Open Domain Event Extraction in Social Media

Alan Ritter

1 Motivation

A long-held application goal for Information Extraction and Natural Language Processing research is the ability to reliably detect and extract structured representations of events from real-time text streams such as news articles (Grishman and Sundheim, 1996; Allan et al., 1998; Yang et al., 1998; Pustejovsky et al., 2003; Doddington et al., 2004; Chambers and Jurafsky, 2011). While structured databases of events contain deep coverage within narrow domains (Becker et al., 2012), natural language text contains broad coverage of notable events across domains. New information about events taking place in the world is often communicated using natural language text before being entered into structured databases. In general people are more likely to mention an event than to fill out structured records describing it.

A system able to achieve acceptable performance at open-domain event extraction would have wide applicability; for example imagine a system which is able to answer questions such as the following:

- *What political events are scheduled this week?*
- *What new phones are coming out on Sprint next month?*
- *What significant events are going to happen in the near future?*

So far, however, applications of event extraction have been limited to narrow domains due to the difficulty of the task.

Previous work in event extraction has mostly focused on applying Information Extraction technologies to News Articles, as this genre of text has been the best source of information on current events since the spread of the printing press. Recently, however social networking sites such as Facebook and Twitter have become an important competing source of such information.

Status messages serve an information-dissemination function similar to news articles, however they have very different characteristics: they are short, easy for anyone to create, in addition to being instantly and widely disseminated. For these reasons, status messages often provide fresher information on a wider variety of topics and from a wider variety of perspectives than news. Status Messages also offer a natural measure of how interesting or important specific events are based on the number of users discussing them.

While status messages contain a wealth of useful information, they are very disorganized; in addition more tweets are written each day than any individual can read. These factors quickly lead to Information Overload. In contrast, anyone can sit down with a newspaper, and get a good overall summary of important events of the day categorized into topics of interest. Event extraction and aggregation, is therefore even more crucial for Twitter than for newswire.

Twitter has several characteristics which present unique *challenges* and *opportunities* for the task of open-domain event extraction. Intuition might lead us to think that extracting a useful structured representations of events from this soup of noisy text may be too difficult. On the other hand, tweets are short and self-contained and therefore don't require complex discourse processing as is the case for texts containing narratives. While there are arguments on either side of the issue, I believe *there is reason to be optimistic that after applying appropriate techniques, extracting an open-domain structured representation of current events from Twitter will be more effective than from newswire.*

1.1 Challenges

Twitter users frequently mention mundane events in their daily lives (such as what they ate for lunch) which are only of interest to their immediate social network. In contrast, if an event is mentioned in newswire text, it is safe to assume it is of general importance. Individual tweets are also very terse, often lacking sufficient context to categorize them into topics of interest (e.g. SPORTS, POLITICS, PRODUCTRELEASE etc...). Further, because Twitter users can talk about whatever they choose, it is unclear in advance which set of event types are appropriate. Finally, tweets are written in an informal style causing NLP tools designed for edited texts to perform extremely poorly.

1.2 Opportunities

On the other hand, the short and self-contained nature of tweets means they have very simple discourse and pragmatic structure, issues which still challenge state-of-the-art NLP systems. For example in newswire, complex reasoning about relations between events such as *before* and *after* is often required to accurately relate events to temporal expressions (Mani et al., 2006; Chambers et al., 2007). The volume of Tweets is also much larger than the volume of news articles, so redundancy of information can be exploited more easily.

1.3 Applicability within a Focused Domain

In addition to providing an up-to-date calendar of popular events, this work has potential to have impact on a wide range of Social Media applications by building and releasing general purpose tools for processing noisy Twitter text.[1] As an example, consider the flood of information encountered by crisis workers in the wake of natural disasters (generated from SMS and increasingly Twitter). Clearly there is great potential for Information Extraction and Text Mining to help by extracting and categorizing actionable information (Munro, 2011; Neubig et al., 2011). One of the main challenges in deploying Information Extraction technologies in these scenarios is the difficulty associated in quickly building systems tailored for the particular circumstances of the disaster. Public availability of general-purpose NLP tools tuned to Twitter's distinct style could potentially reduce the time to develop, thus helping to make this approach more effective.

2 Progress Thus Far / Demo

By extracting named entities (Ritter et al., 2011) and event-referring phrases, in addition to extracting and resolving temporal expressions within Twitter's noisy text, we are able to dynamically produce a calendar of the most popular events (including those in the near future). To demonstrate the promise of

[1] See: `http://github.com/aritter/twitter_nlp`.

this direction, a continuously updating prototype system can be viewed at `http://statuscalendar.cs.washington.edu`. [2]

2.1 Extracting Named Entities

Because of Twitter's noisy and informal style of text, capitalization and punctuation are often used for emphasis and proper nouns are frequently left lowercase. For these reasons, identifying named entities is more difficult in this informal text, furthermore as off-the-shelf NLP systems trained on news-corpora rely heavily on capitalization to identify named entities, they perform extremely poorly when applied to Twitter. In response I have annotated an in-domain dataset, which I have used to train models which significantly outperform the state-of-the-art Stanford NLP tools. In addition I have applied a minimally supervised approach to categorize named entities, which makes use of Freebase type lists to generate constraints in a latent variable model as a source of distant supervision (Ritter et al., 2011).

2.2 Extracting Event Mentions

In addition I have annotated a corpus of tweets with event mentions, which is then used to train sequence models to extract events. Event phrases can consist of many different parts of speech as illustrated in the following examples:

- **Verbs:** Apple to *Announce* iPhone 5 on October 4th?! YES!
- **Nouns:** iPhone 5 *announcement* coming Oct 4th
- **Adjectives:** WOOOHOO *NEW* IPHONE TODAY! CAN'T WAIT!

These phrases are useful to display in connection with entities on a calendar, providing additional insight into the nature of the event. For example displaying the entity, **Steve Jobs** and the event phrase **died** in connection with October 5th, is much more informative at a glance than simply displaying **Steve Jobs**. In addition, event words are helpful in upstream tasks, for example categorizing events into types, as described in §2.3.

[2] Based on data from the publicly available Twitter Streaming API `https://dev.twitter.com/docs/streaming-api` which represents 1% of the total stream of messages.

2.3 Classifying Extracted Events

To categorize the extracted events into high-level types I have investigated an approach based on latent variable models which is capable of both inferring an appropriate set of event types to match our data, and also classify events into types by leveraging large amounts of unlabeled data.

Supervised or semi-supervised classification of event categories is problematic for a number of reasons. First, it is *a priori* unclear which categories are appropriate for Twitter. Secondly, a large amount of manual effort is required to annotate tweets with event types. Third, the set of important categories (and entities) is likely to shift over time, potentially requiring periodic annotation of additional data. Finally many important categories are relatively infrequent, so even a large annotated dataset may contain just a few examples of these categories, making classification difficult.

For these reasons I was motivated to investigate unsupervised approaches that will automatically induce event types which match the data. We adopt an approach based on latent variable models inspired by recent work on modeling selectional preferences (Ritter et al., 2010; Séaghdha, 2010), and unsupervised information extraction Yao et al. (2011); Chambers and Jurafsky (2011).

Each event indicator phrase in our data, e, is modeled as a mixture of types. For example the event phrase "cheered" might appear as part of either a PoliticalEvent, or a SportsEvent. Each type corresponds to a distribution over named entities n involved in specific instances of the type, in addition to a distribution over dates d on which events of the type occur.

2.4 Ranking Significant Events

Because of the terse, sometimes mundane, but highly redundant nature of tweets, I have focused on extracting an aggregate representation of events which provides additional context for tasks such as event categorization, and also filters out mundane events by exploiting redundancy of information. Important events are identified as those whose mentions are strongly associated with references to a unique date as opposed to dates which are evenly distributed across the calendar.

3 Proposed Work

3.1 Distantly Supervised Temporal Reference Resolution

In order to place events on a calendar, we need to extract and resolve temporal expressions. For example, given the phrase "next Friday" we should be able to determine the unique calendar date which is referenced because we have the time at which the tweet was written. Currently we are using TempEx for this purpose (Mani and Wilson, 2000). TempEx is a rule-based system for extracting and resolving temporal expressions designed for use in newswire text, which we have found to have high precision when applied to tweets. TempEx's high precision can be explained by the fact that many temporal expressions are relatively unambiguous, however the creative spelling variations often employed on Twitter (for example "nxt fri") lead to low recall.

A natural approach to improving recall would be to annotate a large corpus of tweets with temporal expressions and manually resolve them to unique calendar dates for use as in-domain training data by discriminatively trained models. This annotation effort would be extremely tedious and time-consuming, however, as non-trivial effort is required to calculate the reference date for each annotated temporal expression.

As an alternative to manual annotation, I would like to investigate a distantly supervised approach to extracting and resolving temporal expressions. First, note that it is easy to retrospectively identify entities involved in significant events as those which are mentioned more than expected in tweets written on a single specific date, and these significant {`Entity,Date`} tuples can readily be identified using a statistical test. Once significant events are identified, we can apply a *distant supervision* assumption (Mintz et al., 2009) which states that tweets written within a fixed window around the date on which the event takes place[3] and also mention a key entity involved in the event should also refer to the date on which the event takes place. As a concrete example, assume the entity "LinkedIn" is mentioned much more than expected on May 19, 2011 (the date of their IPO); we then gather all tweets which mention "LinkedIn" n days before and after May 19, for example:

Tweeted on May 16: LinkedIn will go public on Thursday. I guess we'll

[3]For example += 2 weeks.

see what happens.

Tweeted on May 20: LinkedIn debuted on the NYSE yesterday - currently at $99.50/share (premarket). WAY too overpriced: http://bit.ly/mtV8oO Thoughts?

Then we can consider each such {`Tweet,ReferenceDate,TargetDate`} tuple as a (distantly supervised) training example. Negative examples can consist of dates within a window other than the target date, and feature templates can consist of conjunctions of lexical features from the tweet and properties of the target date. Examples of features generated using these templates might include:

- TENSE = PAST ∧ WORD = "debuted"
- DOW = WEDNESDAY ∧ WORD = "wed".

In practice the distant supervision assumption presented above may be too unrealistic to be effective because tweets which mention an event won't always mention the date on which it takes place. We therefore may need to introduce additional latent variables which determine whether each word is part of a temporal expression. One possible approach would be to model temporal expressions using a constrained Hidden Markov Model, where a subset of the hidden states are arbitrarily mapped to date properties in advance (e.g. TENSE = PAST, DOW = WEDNESDAY, WEEK = NEXT, etc...) and during training the transition matrices of the HMMs are constrained such that the probability of transitioning into each of these states is zero, except if the associated target date has that property active. Parameters would be shared across HMMs except for those set to zero due to constraints. Learning could proceed in a straightforward fashion using EM (the forward-backward algorithm) or sampling-based inference (Goldwater and Griffiths, 2007). During inference (decoding) the HMM would be left unconstrained. After decoding, temporal expressions with associated properties of the target date could be read off based on the inferred values of hidden variables. If necessary, feature engineering (and overlapping correlated features such as prefixes, suffixes and dictionary-based features) could be accommodated by modeling the emission distributions with locally normalized logistic regression models as suggested by Berg-Kirkpatrick et al. (2010). Note that an additional (distantly supervised) classification step would likely be necessary as date properties are often left semantically ambiguous, even though they are pragmatically well defined. For example in the absence of an explicit past tense marker a phrase like "the Wednesday meeting" is usually assumed to refer to the future.

3.2 Event Reference Resolution

An individual tweet can refer to an event in a wide variety of different ways. Because tweets are short, they are likely to mention only a subset of entities involved in the event, and might not even mention the date on which it occurs. Also there are many event phrases which may be used to refer to the same type of event, for example either "game" or "plays" might be used to refer to a sports event.

To address these issues, an important task is to cluster together all mentions of the same distinct event. This grouping of mentions into co-referring events will be useful for many purposes, such as providing a better estimate of the number of references to an individual event (and thus a better estimate of it's importance or popularity), deduplication of event mentions, in addition to giving more complete information about each event for upstream tasks such as schema discovery (see §3.3).

Resolving event references in Twitter is a non-trivial task. Not any clustering of mentions into events will be sufficient. We want our grouping of mentions to be guided by the following intuitions:

- Entities which co-occur together in multiple tweets should be part of the same event, for example the following tweets should be seen as co-referring, assuming we observe *Ben* co-occurring frequently enough with *The Bachelor*.
  - **The Bachelor** tonight. **Ben**'s a hawtie. #marryme #sopumped
  - Was **Ben** totally boring tonight or was it just me??? #bachbloggers
  - **The Bachelor** starts tomorrow. #excited :)
- Co-referring event mentions should have the same event type (see §2.3) For example, the following mentions could be clustered together, because one tweet contains the phrase *playing* and the other contains *game* both of which are indicative of a Sports event. Also the entities *Kobe*, *Knicks*, and *Lakers* should indicate the same high-level type.
  - **Kobe** is **playing** on point tonight. Kudos to him!

  - **Knicks** against **Lakers** tomorrow. Def. watchin that **game**.

- Each event should have a relatively small number of entities involved.

- Each event occurs on a unique calendar date.

  - Mentions of the event should either refer to this date, or be written near the date.

To capture these intuitions, I would like to investigate a generative probabilistic model of events mentioned on Twitter.

3.2.1 Simple Mixture Model

As a first (simple) model, consider the following assumptions:

- Each event is only associated with a single date (this effectively partitions the data across dates).

- Each event has probability distributions over named entities and event phrases involved in the event.

- Each tweet is generated based on a single event.

These assumptions correspond to a relatively straightforward mixture of multinomials, where each mixture component corresponds to a distinct event. A (simplified) Bayesian network representation of this model is presented in Figure 1. Because the number of distinct events will vary on any given day, I believe it is important to use a nonparametric approach, such as the Dirichlet Process Mixture Model (Antoniak, 1974) so that the correct number of events can be inferred based on the data. This is a relatively straightforward mixture model, and can serve as a baseline.

3.2.2 Event Type Mixture Model

In order share information across dates, we can extend the model to incorporate *event types*.

Here the observed data are generated from hidden event types in addition to events, as described below:

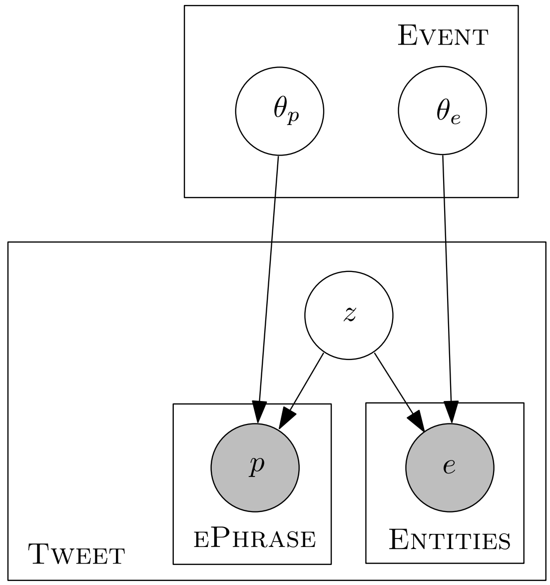

Figure 1: Simple mixture model of events.

EVENTTYPE Each event type has an associated distribution over named entities involved in events of the type, θ_e, in addition to a distribution over event phrases, θ_p.

EVENT Event instances have a type, t_z and a set of named entities, which are generated based on it's type (e.g. they are drawn from the appropriate θ_e)

TWEET Each tweet contains a set of observed named entities and event referring phrases, in addition to a hidden variable z, which determines the aggregate event it belongs to.

A tweet's entities are drawn from the list associated with it's event, and it's event phrases are drawn from the event type associated with the event. The type parameters can be shared across dates. A graphical model representation is presented in Figure 2.

While parameters for the event types could be learned as part of this model, it may be more efficient to use parameters learned independently using the simpler event type model we have already developed (see §2.3), which is efficient and easy to parallelized to extend to large datasets.

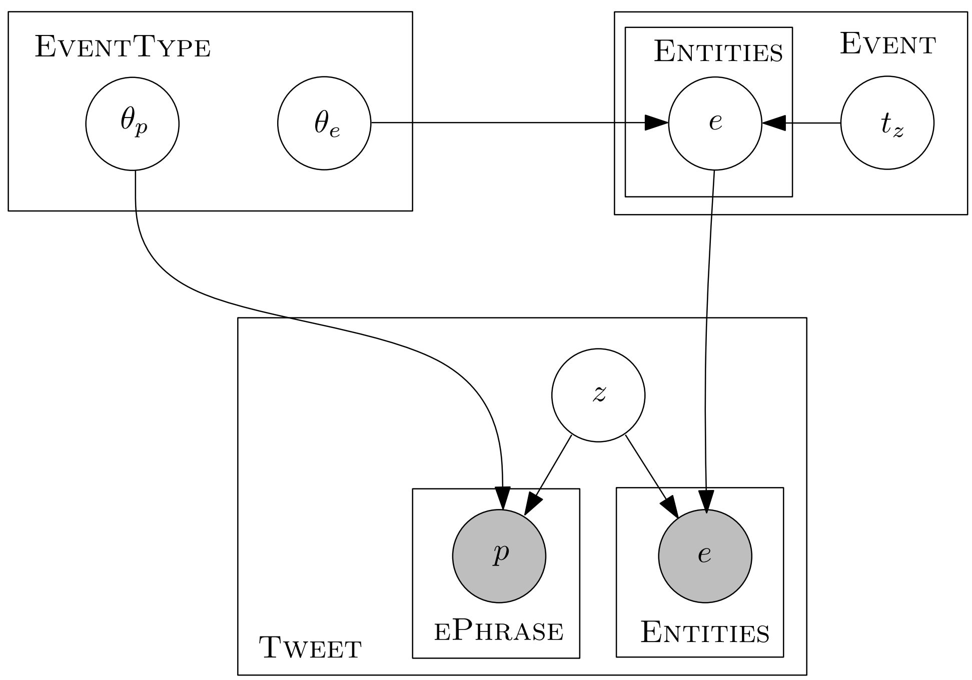

Figure 2: Generative model of events and event types.

3.2.3 Inference

I believe an approach to inference similar to that presented by Haghighi and Klein (2010) which combines mean field variational inference (Jordan et al., 1999) with search (Daumé III, 2009) will be appropriate for this model.

Mean-Field variational inference is fairly straightforward: hidden variables are divided into subsets, each of which is iteratively updated holding the others constant; this is similar to EM, which is closely related to Variational Inference.

In our case we can use the following groups of hidden variables:

- per-tweet event assignments (z_i) and per-event entity lists (L_j)
- per-event type (t_j)
- Conditional parameters (these can actually just come from the event-type topic model (§2.3) - which is trained on more data - in this case the parameters are fixed in advance):

$$\begin{aligned}\theta_e^{t_j} &= P(\text{entity}|t_j) \\ \theta_p^{t_j} &= P(\text{event phrase}|t_j)\end{aligned}$$

For inference, we can alternate between searching for a MAP assignment to the z_is (which determine the event associated with each tweet, in addition to the set of entities associated with each event) and updating the current posterior distribution over the t_j's (which determine the type of each event). Right now I'm using a simple greedy hill-climbing approach to updating the z_i's (e.g. iteratively pick each z_i to maximize the joint probability of the observed data and hidden variables, $P(D, Z)$. This simple approach is a nice place to start as it's easy to implement and fast, however it is also very prone to becoming stuck in local minima[4]. I believe a better solution will be to use common heuristic search methods, such as A^* as applied to similar problems by Daumé III (2009) and Haghighi and Klein (2010).

3.2.4 Preliminary Experiments

Following are some preliminary examples of groups of entities and event phrases generated using the Event-Type mixture model described above.

[4] For example given an intermediate clustering, it's difficult to split or merge clusters which may require transitioning through many low-probability configurations.

date	entities	event phrases
01/02/2012	rose bowl:39, oregon:39, wisconsin:18	watching:8, wearing:4, watch:4, looking:3, sick:3, game:3, school:3, work:3, working:2, live:2, clean:2, ready:2, love:2, praying:2, looking forward:2, making:2, talkifight:7, win:5, watch:4, brock:3, create:2, wins:2, fights:2, hard:2, touch ng:2, look forward:2
01/02/2012	bachelor:28, ben:16, abc:14, the bachelor:4	starts:14, excited:6, watch:5, season:4, premiere:4, new season:3, watching:3, show:2, is on:2, boring:2, starts up:2, waiting:2, coming:2, comes on:2
12/30/2011	brock:38, overeem:34, reem:5	overeem:8, fight:7, win:5, watch:4, brock:3, create:2, wins:2, fights:2, hard:2, touch gloves:2, see:2
1/3/2012	liverpool:60, man city:37, suarez:10	win:16, liverpool:15, game:10, appeal:7, watch:6, supporting:5, playing:4, beat:4, support:4, rise:3, match:3, decided:3, see:3, watching:2, starts:2, match ban:2, miss:2, statement:2, need:2, lose:2, start:2, go fourth:2, cheering:2, beat liverpool:2, said:2
1/6/2012	wizards of waverly place:9, disney channel:7	watching:4, show:3, watch:3, episode:3

3.3 Event Schema Discovery

The entities involved in a given event can be seen as filling a particular field in the schema determined by the event's type. For example, a ProductRelease event might be expected to involve a *Company* and *Product* (for instance *Apple* and *iPhone 5*). For systems focused on extracting events within a narrow domain (Benson et al., 2011), the schema is easily specified in advance. Because our events and types are extracted in a fully open-domain manner, and twitter users can talk about any entities involved in any event, we have no way of knowing in advance what schema is appropriate for each of the types discovered by our model.

Recent work by Cafarella et al. (2007) has explored schema discovery for open-domain relation extraction using a data-mining style algorithm. I would like to investigate an analogous approach to schema discovery for open-domain events extracted from Twitter based on a probabilistic generative model. By extending the model of events presented in §3.2 we can consider our observed event mentions as generated based on 3 classes of hidden objects with appropriate attributes:

EventType Each event type has an associated set of entity types which describe it's schema, in addition to an associated multinomial distribution over event phrases. For example the SportsEvent type might have a schema consisting of the types $t_s \in \{$*Team, Stadium, City, Player, Coach*$\}$, and a multinomial distribution θ_p which assigns high probability to phrases such as *played*, *game*, etc...

Event **(see §3.2)** Each event instance has an associated event type and a set of named entities (which are generated from entity types associated with the event type's schema).

EntityType Each entity type has a corresponding distribution over entity mentions. These type distributions could be bootstrapped from lexical resources like Freebase using an approach such as that presented in Ritter et al. (2011).

Tweet Each tweet contains a set of observed named entities and event referring phrases, in addition to a hidden variable z, which determines the event instance it belongs to.

Finally note that it may be beneficial to perform schema induction jointly with event type categorization and event reference resolution.

A (high-level) Bayesian network representation for our generative model is presented in Figure 3.

4 Evaluation

To evaluate performance at open-domain event extraction from Twitter, I would like to evaluate each of the individual components developed as part of this work against appropriate baselines, in addition to evaluating the overall performance of open-domain event extraction from Twitter as compared against a similar approach applied to News articles.

4.1 Individual Components

4.1.1 Temporal Resolution

In order to evaluate temporal reference resolution in Twitter, I plan to annotate a randomly sampled corpus of tweets with temporal expressions, in addition to the dates they refer to. Precision and recall of temporal resolution can then be evaluated against gold labels. A natural baseline to compare

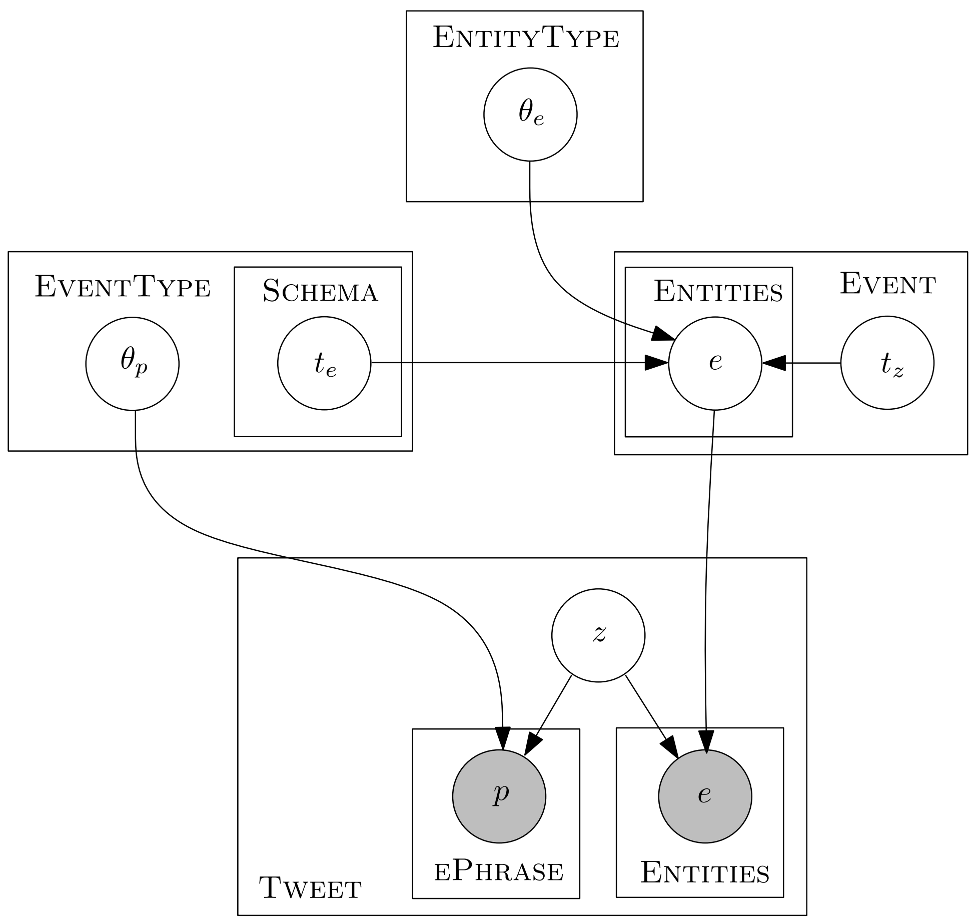

Figure 3: Generative model of events, event types and type schema.

against will be TempEx (Mani and Wilson, 2000), which although developed for news has surprisingly high precision on Tweets. I believe there is much room for improvement in recall, however, which is very important for our purposes, as it provides better coverage of events and will thus benefit all upstream tasks.

4.1.2 Event Reference Resolution

Again, a corpus of tweets will need to be manually annotated with events. Gathering this dataset will be somewhat more difficult, however because in a random sample of tweets of size feasible for annotation, it is unlikely any two will refer to the same distinct event. Instead we may need to rely on a somewhat artificial dataset for evaluation, for example collected using keywords which are carefully chosen to refer to selected events, and then manually filtering out those tweets which do not refer to the event. As a baseline we can compare against standard clustering algorithms such as the Dirichlet Process Mixture Model presented in §3.2.1, in addition to K-nearest neighbors.

4.1.3 Event Schema Discovery

To evaluate the performance of schema discovery, we can create a *gold* set of schema for a few selected event types by inspecting large quantities of data, and manually labeling the types with appropriate schema. We can then evaluate what fraction of fields from the gold schema our automatically discovered fields cover (recall) and what fraction of the automatically generated fields are found in the gold schema (precision).

In addition we will need to evaluate precision and recall at categorizing entities involved in extracted events into the fields associated with the type's schema. Again we can annotate a set of such entities with gold labels and evaluate precision and recall.

As baselines we can compare against TGEN (Cafarella et al., 2007), in addition to the approach to named entity categorization presented in Ritter et al. (2011).

4.2 Comparison to Event Extraction from News

Finally, I would like to compare open-domain extraction of current events from Twitter against that from News. First off, I would like to evaluate both precision and yield of open-domain event extraction in both systems. As discussed in §1 Twitter presents both distinct *challenges* and *opportunities* when compared with news, however I am optimistic that Twitter will be a better data source for open-domain event extraction due to it's simple discourse structure and high level of redundancy.

To perform this experiment, it will be necessary to have access to an open-domain IE system for news which is comparable to what will be developed for Twitter. Fortunately, since all of the techniques proposed are fairly general, it should be straightforward to apply the same approach to news articles, while swapping out appropriate components where they are available (for example using news-trained named entity recognizers and event recognizers). We can then evaluate the precision and yield of the systems extracting open-domain events from both Twitter and news.

Finally, given this data we can ask can ask: qualitatively what is the difference in events extracted from Twitter vs. News? Are there events which are popular in Twitter before news or vice versa? Are there specific kinds of events which are more strongly represented in Twitter?

References

James Allan, Ron Papka, and Victor Lavrenko. On-line new event detection and tracking. In *SIGIR*, 1998.

Charles E. Antoniak. Mixtures of dirichlet processes with applications to bayesian nonparametric problems. *The Annals of Statistics*, 1974.

Hila Becker, Dan Iter, Mor Naaman, and Luis Gravano. Identifying content for planned events across social media sites. In *WSDM*, 2012.

Edward Benson, Aria Haghighi, and Regina Barzilay. Event discovery in social media feeds. In *ACL*, 2011.

Taylor Berg-Kirkpatrick, Alexandre Bouchard-Côté, John DeNero, and Dan Klein. Painless unsupervised learning with features. In *NAACL*, 2010.

Michael J. Cafarella, Dan Suciu, and Oren Etzioni. Navigating extracted data with schema discovery. In *WebDB*, 2007.

Nathanael Chambers and Dan Jurafsky. Template-based information extraction without the templates. In *Proceedings of ACL*, 2011. URL `pubs/chambers-acl2011-muctemplates.pdf`.

Nathanael Chambers, Shan Wang, and Dan Jurafsky. Classifying temporal relations between events. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 173–176, Stroudsburg, PA, USA, 2007. Association for Computational Linguistics. URL `http://dl.acm.org/citation.cfm?id=1557769.1557820`.

Hal Daumé III. Fast search for dirichlet process mixture models. *CoRR*, 2009.

G. Doddington, A. Mitchell, M. Przybocki, L. Ramshaw, S. Strassel, and R. Weischedel. The Automatic Content Extraction (ACE) Program–Tasks, Data, and Evaluation. *LREC*, 2004.

Sharon Goldwater and Tom Griffiths. A fully bayesian approach to unsupervised part-of-speech tagging. In *ACL*, 2007.

Ralph Grishman and Beth Sundheim. Message understanding conference - 6: A brief history. In *Proceedings of the International Conference on Computational Linguistics*, 1996.

Aria Haghighi and Dan Klein. Coreference resolution in a modular, entity-centered model. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association*, 2010.

Michael I. Jordan, Zoubin Ghahramani, Tommi S. Jaakkola, and Lawrence K. Saul. An introduction to variational methods for graphical models. *Mach. Learn.*, 1999.

Inderjeet Mani and George Wilson. Robust temporal processing of news. In *ACL*, 2000.

Inderjeet Mani, Marc Verhagen, Ben Wellner, Chong Min Lee, and James Pustejovsky. Machine learning of temporal relations. In *ACL*, 2006.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of ACL-IJCNLP 2009*, 2009. URL `pubs/mintz09.pdf`.

Robert Munro. Subword and spatiotemporal models for identifying actionable information in haitian kreyol. In *CoNLL*, 2011.

Graham Neubig, Yuichiroh Matsubayashi, Masato Hagiwara, and Koji Murakami. Safety information mining - what can NLP do in a disaster -. In *IJCNLP*, 2011.

J. Pustejovsky, P. Hanks, R. Sauri, A. See, R. Gaizauskas, A. Setzer, D. Radev, B. Sundheim, D. Day, L. Ferro, and M. Lazo. The TIMEBANK corpus. In *Proceedings of Corpus Linguistics 2003*, 2003.

Alan Ritter, Mausam, and Oren Etzioni. A latent dirichlet allocation method for selectional preferences. In *ACL*, 2010.

Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. Named entity recognition in tweets: An experimental study. *EMNLP*, 2011.

Diarmuid Ó. Séaghdha. Latent variable models of selectional preference. In *ACL*, ACL '10, 2010.

Yiming Yang, Tom Pierce, and Jaime Carbonell. A study of retrospective and on-line event detection. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '98, 1998.

Limin Yao, Aria Haghighi, Sebastian Riedel, and Andrew McCallum. Structured relation discovery using generative models. In *EMNLP*, 2011.