Event Discovery in Social Media Feeds

Edward Benson, Aria Haghighi, and Regina Barzilay
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
{eob, aria42, regina}@csail.mit.edu

Abstract

We present a novel method for record extraction from social streams such as Twitter. Unlike typical extraction setups, these environments are characterized by short, one sentence messages with heavily colloquial speech. To further complicate matters, individual messages may not express the full relation to be uncovered, as is often assumed in extraction tasks. We develop a graphical model that addresses these problems by learning a latent set of records and a record-message alignment simultaneously; the output of our model is a set of canonical records, the values of which are consistent with aligned messages. We demonstrate that our approach is able to accurately induce event records from Twitter messages, evaluated against events from a local city guide. Our method achieves significant error reduction over baseline methods.[1]

1 Introduction

We propose a method for discovering event records from social media feeds such as Twitter. The task of extracting event properties has been well studied in the context of formal media (e.g., newswire), but data sources such as Twitter pose new challenges. Social media messages are often short, make heavy use of colloquial language, and require situational context for interpretation (see examples in Figure 1). Not all properties of an event may be expressed in a single message, and the mapping between messages and canonical event records is not obvious.

Twitter Messages

Seated at @carnegiehall waiting for @CraigyFerg's show to begin
RT @leerader : getting REALLY stoked for #CraigyAtCarnegie sat night. Craig, , want to join us for dinner at the pub across the street? 5pm, be there!
@DJPaulyD absolutely killed it at Terminal 5 last night.
@DJPaulyD : DJ Pauly D Terminal 5 NYC Insanity ! #ohyeah @keadour @kellaferr24
Craig, nice seeing you at #noelnight this weekend @becksdavis!

Records

Artist	Venue
Craig Ferguson	Carnegie Hall
DJ Pauly D	Terminal 5

Figure 1: Examples of Twitter messages, along with automatically extracted records.

These properties of social media streams make existing extraction techniques significantly less effective. Despite these challenges, this data exhibits an important property that makes learning amenable: the multitude of messages referencing the same event.

Our goal is to induce a comprehensive set of event records given a seed set of example records, such as a city event calendar table. While such resources are widely available online, they are typically high precision, but low recall. Social media is a natural place to discover new events missed by curation, but mentioned online by someone planning to attend.

We formulate our approach as a structured graphical model which simultaneously analyzes individual messages, clusters them according to event, and induces a canonical value for each event property. At the message level, the model relies on a conditional random field component to extract field values such

[1]Data and code available at `http://groups.csail.mit.edu/rbg/code/twitter`

Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, pages 389–398,
Portland, Oregon, June 19-24, 2011.

as location of the event and artist name. We bias local decisions made by the CRF to be consistent with canonical record values, thereby facilitating consistency within an event cluster. We employ a factor-graph model to capture the interaction between each of these decisions. Variational inference techniques allow us to effectively and efficiently make predictions on a large body of messages.

A seed set of example records constitutes our only source of supervision; we do not observe alignment between these seed records and individual messages, nor any message-level field annotation. The output of our model consists of an event-based clustering of messages, where each cluster is represented by a single multi-field record with a canonical value chosen for each field.

We apply our technique to construct entertainment event records for the city calendar section of `NYC.com` using a stream of Twitter messages. Our method yields up to a 63% recall against the city table and up to 85% precision evaluated manually, significantly outperforming several baselines.

2 Related Work

A large number of information extraction approaches exploit redundancy in text collections to improve their accuracy and reduce the need for manually annotated data (Agichtein and Gravano, 2000; Yangarber et al., 2000; Zhu et al., 2009; Mintz et al., 2009a; Yao et al., 2010b; Hasegawa et al., 2004; Shinyama and Sekine, 2006). Our work most closely relates to methods for multi-document information extraction which utilize redundancy in input data to increase the accuracy of the extraction process. For instance, Mann and Yarowsky (2005) explore methods for fusing extracted information across multiple documents by performing extraction on each document independently and then merging extracted relations by majority vote. This idea of consensus-based extraction is also central to our method. However, we incorporate this idea into our model by simultaneously clustering output and labeling documents rather than performing the two tasks in serial fashion. Another important difference is inherent in the input data we are processing: it is not clear a priori which extraction decisions should agree with each other. Identifying messages that refer to the same event is a large part of our challenge.

Our work also relates to recent approaches for relation extraction with *distant supervision* (Mintz et al., 2009b; Bunescu and Mooney, 2007; Yao et al., 2010a). These approaches assume a database and a collection of documents that verbalize some of the database relations. In contrast to traditional supervised IE approaches, these methods do not assume that relation instantiations are annotated in the input documents. For instance, the method of Mintz et al. (2009b) induces the mapping automatically by bootstrapping from sentences that directly match record entries. These mappings are used to learn a classifier for relation extraction. Yao et al. (2010a) further refine this approach by constraining predicted relations to be consistent with entity types assignment. To capture the complex dependencies among assignments, Yao et al. (2010a) use a factor graph representation. Despite the apparent similarity in model structure, the two approaches deal with various types of uncertainties. The key challenge for our method is modeling message to record alignment which is not an issue in the previous set up.

Finally, our work fits into a broader area of text processing methods designed for social-media streams. Examples of such approaches include methods for conversation structure analysis (Ritter et al., 2010) and exploration of geographic language variation (Eisenstein et al., 2010) from Twitter messages. To our knowledge no work has yet addressed record extraction from this growing corpus.

3 Problem Formulation

Here we describe the key latent and observed random variables of our problem. A depiction of all random variables is given in Figure 2.

Message (x): Each message x is a single posting to Twitter. We use x^j to represent the j^{th} token of x, and we use $\mathbf{x}$ to denote the entire collection of messages. Messages are always observed during training and testing.

Record (R): A record is a representation of the canonical properties of an event. We use R_i to denote the i^{th} record and R_i^ℓ to denote the value of the ℓ^{th} property of that record. In our experiments, each record R_i is a tuple $\langle R_i^1, R_i^2 \rangle$ which represents that

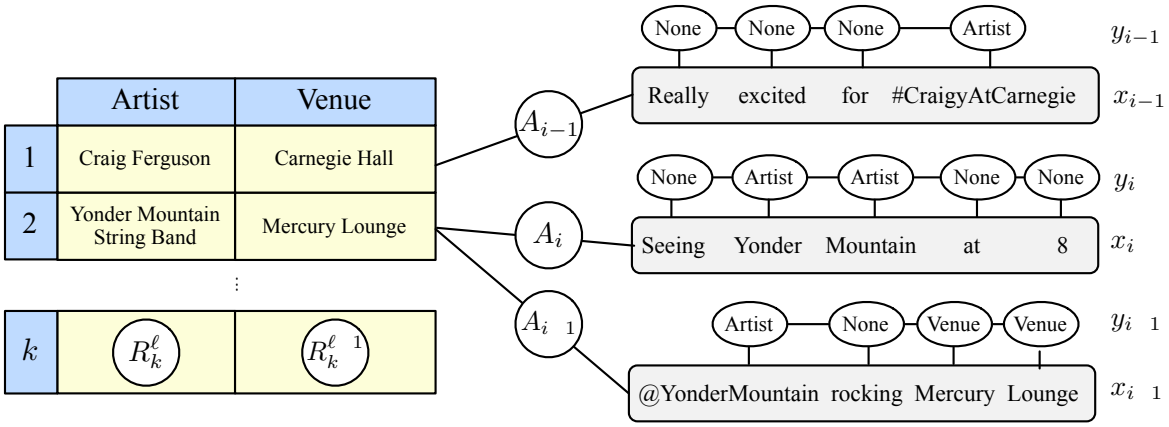

Figure 2: The key variables of our model. A collection of K latent records R_k, each consisting of a set of L properties. In the figure above, R_1^1 ="Craig Ferguson" and R_1^2 ="Carnegie Hall." Each tweet x_i is associated with a labeling over tokens y_i and is aligned to a record via the A_i variable. See Section 3 for further details.

record's values for the schema ⟨ARTIST, VENUE⟩. Throughout, we assume a known fixed number K of records $R_1, \ldots, R_K$, and we use $\mathbf{R}$ to denote this collection of records. For tractability, we consider a finite number of possibilities for each R_k^ℓ which are computed from the input $\mathbf{x}$ (see Section 5.1 for details). Records are observed during training and latent during testing.

Message Labels (y): We assume that each message has a sequence labeling, where the labels consist of the record fields (e.g., ARTIST and VENUE) as well as a NONE label denoting the token does not correspond to any domain field. Each token x^j in a message has an associated label y^j. Message labels are always latent during training and testing.

Message to Record Alignment (A): We assume that each message is aligned to some record such that the event described in the message is the one represented by that record. Each message x_i is associated with an alignment variable A_i that takes a value in $\{1, \ldots, K\}$. We use $\mathbf{A}$ to denote the set of alignments across all x_i. Multiple messages can and do align to the same record. As discussed in Section 4, our model will encourage tokens associated with message labels to be "similar" to corresponding aligned record values. Alignments are always latent during training and testing.

4 Model

Our model can be represented as a factor graph which takes the form,

$$P(R, A, y|x) \propto$$
$$\left(\prod_i \phi_{SEQ}(x_i, y_i)\right) \quad \text{(Seq. Labeling)}$$
$$\left(\prod_\ell \phi_{UNQ}(R^\ell)\right) \quad \text{(Rec. Uniqueness)}$$
$$\left(\prod_{i,\ell} \phi_{POP}(x_i, y_i, R^\ell_{A_i})\right) \quad \text{(Term Popularity)}$$
$$\left(\prod_i \phi_{CON}(x_i, y_i, R_{A_i})\right) \quad \text{(Rec. Consistency)}$$

where $\mathbf{R}^\ell$ denotes the sequence $R_1^\ell, \ldots, R_K^\ell$ of record values for a particular domain field ℓ. Each of the potentials takes a standard log-linear form:

$$\phi(z) = \theta^T f(z)$$

where θ are potential-specific parameters and $f(\cdot)$ is a potential-specific feature function. We describe each potential separately below.

4.1 Sequence Labeling Factor

The sequence labeling factor is similar to a standard sequence CRF (Lafferty et al., 2001), where the potential over a message label sequence decomposes

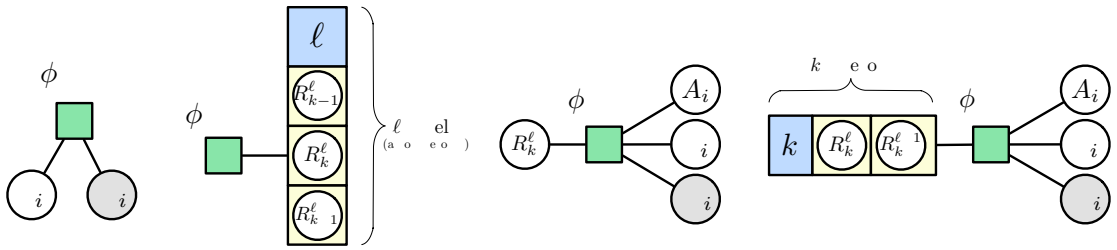

Figure 3: Factor graph representation of our model. Circles represent variables and squares represent factors. For readability, we depict the graph broken out as a set of templates; the full graph is the combination of these factor templates applied to each variable. See Section 4 for further details.

over pairwise cliques:

$$\phi_{SEQ}(x, y) = \exp\{\theta_{SEQ}^T f_{SEQ}(x, y)\}$$
$$= \exp\left\{\theta_{SEQ}^T \sum_j f_{SEQ}(x, y^j, y^{j+1})\right\}$$

This factor is meant to encode the typical message contexts in which fields are evoked (e.g. *going to see X tonight*). Many of the features characterize how likely a given token label, such as ARTIST, is for a given position in the message sequence conditioning arbitrarily on message text context.

The feature function $f_{SEQ}(x, y)$ for this component encodes each token's identity; word shape[2]; whether that token matches a set of regular expressions encoding common emoticons, time references, and venue types; and whether the token matches a bag of words observed in artist names (scraped from Wikipedia; 21,475 distinct tokens from 22,833 distinct names) or a bag of words observed in New York City venue names (scraped from `NYC.com`; 304 distinct tokens from 169 distinct names).[3] The only edge feature is label-to-label.

4.2 Record Uniqueness Factor

One challenge with Twitter is the so-called echo chamber effect: when a topic becomes popular, or "trends," it quickly dominates the conversation online. As a result some events may have only a few referent messages while other more popular events may have thousands or more. In such a circumstance, the messages for a popular event may collect to form multiple identical record clusters. Since we fix the number of records learned, such behavior inhibits the discovery of less talked-about events. Instead, we would rather have just two records: one with two aligned messages and another with thousands. To encourage this outcome, we introduce a potential that rewards fields for being unique across records.

The uniqueness potential $\phi_{UNQ}(\mathbf{R}^\ell)$ encodes the preference that each of the values $R^\ell, \ldots, R^\ell_K$ for each field ℓ do not overlap textually. This factor factorizes over pairs of records:

$$\phi_{UNQ}(\mathbf{R}^\ell) = \prod_{k \neq k'} \phi_{UNQ}(R^\ell_k, R^\ell_{k'})$$

where R^ℓ_k and $R^\ell_{k'}$ are the values of field ℓ for two records R_k and $R_{k'}$. The potential over this pair of values is given by:

$$\phi_{UNQ}(R^\ell_k, R^\ell_{k'}) = \exp\{-\theta_{SIM}^T f_{SIM}(R^\ell_k, R^\ell_{k'})\}$$

where f_{SIM} is computes the likeness of the two values at the token level:

$$f_{SIM}(R^\ell_k, R^\ell_{k'}) = \frac{|R^\ell_k \cap R^\ell_{k'}|}{\max(|R^\ell_k|, |R^\ell_{k'}|)}$$

This uniqueness potential does not encode any preference for record values; it simply encourages each field ℓ to be distinct across records.

4.3 Term Popularity Factor

The term popularity factor ϕ_{POP} is the first of two factors that guide the clustering of messages. Because speech on Twitter is colloquial, we would like these clusters to be amenable to many variations of the canonical record properties that are ultimately learned. The ϕ_{POP} factor accomplishes this by representing a lenient compatibility score between a

[2] e.g.: xxx, XXX, Xxx, or other

[3] These are just features, not a filter; we are free to extract any artist or venue regardless of their inclusion in this list.

message x, its labels y, and some candidate value v for a record field (e.g., *Dave Matthews Band*).

This factor decomposes over tokens, and we align each token x^j with the best matching token v^k in v (e.g., *Dave*). The token level sum is scaled by the length of the record value being matched to avoid a preference for long field values.

$$\phi_{POP}(x, y, R^\ell_A = v) = \sum_j \max_k \frac{\phi_{POP}(x^j, y^j, R^\ell_A = v^k)}{|v|}$$

This token-level component may be thought of as a compatibility score between the labeled token x^j and the record field assignment $R^\ell_A = v$. Given that token x^j aligns with the token v^k, the token-level component returns the sum of three parts, subject to the constraint that $y^j = \ell$:

- $IDF(x^j)\mathbb{I}[x^j = v^k]$, an equality indicator between tokens x^j and v^k, scaled by the inverse document frequency of x^j
- $\alpha IDF(x^j)\left(\mathbb{I}[x^{j-1} = v^{k-1}] + \mathbb{I}[x^{j+1} = v^{k+1}]\right)$, a small bonus of $\alpha = 0.3$ for matches on adjacent tokens, scaled by the IDF of x^j
- $\mathbb{I}[x^j = v^k$ and x contains $v]/|v|$, a bonus for a complete string match, scaled by the size of the value. This is equivalent to this token's contribution to a complete-match bonus.

4.4 Record Consistency Factor

While the uniqueness factor discourages a flood of messages for a single event from clustering into multiple event records, we also wish to discourage messages from multiple events from clustering into the same record. When such a situation occurs, the model may either resolve it by changing inconsistent token labelings to the NONE label or by reassigning some of the messages to a new cluster. We encourage the latter solution with a record consistency factor ϕ_{CON}.

The record consistency factor is an indicator function on the field values of a record being present and labeled correctly in a message. While the popularity factor encourages agreement on a per-label basis, this factor influences the joint behavior of message labels to agree with the aligned record. For a given record, message, and labeling, $\phi_{CON}(x, y, R_A) = 1$ if $\phi_{POP}(x, y, R^\ell_A) > 0$ for all ℓ, and 0 otherwise.

4.5 Parameter Learning

The weights of the CRF component of our model, θ_{SEQ}, are the only weights learned at training time, using a distant supervision process described in Section 6. The weights of the remaining three factors were hand-tuned[4] using our training data set.

5 Inference

Our goal is to predict a set of records $\mathbf{R}$. Ideally we would like to compute $P(\mathbf{R}|\mathbf{x})$, marginalizing out the nuisance variables $\mathbf{A}$ and $\mathbf{y}$. We approximate this posterior using variational inference.[5] Concretely, we approximate the full posterior over latent variables using a mean-field factorization:

$$P(\mathbf{R}, \mathbf{A}, \mathbf{y}|\mathbf{x}) \approx Q(\mathbf{R}, \mathbf{A}, \mathbf{y}) = \left(\prod_{k=1}^{K}\prod_{\ell} q(R^\ell_k)\right)\left(\prod_{i=1}^{n} q(A_i)q(y_i)\right)$$

where each variational factor $q(\cdot)$ represents an approximation of that variable's posterior given observed random variables. The variational distribution $Q(\cdot)$ makes the (incorrect) assumption that the posteriors amongst factors are independent. The goal of variational inference is to set factors $q(\cdot)$ to optimize the variational objective:

$$\min_{Q(\cdot)} KL(Q(\mathbf{R}, \mathbf{A}, \mathbf{y}) \| P(\mathbf{R}, \mathbf{A}, \mathbf{y}|\mathbf{x}))$$

We optimize this objective using coordinate descent on the $q(\cdot)$ factors. For instance, for the case of $q(y_i)$ the update takes the form:

$$q(y_i) \leftarrow \mathbb{E}_{Q/q(y_i)} \log P(\mathbf{R}, \mathbf{A}, \mathbf{y}|\mathbf{x})$$

where $Q/q(y_i)$ denotes the expectation under all variables except y_i. When computing a mean field update, we only need to consider the potentials involving that variable. The complete updates for each of the kinds of variables (y, A, and R^ℓ) can be found in Figure 4. We briefly describe the computations involved with each update.

$q(y)$ **update**: The $q(y)$ update for a single message yields an implicit expression in terms of pairwise cliques in y. We can compute arbitrary

[4] Their values are: $\theta_{UNQ} = -10$, $\theta^{\text{Phrase}}_{POP} = 5$, $\theta^{\text{Token}}_{POP} = 10$, $\theta_{CON} = 2e8$

[5] See Liang and Klein (2007) for an overview of variational techniques.

Message labeling update:

$$\begin{aligned}\ln q(y) &\propto \left\{\mathbb{E}_{Q/q(y)} \ln \phi_{SEQ}(x,y) + \ln\left[\phi_{POP}(x,y,R^\ell_A)\phi_{CON}(x,y,R_A)\right]\right\} \\ &= \ln \phi_{SEQ}(x,y) + \mathbb{E}_{Q/q(y)} \ln\left[\phi_{POP}(x,y,R^\ell_A)\phi_{CON}(x,y,R_A)\right] \\ &= \ln \phi_{SEQ}(x,y) + \sum_{z,v,\ell} q(A=z)q(y^j=\ell)q(R^\ell_z=v)\ln\left[\phi_{POP}(x,y,R^\ell_z=v)\phi_{CON}(x,y,R^\ell_z=v)\right]\end{aligned}$$

Mention record alignment update:

$$\begin{aligned}\ln q(A=z) &\propto \mathbb{E}_{Q/q(A)}\left\{\ln \phi_{SEQ}(x,y) + \ln\left[\phi_{POP}(x,y,R^\ell_A)\phi_{CON}(x,y,R_A)\right]\right\} \\ &\propto \mathbb{E}_{Q/q(A)}\left\{\ln\left[\phi_{POP}(x,y,R^\ell_A)\phi_{CON}(x,y,R_A)\right]\right\} \\ &= \sum_{z,v,\ell} q(R^\ell_z=v)\left\{\ln\left[\phi_{POP}(x,y,R^\ell_z=v)\phi_{CON}(x,y,R^\ell_z=v)\right]\right\} \\ &= \sum_{z,v,\ell} q(R^\ell_z=v)q(y^j_i=\ell)\ln\left[\phi_{POP}(x,y,R^\ell_z=v)\phi_{CON}(x,y,R^\ell_z=v)\right]\end{aligned}$$

Record Field update:

$$\begin{aligned}\ln q(R^\ell_k=v) &\propto \mathbb{E}_{Q/q(R^\ell_k)}\left\{\sum_{k'}\ln\phi_{UNQ}(R^\ell_{k'},v) + \sum_i \ln\left[\phi_{POP}(x_i,y_i,v)\phi_{CON}(x_i,y_i,v)\right]\right\} \\ &= \sum_{k'\neq k,v'}\Bigg(q(R^\ell_{k'}=v')\ln\phi_{UNQ}(v,v') \\ &+ \sum_i q(A_i=k)\sum_j q(y^j_i=\ell)\ln\left[\phi_{POP}(x,y,R^\ell_z=v,j)\phi_{CON}(x,y,R^\ell_z=v,j)\right]\Bigg)\end{aligned}$$

Figure 4: The variational mean-field updates used during inference (see Section 5). Inference consists of performing updates for each of the three kinds of latent variables: message labels (y), record alignments (A), and record field values (R^ℓ). All are relatively cheap to compute except for the record field update $q(R^\ell_k)$ which requires looping potentially over all messages. Note that at inference time all parameters are fixed and so we only need to perform updates for latent variable factors.

marginals for this distribution by using the forwards-backwards algorithm on the potentials defined in the update. Therefore computing the $q(y)$ update amounts to re-running forward backwards on the message where there is an expected potential term which involves the belief over other variables. Note that the popularity and consensus potentials (ϕ_{POP} and ϕ_{CON}) decompose over individual message tokens so this can be tractably computed.

$q(A)$ **update:** The update for individual record alignment reduces to being log-proportional to the expected popularity and consensus potentials.

$q(R^\ell_k)$ **update:** The update for the record field distribution is the most complex factor of the three. It requires computing expected similarity with other record field values (the ϕ_{UNQ} potential) and looping over all messages to accumulate a contribution from each, weighted by the probability that it is aligned to the target record.

5.1 Initializing Factors

Since a uniform initialization of all factors is a saddle-point of the objective, we opt to initialize the $q(y)$ factors with the marginals obtained using just the CRF parameters, accomplished by running forwards-backwards on all messages using only the

ϕ_{SEQ} potentials. The $q(R)$ factors are initialized randomly and then biased with the output of our baseline model. The $q(A)$ factor is initialized to uniform plus a small amount of noise.

To simplify inference, we pre-compute a finite set of values that each R_k^ℓ is allowed to take, conditioned on the corpus. To do so, we run the CRF component of our model (ϕ_{SEQ}) over the corpus and extract, for each ℓ, all spans that have a token-level probability of being labeled ℓ greater than $\lambda = 0.1$. We further filter this set down to only values that occur at least twice in the corpus.

This simplification introduces sparsity that we take advantage of during inference to speed performance. Because each term in ϕ_{POP} and ϕ_{CON} includes an indicator function based on a token match between a field-value and a message, knowing the possible values $\mathbf{v}$ of each R_k^ℓ enables us to precompute the combinations of (x, ℓ, v) for which nonzero factor values are possible. For each such tuple, we can also precompute the best alignment position k for each token x^j.

6 Evaluation Setup

Data We apply our approach to construct a database of concerts in New York City. We used Twitter's public API to collect roughly 4.7 Million tweets across three weekends that we subsequently filter down to 5,800 messages. The messages have an average length of 18 tokens, and the corpus vocabulary comprises 468,000 unique words[6]. We obtain labeled gold records using data scraped from the `NYC.com` music event guide; totaling 110 extracted records. Each gold record had two fields of interest: ARTIST and VENUE.

The first weekend of data (messages and events) was used for training and the second two weekends were used for testing.

Preprocessing Only a small fraction of Twitter messages are relevant to the target extraction task. Directly processing the raw unfiltered stream would prohibitively increase computational costs and make learning more difficult due to the noise inherent in the data. To focus our efforts on the promising portion of the stream, we perform two types of filtering. First, we only retain tweets whose authors list some variant of New York as their location in their profile. Second, we employ a MIRA-based binary classifier (Ritter et al., 2010) to predict whether a message mentions a concert event. After training on 2,000 hand-annotated tweets, this classifier achieves an F_1 of 46.9 (precision of 35.0 and recall of 71.0) when tested on 300 messages. While the two-stage filtering does not fully eliminate noise in the input stream, it greatly reduces the presence of irrelevant messages to a manageable 5,800 messages without filtering too many 'signal' tweets.

We also filter our gold record set to include only records in which each field value occurs at least once somewhere in the corpus, as these are the records which are possible to learn given the input. This yields 11 training and 31 testing records.

Training The first weekend of data (2,184 messages and 11 records after preprocessing) is used for training. As mentioned in Section 4, the only learned parameters in our model are those associated with the sequence labeling factor ϕ_{SEQ}. While it is possible to train these parameters via direct annotation of messages with label sequences, we opted instead to use a simple approach where message tokens from the training weekend are labeled via their intersection with gold records, often called "distant supervision" (Mintz et al., 2009b). Concretely, we automatically label message tokens in the training corpus with either the ARTIST or VENUE label if they belonged to a sequence that matched a gold record field, and with NONE otherwise. This is the only use that is made of the gold records throughout training. θ_{SEQ} parameters are trained using this labeling with a standard conditional likelihood objective.

Testing The two weekends of data used for testing totaled 3,662 tweets after preprocessing and 31 gold records for evaluation. The two weekends were tested separately and their results were aggregated across weekends.

Our model assumes a fixed number of records $K = 130$.[7] We rank these records according to a heuristic ranking function that favors the uniqueness of a record's field values across the set and the number of messages in the testing corpus that have

[6] Only considering English tweets and not counting user names (so-called -mentions.)

[7] Chosen based on the training set

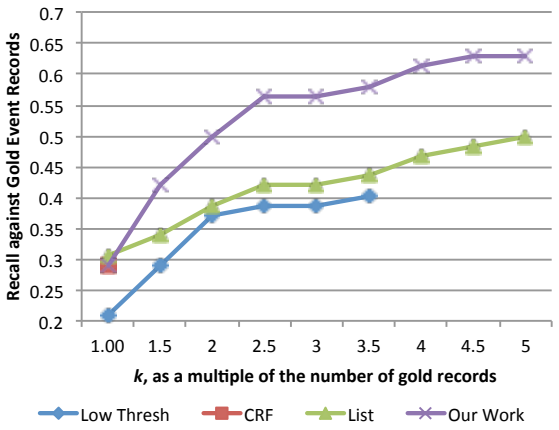

Figure 5: Recall against the gold records. The horizontal axis is the number of records kept from the ranked model output, as a multiple of the number of golds. The CRF lines terminate because of low record yield.

token overlap with these values. This ranking function is intended to push garbage collection records to the bottom of the list. Finally, we retain the top k records, throwing away the rest. Results in Section 7 are reported as a function of this k.

Baseline We compare our system against three baselines that employ a voting methodology similar to Mann and Yarowsky (2005). The baselines label each message and then extract one record for each combination of labeled phrases. Each extraction is considered a vote for that record's existence, and these votes are aggregated across all messages.

Our *List Baseline* labels messages by finding string overlaps against a list of musical artists and venues scraped from web data (the same lists used as features in our CRF component). The *CRF Baseline* is most similar to Mann and Yarowsky (2005)'s CRF Voting method and uses the maximum likelihood CRF labeling of each message. The *Low Threshold Baseline* generates all possible records from labelings with a token-level likelihood greater than $\lambda = 0.1$. The output of these baselines is a set of records ranked by the number of votes cast for each, and we perform our evaluation against the top k of these records.

7 Evaluation

The evaluation of record construction is challenging because many induced music events discussed in Twitter messages are not in our gold data set; our gold records are precise but incomplete. Because of this, we evaluate recall and precision separately. Both evaluations are performed using hard zero-one loss at record level. This is a harsh evaluation criterion, but it is realistic for real-world use.

Recall We evaluate recall, shown in Figure 5, against the gold event records for each weekend. This shows how well our model could do at replacing the a city event guide, providing Twitter users chat about events taking place.

We perform our evaluation by taking the top k records induced, performing a stable marriage matching against the gold records, and then evaluating the resulting matched pairs. Stable marriage matching is a widely used approach that finds a bipartite matching between two groups such that no pairing exists in which both participants would prefer some other pairing (Irving et al., 1987). With our hard loss function and no duplicate gold records, this amounts to the standard recall calculation. We choose this bipartite matching technique because it generalizes nicely to allow for other forms of loss calculation (such as token-level loss).

Precision To evaluate precision we assembled a list of the distinct records produced by all models and then manually determined if each record was correct. This determination was made blind to which model produced the record. We then used this aggregate list of correct records to measure precision for each individual model, shown in Figure 6.

By construction, our baselines incorporate a hard constraint that each relation learned must be expressed in entirety in at least one message. Our model only incorporates a soft version of this constraint via the ϕ_{CON} factor, but this constraint clearly has the ability to boost precision. To show it's effect, we additionally evaluate our model, labeled *Our Work + Con*, with this constraint applied in hard form as an output filter.

The downward trend in precision that can be seen in Figure 6 is the effect of our ranking algorithm, which attempts to push garbage collection records towards the bottom of the record list. As we incorporate these records, precision drops. These lines trend up for two of the baselines because the rank-

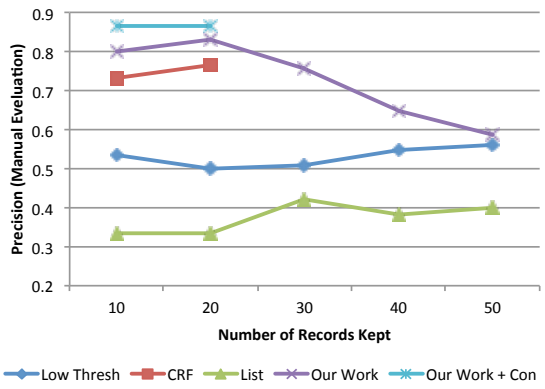

Figure 6: Precision, evaluated manually by cross-referencing model output with event mentions in the input data. The CRF and hard-constrained consensus lines terminate because of low record yield.

ing heuristic is not as effective for them.

These graphs confirm our hypothesis that we gain significant benefit by intertwining constraints on extraction consistency in the learning process, rather than only using this constraint to filter output.

7.1 Analysis

One persistent problem is a popular phrase appearing in many records, such as the value "New York" filling many ARTIST slots. The uniqueness factor θ_{UNQ} helps control this behavior, but it is a relatively blunt instrument. Ideally, our model would learn, for each field ℓ, the degree to which duplicate values are permitted. It is also possible that by learning, rather than hand-tuning, the θ_{CON}, θ_{POP}, and θ_{UNQ} parameters, our model could find a balance that permits the proper level of duplication for a particular domain.

Other errors can be explained by the lack of constituent features in our model, such as the selection of VENUE values that do not correspond to noun phrases. Further, semantic features could help avoid learning syntactically plausible artists like "Screw the Rain" because of the message:

Screw the rain$_{Artist}$! Grab an umbrella and head down to **Webster Hall**$_{Venue}$ for some American rock and roll.

Our model's soft string comparison-based clustering can be seen at work when our model uncovers records that would have been impossible without this approach. One such example is correcting the misspelling of venue names (e.g. *Terminal Five* → *Terminal 5*) even when no message about the event spells the venue correctly.

Still, the clustering can introduce errors by combining messages that provide orthogonal field contributions yet have overlapping tokens (thus escaping the penalty of the consistency factor). An example of two messages participating in this scenario is shown below; the shared term "holiday" in the second message gets relabeled as ARTIST:

Come check out the **holiday cheer**$_{Artist}$ parkside is bursting..
Pls tune in to TV **Guide Network**$_{Venue}$ TONIGHT at 8 pm for 25 Most Hilarious *Holiday* TV Moments...

While our experiments utilized binary relations, we believe our general approach should be useful for n-ary relation recovery in the social media domain. Because short messages are unlikely to express high arity relations completely, tying extraction and clustering seems an intuitive solution. In such a scenario, the record consistency constraints imposed by our model would have to be relaxed, perhaps examining pairwise argument consistency instead.

8 Conclusion

We presented a novel model for record extraction from social media streams such as Twitter. Our model operates on a noisy feed of data and extracts canonical records of events by aggregating information across multiple messages. Despite the noise of irrelevant messages and the relatively colloquial nature of message language, we are able to extract records with relatively high accuracy. There is still much room for improvement using a broader array of features on factors.

9 Acknowledgements

The authors gratefully acknowledge the support of the DARPA Machine Reading Program under AFRL prime contract no. FA8750-09-C-0172. Any opinions, findings, and conclusions expressed in this material are those of the author(s) and do not necessarily reflect the views of DARPA, AFRL, or the US government. Thanks also to Tal Wagner for his development assistance and the MIT NLP group for their helpful comments.

References

Eugene Agichtein and Luis Gravano. 2000. Snowball: Extracting relations from large plain-text collections. In *Proceedings of DL*.

Razvan C. Bunescu and Raymond J. Mooney. 2007. Learning to extract relations from the web using minimal supervision. In *Proceedings of the ACL*.

J Eisenstein, B O'Connor, and N Smith.... 2010. A latent variable model for geographic lexical variation. *Proceedings of the 2010 ...*, Jan.

Takaaki Hasegawa, Satoshi Sekine, and Ralph Grishman. 2004. Discovering relations among named entities from large corpora. In *Proceedings of ACL*.

Robert W. Irving, Paul Leather, and Dan Gusfield. 1987. An efficient algorithm for the optimal stable marriage. *J. ACM*, 34:532–543, July.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of International Conference of Machine Learning (ICML)*, pages 282–289.

P. Liang and D. Klein. 2007. Structured Bayesian nonparametric models with variational inference (tutorial). In *Association for Computational Linguistics (ACL)*.

Gideon S. Mann and David Yarowsky. 2005. Multi-field information extraction and cross-document fusion. In *Proceeding of the ACL*.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009a. Distant supervision for relation extraction without labeled data. In *Proceedings of ACL/IJCNLP*.

Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009b. Distant supervision for relation extraction without labeled data. In *Proceedings of the ACL*, pages 1003–1011.

A Ritter, C Cherry, and B Dolan. 2010. Unsupervised modeling of twitter conversations. *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 172–180.

Yusuke Shinyama and Satoshi Sekine. 2006. Preemptive information extraction using unrestricted relation discovery. In *Proceedings of HLT/NAACL*.

Roman Yangarber, Ralph Grishman, Pasi Tapanainen, and Silja Huttunen. 2000. Automatic acquisition of domain knowledge for information extraction. In *Proceedings of COLING*.

Limin Yao, Sebastian Riedel, and Andrew McCallum. 2010a. Collective cross-document relation extraction without labelled data. In *Proceedings of the EMNLP*, pages 1013–1023.

Limin Yao, Sebastian Riedel, and Andrew McCallum. 2010b. Cross-document relation extraction without labelled data. In *Proceedings of EMNLP*.

Jun Zhu, Zaiqing Nie, Xiaojing Liu, Bo Zhang, and Ji-Rong Wen. 2009. StatSnowball: a statistical approach to extracting entity relationships. In *Proceedings of WWW*.