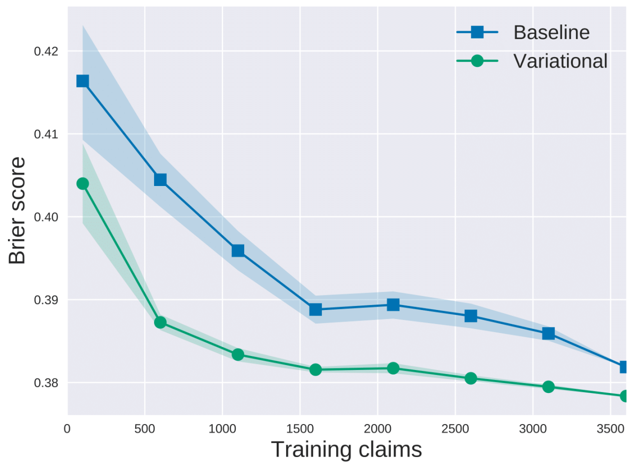


Figure 4: Veracity prediction on the Snopes dataset, transferring the stance classifier trained on Emergent.

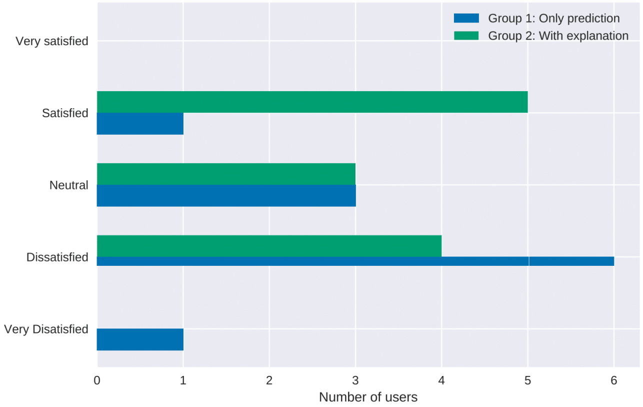


Figure 5: The histogram of user satisfaction in two groups.

train; Gibbs sampling requires about an hour. In transferring to the Snopes dataset, Variational takes nearly 2 hours (on a 3.50GHz machine).

Model interpretation and user study

For a fact-checking method to be useful, it has to be interpretable. Our final veracity prediction is based on very intuitive concepts: articles stance and sources reputation. In **Figure 6** we present an illustrative example of an explanation that can be presented to users.

To understand how users interpret our predictions, we conducted a small user study. We developed a tool with a web interface where users can enter their own claims or select from a number of example claims from the Emergent dataset. The tool uses Google to retrieve 10 articles relevant to the claim, and then makes a prediction regarding its veracity. We recruited 23 graduate students (mostly in computer and information science) to use the tool. Students were randomly assigned into one of two groups. Users in the first group were shown only the final veracity prediction, while those in the second group were provided an accompanying explanation similar to **Figure 6**. Students then completed an eight question survey regarding their satisfaction with, and trust of, the tool. In **Figure 5**, we present overall satisfaction results. In general, explanations increased user satisfaction. Assuming that the responses are numbered from 'Very dissatisfied' = 1 to 'Very satisfied' = 5, a two sample t-test yields $p = 0.058$ (the null here being no difference in satisfaction between groups). Regarding trust, users who saw the explanation found our tool more trustworthy, although the effect was less pronounced ($p = 0.138$).

Users who were dissatisfied with the tool tended to enter claims they believed to be true or false and did not get what they expected. However, in these cases, model explanations helped them to understand how the tool arrived at its prediction. Other survey questions revealed that users are generally open to fact-checking using crowdsourcing, although some were concerned about workers ability and potential biases.

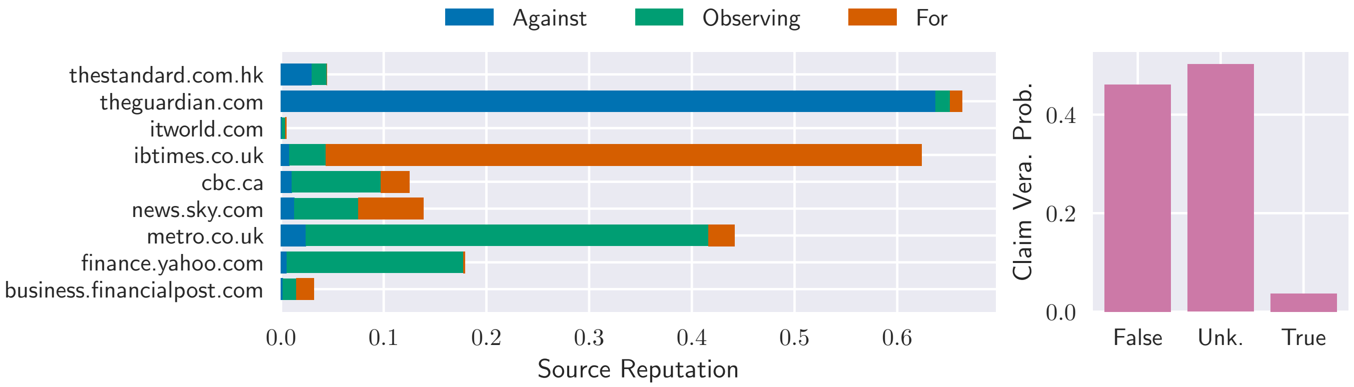


Figure 6: **Claim:** *IBM will cut more than 110,000 jobs this week* (`www.emergent.info/IBM-job-cuts`). **Left**: Bar height shows the predicted reputation of each source. Bar pattern shows the variational inferred posterior distribution over each article's stance (*for*, *against*, or *observing*). **Right**: The posterior over the claim veracity. This claim is predicted unlikely to be *true* because credible sources are mixed in stance, which our model explicitly captures and users may inspect.

Related Work

Truth discovery. Work on truth discovery (Pasternack and Roth 2013; Dong et al. 2015; Li et al. 2016) has aimed to resolve conflicts in data from multiple sources by estimating their quality. This work has typically assumed that 'claims' have a concrete structure. For example, (Pasternack and Roth 2013) considered a dataset involving book authorship, where claims have the structure 'person X authors book Y'. Recent work has considered unstructured text claims (Ferreira and Vlachos 2016; Popat et al. 2016; 2017; Wang 2017), but these approaches still depend on training data from expert fact-checkers. By contrast, we exploit both expert and crowd labels for training and consider an online scenario in which crowd labels are collected at run-time.

Deceptive opinion. Another related task is detecting 'deceptive' opinions, i.e., fake reviews for hotels or restaurants (Ott et al. 2011; Li et al. 2014). The standard approach is supervised classification based on linguistic features of the review texts, which is often much longer than claims text. In contrast, our method predicts the veracity of a claim based on the articles reporting it, their stances (support/refute), and the (estimated) credibilities of the sources.

Rumor detection in social media. Previous work in this space (Derczynski et al. 2017; Liu et al. 2015; Volkova et al. 2017) has focused on analyzing existing social media content, typically about some specific events, to predict their stance and veracity. Our work considers a more general and challenging task where users can enter an arbitrary claim.

Crowdsourcing aggregation. An oft-studied problem in crowdsourcing is how to best aggregate crowd responses to infer the true label (Sheshadri and Lease 2013). PGMs provide a natural framework for this; true labels for instances can be treated as hidden variables (Dawid and Skene 1979; Raykar et al. 2010; Liu, Peng, and Ihler 2012; Bi et al. 2014; Tian and Zhu 2015). While most previous work assumes aggregations for different instances to be independent, recent work has proposed models accounting for the structure of these instances. Rodrigues, Pereira, and Ribeiro (2014) extend CRFs to aggregate crowdsourced sequential data, where the true labels of neighboring instances are correlated. Lakkaraju et al. (2015) propose clustering to exploit the relationship between similar instances (they also cluster the workers). We similarly relate the instances (article stances) through the sources they belong to and the claims they report.

Interpretable machine learning. While machine learning has traditionally optimized predictive performance, recent work has sought to improve interpretability. A PGM approach has been used for clustering (Kim, Rudin, and Shah 2014) and human decision making (Lakkaraju and Leskovec 2016). For classification, popular interpretable methods are decision trees (Quinlan 1986) and sparse/prototype models (Tibshirani 1996; Bien and Tibshirani 2011); but they assume that features are given. Here, the features for veracity prediction are article headline stances, which are predicted.

Conclusions

Fact-checking has become an increasingly important social problem, and automating it presents many technical challenges. We have presented a hybrid machine-crowd PGM approach that integrates human intelligence with system scalability to jointly model stance, veracity and crowdsourced labels. Our method achieves relatively strong predictive performance, and, crucially, provides transparency that affords critical interpretation and assessment of system outputs. We share our online demo, source code, and 13K collected crowd labels.

Acknowledgments. We thank Greg Durrett and the anonymous reviewers for valuable feedback. We also thank the crowdworkers and journalists for their participation. This work is supported in part by by National Science Foundation grant No. 1253413. Any opinions, findings, and conclusions or recommendations expressed by the authors are entirely their own and do not represent those of the sponsoring agencies.

References

Bi, W.; Wang, L.; Kwok, J. T.; and Tu, Z. 2014. Learning to predict from crowdsourced data. In *UAI*.

Bien, J., and Tibshirani, R. 2011. Prototype selection for interpretable classification. *The Annals of Applied Statistics*.

Blei, D. M., and Lafferty, J. D. 2007. A correlated topic model of science. *The Annals of Applied Statistics* 17–35.

Brier, G. W. 1950. Verification of forecasts expressed in terms of probability. *Monthly weather review* 78(1):1–3.

Dawid, A. P., and Skene, A. M. 1979. Maximum likelihood estimation of observer error-rates using the em algorithm. *Applied statistics* 20–28.

Dempster, A. P.; Laird, N. M.; and Rubin, D. B. 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society* 1–38.

Derczynski, L.; Bontcheva, K.; Liakata, M.; Procter, R.; Hoi, G. W. S.; and Zubiaga, A. 2017. Semeval-2017 task 8: Rumoureval: Determining rumour veracity and support for rumours. *arXiv preprint arXiv:1704.05972*.

Dong, X. L.; Gabrilovich, E.; Murphy, K.; Dang, V.; Horn, W.; Lugaresi, C.; Sun, S.; and Zhang, W. 2015. Knowledge-based trust: Estimating the trustworthiness of web sources. *Proceedings of the VLDB Endowment* 8(9):938–949.

Ferreira, W., and Vlachos, A. 2016. Emergent: a novel dataset for stance classification. In *North American Chapter of the Association for Computational Linguistics*. ACL.

Geman, S., and Geman, D. 1984. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. on pattern analysis and machine intelligence*.

Khan, M. E.; Bouchard, G.; Murphy, K. P.; and Marlin, B. M. 2010. Variational bounds for mixed-data factor analysis. In *Advances in Neural Information Processing Systems*.

Kim, B.; Rudin, C.; and Shah, J. A. 2014. The bayesian case model: A generative approach for case-based reasoning and prototype classification. In *NIPS*.

Lakkaraju, H., and Leskovec, J. 2016. Confusions over time: An interpretable bayesian model to characterize trends in decision making. In *NIPS*.

Lakkaraju, H.; Leskovec, J.; Kleinberg, J.; and Mullainathan, S. 2015. A bayesian framework for modeling human evaluations. In *SIAM Conference on Data Mining*.

Li, J.; Ott, M.; Cardie, C.; and Hovy, E. H. 2014. Towards a general rule for identifying deceptive opinion spam. In *ACL*.

Li, Y.; Gao, J.; Meng, C.; Li, Q.; Su, L.; Zhao, B.; Fan, W.; and Han, J. 2016. A survey on truth discovery. *SIGKDD Explor. Newsl.* 17(2):1–16.

Liu, X.; Nourbakhsh, A.; Li, Q.; Fang, R.; and Shah, S. 2015. Real-time rumor debunking on twitter. In *CIKM*.

Liu, Q.; Peng, J.; and Ihler, A. T. 2012. Variational inference for crowdsourcing. In *NIPS*.

McDonnell, T.; Lease, M.; Elsayad, T.; and Kutlu, M. 2016. Why is that relevant? collecting annotator rationales for relevance judgments. In *AAAI HCOMP*.

Nakashole, N., and Mitchell, T. M. 2014. Language-aware truth assessment of fact candidates. In *ACL (1)*, 1009–1019.

Opper, M., and Saad, D. 2001. *Advanced mean field methods: Theory and practice*. MIT press.

Ott, M.; Choi, Y.; Cardie, C.; and Hancock, J. T. 2011. Finding deceptive opinion spam by any stretch of the imagination. In *ACL*, 309–319. ACL.

Pan, S. J., and Yang, Q. 2010. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*.

Pasternack, J., and Roth, D. 2013. Latent credibility analysis. In *Proceedings of the 22nd international conference on World Wide Web*, 1009–1020. ACM.

Popat, K.; Mukherjee, S.; Strötgen, J.; and Weikum, G. 2016. Credibility assessment of textual claims on the web. In *CIKM*, 2173–2178.

Popat, K.; Mukherjee, S.; Strötgen, J.; and Weikum, G. 2017. Where the truth lies: Explaining the credibility of emerging claims on the web and social media. In *WWW*.

Quinlan, J. R. 1986. Induction of decision trees. *Machine learning* 1(1):81–106.

Raina, R.; Battle, A.; Lee, H.; Packer, B.; and Ng, A. Y. 2007. Self-taught learning: transfer learning from unlabeled data. In *ICML 2007*.

Ranganath, R.; Gerrish, S.; and Blei, D. M. 2014. Black box variational inference. In *AISTATS*, 814–822.

Raykar, V. C.; Yu, S.; Zhao, L. H.; Valadez, G. H.; Florin, C.; Bogoni, L.; and Moy, L. 2010. Learning from crowds. *Journal of Machine Learning Research* 11:1297–1322.

Rodrigues, F.; Pereira, F.; and Ribeiro, B. 2014. Sequence labeling with multiple annotators. *Machine learning*.

Samadi, M.; Talukdar, P.; Veloso, M.; and Blum, M. 2016. Claimeval: Integrated and flexible framework for claim evaluation using credibility of sources. In *AAAI*.

Sheshadri, A., and Lease, M. 2013. Square: A benchmark for research on computing crowd consensus. In *First AAAI Conference on Human Computation and Crowdsourcing*.

Tian, T., and Zhu, J. 2015. Max-margin majority voting for learning from crowds. In *Advances in Neural Information Processing Systems*, 1621–1629.

Tibshirani, R. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)* 267–288.

Volkova, S.; Shaffer, K.; Jang, J. Y.; and Hodas, N. 2017. Separating facts from fiction: Linguistic models to classify suspicious and trusted news posts on twitter. In *ACL(2)*.

Wainwright, M. J., and Jordan, M. I. 2008. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning* 1(1–2):1–305.

Wang, C., and Blei, D. M. 2013. Variational inference in nonconjugate models. *Journal of Machine Learning Research* 14(Apr):1005–1031.

Wang, W. Y. 2017. “liar, liar pants on fire”: A new benchmark dataset for fake news detection. In *ACL(2)*.