Supplemental Material for ConStance: Modeling Annotation Contexts to Improve Stance Classification

This document provides more details on the annotation study we developed (Appendix A), of the EM algorithm we briefly cover in the main text (Appendix B), a brief description of how we debugged this algorithm (Appendix C) and a description of how hyperparameters were set using a test set (Appendix D).

A More Details on Annotation Study

Figure 1 presents an overview of our study design. Outlined in black on the top left is an example of one of the questions posed to annotators. Each question developed consists of three main parts: a *target*, the text of a particular *tweet*, and a set of *additional information* about the tweet's author. At a high level, we first selected a set of tweet/target pairs. We then produced six questions for each tweet/target pair, one for each type of additional information/context we considered.

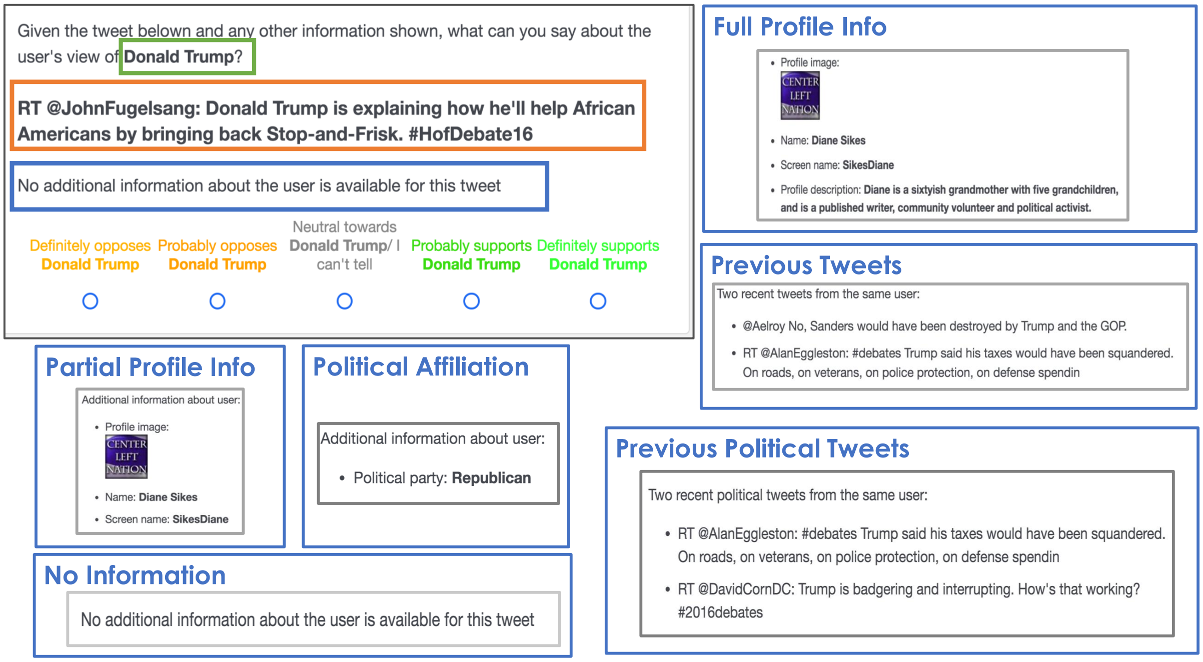
We initially selected a set of 480 tweet/target pairs to annotate, split evenly between the two targets. The 240 tweets for each target were selected by choosing 40 tweets from each possible combination of these two tweet-level properties (2 "tweet originality" types x 3 "target mention" types). After an initial investigation of results, we observed that the sample contained relatively few tweets from Republican users. To address this issue we sampled an additional 82 tweets from Republican users, for a final sample size of 562 tweets. These tweets imbalanced the sample design—they were no longer evenly distributed across the original categories—but they ensured sufficient counts for Republican users, which we believed might be useful for our analyses.

As a final point, we replace all URLs in both the tweets to be labeled and tweets shown in the additional information portion of the questions with the text "{{link}}". This decision was made in order to maintain control over the amount of information seen by annotators; if the links were left visible, annotators would vary in whether or not they clicked through. However, since URL information (e.g. domain name, page title, page content) provides useful information to the annotator, obscuring the links artificially increased the task's difficulty.

B Derivation of EM Algorithm

Figure 2 provides a graphical overview of the model; both it and Table 1 are reproduced from the main paper for convenience. Below, we outline the derivation of the EM algorithm used for inference.

The model's incomplete data likelihood function, Eq. (1), describes the joint probability, across all items, of Y_i, all values of S_i^c, and all values of R_i^{ca} assuming $\underline{X_i}$ is known and fixed.



Given the tweet belown and any other information shown, what can you say about the user's view of Donald Trump?

RT @JohnFugelsang: Donald Trump is explaining how he'll help African Americans by bringing back Stop-and-Frisk. #HofDebate16

No additional information about the user is available for this tweet

Definitely opposes Donald Trump
Probably opposes Donald Trump
Neutral towards Donald Trump/ I can't tell
Probably supports Donald Trump
Definitely supports Donald Trump
○ ○ ○ ○ ○
Full Profile Info
Previous Tweets
Partial Profile Info
Political Affiliation
Previous Political Tweets
No Information
CENTER LEFT NATION
CENTER LEFT NATION

- Profile image:
- Name: Diane Sikes
- Screen name: SikesDiane
- Profile description: Diane is a sixtyish grandmother with five grandchildren, and is a published writer, community volunteer and political activist.

Two recent tweets from the same user:

- @Aelroy No, Sanders would have been destroyed by Trump and the GOP.
- RT @AlanEggleston: #debates Trump said his taxes would have been squandered. On roads, on veterans, on police protection, on defense spendin

Additional information about user:

- Profile image:
- Name: Diane Sikes
- Screen name: SikesDiane

Additional information about user:

- Political party: Republican

Two recent political tweets from the same user:

- RT @AlanEggleston: #debates Trump said his taxes would have been squandered. On roads, on veterans, on police protection, on defense spendin
- RT @DavidCornDC: Trump is badgering and interrupting. How's that working? #2016debates

No additional information about the user is available for this tweet

Figure 1: In the black box is a single annotation question. The green box displays where the target is given, the orange box where the tweet text is displayed and the blue box where any additional information is given. The six information conditions are shown in the blue boxes.

Uppercase denotes random variables; lowercase, specific values. In line (2), we substitute in the equivalent model parameters.

$$\begin{aligned} p(\mathcal{D}|\theta, X) &= \prod_{i=1}^{N}\sum_{y}^{V} p(Y_i = y|\underline{x_i}, \mathcal{M}) \prod_{c}^{C_i} \\ &\qquad \sum_{s}^{V} p(S_i^c = s|y,\gamma)\prod_{a}^{A_i^c} p(r_i^{ca}|s,\alpha) && (1) \\ &= \prod_{i=1}^{N}\sum_{y}^{V} \mathcal{M}_y(\underline{x_i}) \prod_{c}^{C_i}\sum_{s}^{V} \gamma_{ys}^{c} \prod_{a}^{A_i^c} \alpha_{sr}^{a} && (2) \end{aligned}$$

To derive EM for this model, we treat the latent variables as a block, moving their joint distribution into a single term. A given tweet has a latent value y_i and a latent vector $\underline{s_i}$ containing one entry per context: $\underline{s_i} = (s_i^1, \ldots, s_i^{C_i})$. Returning to (1), we move the term $p(S_i^c = s|y,\gamma)$ left, outside the product over C_i contexts, explicitly representing each component s_i^c of $\underline{s_i}$ and summing over its latent values. With all of $\underline{s_i}$ in scope at once, we can rearrange the latent variables into

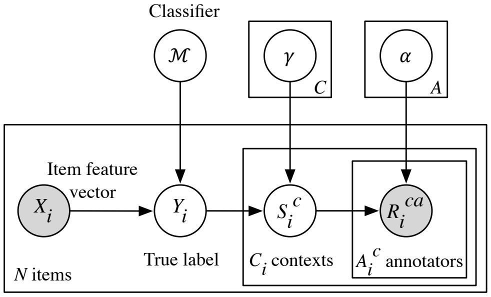


Figure 2: Graphical model for ConStance.

Var.	Meaning
$\underline{X_i}$	Feature vector of item i
Y_i	Latent true label of item i
S_i^c	Latent context-specific label of item i after noise from context c
R_i^{ca}	Label given by annotator a to item i in context c
V	Set of values for labels and annotations: $\{-1, 0, 1\}$
N	# of items, indexed by i
C	Set of contexts, indexed by c
A	Set of annotators, indexed by a
$\mathcal{M}$	Learned classifier
γ^c	$V \times V$ parameter matrix for context c
α^a	$V \times V$ parameter matrix for annotator a
$\mathcal{D}$	All observed data: all values of $\underline{X_i}$ and R_i^{ca}
Z	All latent variables: all values of Y_i and $\underline{S_i}$
θ	All model parameters: $\mathcal{M}, \gamma, \alpha$
T_i	All latent variables for item i: $(Y_i, \underline{S_i})$
$\tau_{i(y\underline{s})}$	Current estimate of all latent values for item i: $p(Y_i = y, \underline{S_i} = \underline{s} \mid \mathcal{D}, \theta)$

Table 1: Model variables.

a single term.

$$\begin{aligned} p(\mathcal{D}|\theta) &= \prod_{i=1}^{N} \sum_{y}^{V} p(y_i = y|\underline{x_i}, \mathcal{M}) \left(\sum_{s_i^1}^{V} \cdots \sum_{s_i^{C_i}}^{V} \right) \prod_{c}^{C_i} p(s_i^c = s|y_i, \gamma) \prod_{a}^{A_i^c} p(r_i^{ca}|s_i^c, \alpha) \\ &= \prod_{i=1}^{N} \sum_{y}^{V} \left(\sum_{s_i^1}^{V} \cdots \sum_{s_i^{C_i}}^{V} \right) p(y_i = y|\underline{x_i}, \mathcal{M}) p(\underline{s_i} = \underline{s}|y_i, \gamma) \prod_{c}^{C_i} \prod_{a}^{A_i^c} p(r_i^{ca}|s_i^c, \alpha) \\ &= \prod_{i=1}^{N} \sum_{y}^{V} \left(\sum_{s_i^1}^{V} \cdots \sum_{s_i^{C_i}}^{V} \right) p(y_i = y, \underline{s_i} = \underline{s} \mid \underline{x_i}, \mathcal{M}, \gamma) \prod_{c}^{C_i} \prod_{a}^{A_i^c} p(r_i^{ca}|s_i^c, \alpha) \end{aligned}$$

Next we introduce an indicator variable $T_{i(y\underline{s})} \in \{0, 1\}$ representing a configuration of latent variable assignments $(y_i, \underline{s_i}) \in Z$. We define $T_{i(y\underline{s})} = 1$ when tweet i has the specific configuration $(y_i = y, \underline{s_i} = \underline{s})$. This gives:

$$p(\mathcal{D}|\theta) = \prod_{i=1}^{N} \sum_{y}^{V} \left(\sum_{s_i^1}^{V} \cdots \sum_{s_i^{C_i}}^{V} \right) p(T_{i(y\underline{s})} \mid \underline{x_i}, \mathcal{M}, \gamma) \prod_{c}^{C_i} \prod_{a}^{A_i^c} p(r_i^{ca}|s_i^c, \alpha)$$

During the E step, we will use analogous variables $\tau_{i(y\underline{s})} \in [0, 1]$ to represent the conditional probabilities of $T_{i(y\underline{s})}$.

Below, we derive the E-step and the M-step. For clarity, we first express the complete-data likelihood function (and the complete data log-likelihood) and the expected complete log-likelihood, which we then use to determine solutions for the E-step and the M-step.

Complete data likelihood function

Here, we assume that we have the observed values of every $T_{i(y\underline{s})}$. The $T_{i(y\underline{s})}$ in the exponent is an observed 0 or 1, while the $p(T_{i(y\underline{s})} = 1 \mid \ldots)$ is still a prior probability to compute. (That prior does use its parent variables $\underline{x_i}$, but importantly, doesn't use r_i^{ca}.)

$$p(\mathcal{D}, Z \mid \theta) = \prod_{i=1}^{N} \prod_{y}^{V} \left(\prod_{s_i^1=1}^{V} \cdots \prod_{s_i^{C_i}=1}^{V} \right) \left(p(T_{i(y\underline{s})} = 1 \mid \underline{x_i}, \mathcal{M}, \gamma) \prod_{c}^{C_i} \prod_{a}^{A_i^c} p(r_i^{ca} | s_i^c, \alpha) \right)^{T_{i(y\underline{s})}}$$

Complete data log-likelihood

$$\ell(\mathcal{D}, Z | \theta) = \sum_{i=1}^{N} \sum_{y}^{V} \left(\sum_{s_i^1}^{V} \cdots \sum_{s_i^{C_i}}^{V} \right) T_{i(y\underline{s})} \left(\log p(T_{i(y\underline{s})} = 1 \mid \underline{x_i}, \mathcal{M}, \gamma) + \sum_{c}^{C_i} \sum_{a}^{A_i^c} \log p(r_i^{ca} | s_i^c, \alpha) \right)$$

Expected value of the complete data log-likelihood

$$\begin{aligned} \mathbb{E}_Z[\ell(\mathcal{D}, Z | \theta)] &= \sum_{i=1}^{N} \sum_{y}^{V} \left(\sum_{s_i^1}^{V} \cdots \sum_{s_i^{C_i}}^{V} \right) \mathbb{E}_Z[T_{i(y\underline{s})}] \left(\log p(T_{i(y\underline{s})} = 1 \mid \underline{x_i}, \mathcal{M}, \gamma) + \sum_{c}^{C_i} \sum_{a}^{A_i^c} \log p(r_i^{ca} | s_i^c, \alpha) \right) \\ &= \sum_{i=1}^{N} \sum_{y}^{V} \left(\sum_{s_i^1}^{V} \cdots \sum_{s_i^{C_i}}^{V} \right) \tau_{i(y\underline{s})} \left(\log p(T_{i(y\underline{s})} = 1 \mid \underline{x_i}, \mathcal{M}, \gamma) + \sum_{c}^{C_i} \sum_{a}^{A_i^c} \log p(r_i^{ca} | s_i^c, \alpha) \right) \end{aligned}$$

In the E step, we update the values τ.

B.1 E step

Here, we need an expected value for the latent variables conditioned on observed variables and θ. Define:

$$\begin{aligned} \tau_{i(y\underline{s})} &= p(T_{i(y\underline{s})} = 1 \mid \mathcal{D}, \theta) \\ &= p(y_i = y, \underline{s_i} = \underline{s} \mid \underline{r_i}, \underline{x_i}, \mathcal{M}, \gamma, \alpha) \end{aligned}$$

Using Bayes' rule, we have that the update for $\tau_{i(y\underline{s})}$ is:

$$\begin{aligned} &= \frac{p(\underline{r_i} \mid y_i = y, \underline{s_i} = \underline{s}, \underline{x_i}, \mathcal{M}, \gamma, \alpha) p(y_i = y, \underline{s_i} = \underline{s} \mid \underline{x_i}, \mathcal{M}, \gamma, \alpha)}{p(\underline{r_i} \mid \underline{x_i}, \mathcal{M}, \gamma, \alpha)} \\ &= \frac{p(y_i = y, \underline{s_i} = \underline{s} \mid \underline{x_i}, \mathcal{M}, \gamma) p(\underline{r_i} \mid \underline{s_i} = \underline{s}, \alpha)}{p(\underline{r_i} \mid \underline{x_i}, \mathcal{M}, \gamma, \alpha)} \\ &= \frac{p(y_i = y \mid \underline{x_i}, \mathcal{M}) p(\underline{s_i} = \underline{s} \mid y_i = y, \gamma) p(\underline{r_i} \mid \underline{s_i} = \underline{s}, \alpha)}{\sum_{y'=1}^{V} \left(\sum_{s'^1_i}^{V} \cdots \sum_{s'^{C_i}_i}^{V} \right) p(y_i = y' \mid \underline{x_i}, \mathcal{M}) p(\underline{s_i} = \underline{s'} \mid y_i = y', \gamma) p(\underline{r_i} \mid \underline{s_i} = \underline{s'}, \alpha)}. \end{aligned}$$

The numerator is simply the likelihood of a fully observed instance (a tweet and its labels, with the specified setting of latent variables), while the denominator ensures that the distribution sums to 1.

B.2 M step for γ

Recall that:

$$\begin{aligned}\mathbb{E}_Z[\ell(\mathcal{D}, Z|\theta)] &= \sum_{i=1}^{N}\sum_{y}^{V}\left(\sum_{s_i^1}^{V}\cdots\sum_{s_i^{C_i}}^{V}\right)\tau_{i(y\underline{s})}\left(\log p(T_{i(y\underline{s})} \mid \underline{x_i}, \mathcal{M}, \gamma) + \sum_{c}^{C_i}\sum_{a}^{A_i^c}\log p(r_i^{ca}|s_i^c, \alpha)\right)\\ &= \sum_{i=1}^{N}\sum_{y}^{V}\left(\sum_{s_i^1}^{V}\cdots\sum_{s_i^{C_i}}^{V}\right)\tau_{i(y\underline{s})}\left(\log \mathcal{M}_y(\underline{x_i}) + (\sum_{c}^{C_i}\log\gamma_{ys}^c) + \sum_{c}^{C_i}\sum_{a}^{A_i^c}\log\alpha_{sr}^a\right). \quad (3)\end{aligned}$$

Recall that γ_{ys}^c denotes the matrix entry describing $p(s_i^c = s \mid y_i = y)$. We have a constraint that $\sum_{s'}^{V}\gamma_{ys'}^c = 1$. (That is, within the cth matrix of γ, row y must sum to 1.) Collect terms from $\mathbb{E}_Z[\ell(\mathcal{D}, Z|\theta)]$ that depend on a particular matrix entry γ_{ys}^c into one expression $J(\gamma_{ys}^c)$, together with the Lagrange multiplier term from the constraint.

$$\begin{aligned}J(\gamma_{ys}^c) &= \left(\sum_{i=1}^{N}\sum_{y'}^{V}(\sum_{s'^1_i}^{V}\cdots\sum_{s'^{C_i}_i}^{V})\sum_{c'}^{C_i}\tau_{iy'\underline{s}'}\log\gamma_{y's'}^{c'}\right) - \lambda(\sum_{s'}^{V}\gamma_{ys'}^c - 1)\\ &= \left(\sum_{i=1}^{N}(\sum_{s'^1_i}^{V}\cdots\sum_{s'^{C_i}_i}^{V})\tau_{i(y\underline{s}')}\log\gamma_{ys'}^c\right) - \lambda(\gamma_{ys}^c)\end{aligned}$$

From the summations over c' and y', only the term with the desired c and y depends on γ_{ys}^c. In the summations over the values of $\underline{s}$, only the cth summation pertains to γ_{ys}^c (i.e., γ_{ys}^c appears only when $s'^c_i = s$). However, the other components of $\underline{s'}$ are latent variables whose probability we need to sum over.

$$\begin{aligned}&= \left(\sum_{i=1}^{N}(\sum_{s'^1_i}^{V}\ldots\text{[except component } c]\ldots\sum_{s'^{C_i}_i}^{V})\tau_{i(y\underline{s}')}\log\gamma_{ys}^c\right) - \lambda(\gamma_{ys}^c)\\ &= \log\gamma_{ys}^c\left(\sum_{i=1}^{N}(\sum_{s'^1_i}^{V}\ldots\text{[except component } c]\ldots\sum_{s'^{C_i}_i}^{V})\tau_{i(y\underline{s})'}\right) - \lambda(\gamma_{ys}^c)\\ &= \log\gamma_{ys}^c\,(\text{Weighted number of tweets with } y_i = y \text{ and } s_i^c = s) - \lambda(\gamma_{ys}^c)\end{aligned}$$

$$\frac{\delta\ell}{\delta\gamma_{ys}^c} = J'(\gamma_{ys}^c) = \frac{(\text{Weighted number of tweets with } y_i = y \text{ and } s_i^c = s)}{\gamma_{ys}^c} - \lambda$$

Note that "weighted" always means "weighted using the current assignment probabilities $\tau_{i(y\underline{s})}$."

Set $J'(\gamma^c_{ys})$ to 0 to get:

$$\gamma^c_{ys} = \frac{\text{(Weighted number of tweets with } y_i = y \text{ and } s^c_i = s\text{)}}{\lambda}.$$

Go back to the constraint equation and plug in expression above for each γ:

$$\sum_{s'}^{V} \gamma^c_{ys'} = 1$$

$$\sum_{s'}^{V} \left(\frac{\text{(Weighted number of tweets with } y_i = y \text{ and } s^c_i = s'\text{)}}{\lambda}\right) = 1$$

$$\lambda = \text{(Weighted number of tweets with } y_i = y \text{ (and any value for } s^c_i\text{)))}$$

So,

$$\gamma^c_{ys} = \frac{\text{(Weighted number of tweets with } y_i = y \text{ and } s^c_i = s\text{)}}{\text{(Weighted number of tweets with } y_i = y \text{ (and any value for } s^c_i\text{)}}.$$

B.3 M step for α

Recall that α^a_{sr} is the matrix entry describing $p(r^{ca}_i = r \mid s^c_i = s)$—that is, the probability that the ath annotator writes r when the tweet (as they saw it in context c) had a context-specific label of $s^c_i = s$. Note that each annotation r^{ca}_i takes place in a particular known context c, but α does not depend on c. To keep track—while we move terms around—of the context associated with each annotation, we re-expand $\log p(r^{ca}_i | s^c_i, \alpha)$ to $\alpha^a_{sr}\delta(s^c_i = s)$.

Referring back to Eq. (3), collect terms that depend on α^a_{sr}. Also add the normalization constraint that $\sum_{r'}^{V} \alpha^a_{sr'} = 1$.

$$\mathbb{E}_T[\ell(\mathcal{D}, Z|\theta)] = \sum_{i=1}^{N} \sum_{y}^{V} \left(\sum_{s_i^1}^{V} \cdots \sum_{s_i^{C_i}}^{V} \right) \tau_{iy\underline{s}} \left(\log \mathcal{M}_y(\underline{x_i}) + (\sum_{c}^{C_i} \log \gamma_{ys}^c) + \sum_{c}^{C_i} \sum_{a}^{A_i^c} \delta(s_i^c = s) \log \alpha_{sr}^a \right)$$

$$\begin{aligned}
J(\alpha_{sr}^a) &= \left(\sum_{i=1}^{N} \sum_{y}^{V} (\sum_{{s'}_i^1}^{V} \cdots \sum_{{s'}_i^{C_i}}^{V}) \sum_{c}^{C_i} \sum_{a'}^{A_i^c} \tau_{i(y\underline{s})} \delta(s_i^c = s) \log \alpha_{sr}^{a'} \right) - \lambda(\sum_{r'}^{V} \alpha_{sr'}^a - 1) \\
&= \left(\sum_{i=1}^{N} \sum_{y}^{V} \sum_{c}^{C_i} (\sum_{{s'}_i^1}^{V} \cdots \sum_{{s'}_i^{C_i}}^{V}) \tau_{i(y\underline{s})} \delta(s_i^c = s) \log \alpha_{sr}^a \right) - \lambda(\alpha_{sr}^a) \\
&= \left(\sum_{i=1}^{N} \sum_{y}^{V} \sum_{c}^{C_i} (\sum_{{s'}_i^1}^{V} \ldots \text{[except component } c] \ldots \sum_{{s'}_i^{C_i}}^{V}) \tau_{i(y\underline{s})} \log \alpha_{sr}^a \right) - \lambda(\alpha_{sr}^a) \\
&= \log \alpha_{sr}^a \left(\sum_{i=1}^{N} \sum_{y}^{V} \sum_{c}^{C_i} (\sum_{{s'}_i^1}^{V} \ldots \text{[except component } c] \ldots \sum_{{s'}_i^{C_i}}^{V}) \tau_{i(y\underline{s})} \right) - \lambda(\alpha_{sr}^a) \\
&= \log \alpha_{sr}^a (\text{Weighted number of annotations with value } r \text{ by annotator } a \text{ having } s_i^c = s) - \lambda(\alpha_{sr}^a)
\end{aligned}$$

$$\frac{\delta \ell}{\delta \alpha_{sr}^a} = J'(\alpha_{sr}^a) = \frac{(\text{Weighted number of annotations with value } r \text{ by annotator } a \text{ having } s_i^c = s)}{\alpha_{sr}^a} - \lambda$$

Notice that for γ, we were counting tweets in a particular context (and looking at their configuration of latent variables). For α here, the sum is over tweets + contexts; we are counting all annotations made by a particular annotator (and looking at the s_i for the context in which the annotation took place).

Set $J'(\alpha_{sr}^a)$ to 0 to get:

$$\alpha_{sr}^a = \frac{(\text{Weighted number of annotations with value } r \text{ by annotator } a \text{ in which } s_i^c = s)}{\lambda}.$$

The constraint equation works just like it did for γ:

$$\sum_{r'}^{V} \alpha_{sr'}^a = 1$$

$$\frac{1}{\lambda} \sum_{r'}^{V} (\text{Weighted number of annotations with value } r' \text{ by annotator } a \text{ in which } s_i^c = s) = 1$$

$$\lambda = (\text{Weighted number of annotations by annotator } a \text{ in which } s_i^c = s).$$

Finally,

$$\alpha_{sr}^a = \frac{(\text{Weighted number of annotations with value } r \text{ by annotator } a \text{ in which } s_i^c = s)}{(\text{Weighted number of annotations by annotator } a \text{ in which } s_i^c = s)}.$$

B.4 Computing labels to use for classifiers

Using Raykar et al.'s suggestion, we decide to test a variety of classifiers. In order to do so, we must recover $\mathbb{E}_Z[y_i = k]$, the expected likelihood of y_i taking on the particular value k. Starting from $\tau_{i(k\underline{s})}$, which is computed during the E step and is defined as $p(y_i = k, \underline{s_i} = \underline{s}|\mathcal{D}, \theta)$, we marginalize out $\underline{s_i}$:

$$\mathbb{E}_Z[y_i = k] = p(y_i = k \mid \mathcal{D}, \theta) = \sum_{s_i^1}^{V} \cdots \sum_{s_i^{C_i}}^{V} \tau_{i(k\underline{s})}$$

We can then use these probability values to train any multi-class classifier we wish by performing sampling based on the obtained weights. During model testing and evaluation, we observed that the number of samples per item did not significantly impact model performance. Therefore, for all results presented in the paper, we simply fixed the number of samples per item to 10.

C EM Algorithm Debugging

In order to ensure the algorithm, as coded, correctly learns parameters, we take two steps. First, we ensure that the log-likelihood of the model decreases on every iteration. Second, we developed simulations to ensure that we can recover known parameters for simulated data. Simulations suggested that the model was easily able to uncover known parameters for γ across a variety of tested values and conditions similar to those that generated our data (i.e. with the same numbers of tweets, context conditions and annotators). However, we observe that the model does struggle to recover some parameterizations of α; we expect the cause of this is a combination of the randomness induced by the data generating process and the sheer number of α parameters in the model ($9|A|$). Future work might consider how to limit the number of parameters in α by, e.g., assuming annotators are a mixture over a smaller number of prototypical annotation styles.

D Hyperparameter Optimization

For all hyperparameter tuning, we use a rough grid search approach, testing various settings on performance on the development set (focusing on both Log-Loss and Average F1). As our intention was to focus on the impact of different labeling schemes, our goal in hyperparameter tuning was simply to find reasonable and, more importantly, consistent models that we could use to address the impact of the labeling structure (and our model ablations). As noted in the paper, future work will focus on improving our model and

For hyperparameter tuning of the baseline models, we tune parameters for the maximum depth of the tree and the number of estimators. For our model, our model ablations and each of the baselines, a maximum depth of 30 and 3000 estimators were used, as results stabilized around these numbers.

Development and validation differed in that the development data consisted of only registered Democrats and Republicans, while the validation data also generalized to non-labeled Democrats and Republicans. Because of this, we allowed the baseline models to "cheat" by setting class weights for the Random Forest to the ratio of true label counts in the validation data versus the training data. Note we did *not* do this for ConStance or its ablations, setting the prior on y via tuning on the test set.

To tune ConStance and the tested ablations, we considered varying the initializations of α and γ, providing Dirichlet priors on α and γ and by varying a prior on y. In the end, initialization of α and γ made little difference, we elected not to provide the model with any priors on α or γ. However, we did set the prior on y to $[.495, .01, .495]$ for the "Trump", "Neutral" and "Clinton" labels, respectively. Like the class weights for the baseline Random Forest models, this prior urged the model away from selecting the "Neutral" option, which was far less prevalent in the test and validation data than it was in the annotations. This, of course, is because annotating with full context allowed for a significantly more discriminative take on the support of Twitter users as compared to the context seen by the AMT workers.

For the ablations, models performed better with a different prior on y ($[.45, .1, .45]$), we therefore use this for validation.