

# Community-Based Classification of Noun Phrases in Twitter

Freddy Chong Tat Chua<sup>\*</sup>  
Living Analytics Research Centre  
Singapore Management University  
freddy.chua.2009@smu.edu.sg

Justin Betteridge  
Language Technologies Institute  
Carnegie Mellon University  
jbetter@cs.cmu.edu

William W. Cohen  
Machine Learning Department  
Carnegie Mellon University  
wcohen@cs.cmu.edu

Ee-Peng Lim  
Living Analytics Research Centre  
Singapore Management University  
eplim@smu.edu.sg

## ABSTRACT

Many event monitoring systems rely on counting known keywords in streaming text data to detect sudden spikes in frequency. But the dynamic and conversational nature of Twitter makes it hard to select known keywords for monitoring. Here we consider a method of automatically finding noun phrases (NPs) as keywords for event monitoring in Twitter. Finding NPs has two aspects, identifying the boundaries for the subsequence of words which represent the NP, and classifying the NP to a specific broad category such as politics, sports, etc. To classify an NP, we define the feature vector for the NP using not just the words but also the author's behavior and social activities. Our results show that we can classify many NPs by using a sample of training data from a knowledge-base.

## Categories and Subject Descriptors

H.2.8 [Database Applications]: [Data Mining]; I.2.7 [Natural Language Processing]: [Text analysis]

## General Terms

Measurement, Experimentation

## Keywords

Social Media, Noun Phrases, Twitter, Named Entities

<sup>\*</sup>This work was completed when the author was a visiting student at Carnegie Mellon University.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CIKM'12, October 29–November 2, 2012, Maui, HI, USA.  
Copyright 2012 ACM 978-1-4503-1156-4/12/10 ...\$10.00.

## 1. INTRODUCTION

*Twitter*, a micro-blogging website has very wide coverage in people's life. Snippets of text, which we call *tweets*, are written by Twitter users commenting on their sentiments toward consumer products, political opinions, sports and daily mundane activities. To understand a tweet's meaning, important key words such as noun phrases (**NPs**) need to be recognized in the tweet. Each noun phrase (**NP**) represents a consecutive sequence of words that mentions a specific noun or entity.

A direct application of NP recognition is event-monitoring in real time Twitter streams. Prior research by Sakaki et al. have shown that by monitoring specific known keywords in Twitter, it is possible to perform event detection [10]. These event-monitoring systems usually track a fixed set of manually chosen NPs. But given the dynamic nature of Twitter, we often do not know what NPs should be monitored. Some important NPs that describe the subject of the tweet might be  $n$ -grams where the value of  $n$  varies widely. Users on social media are also often inventing new words and NPs to comment on recent events. Such dynamically changing vocabulary means that a fixed dictionary of monitored keywords are unlikely to capture recent events.

**Problem Definition:** Recognizing NPs has two aspects. 1) Identify the boundaries of the word subsequence that represents each NP. 2) Given the NPs, classify each NP into a pre-defined set of categories such as politics or sports. For example, in the tweets mentioned previously, we want to classify the NP *a balance budget amendment* as political and *a shitty NFL team* as sports. In order to perform classification, we propose to use the features of a NP based on the author's behavior and community preferences.

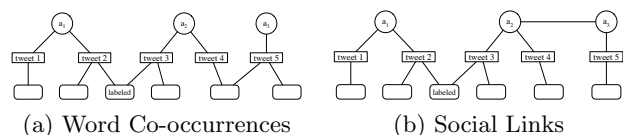


Figure 1: Features for Classification

For example in Figure 1(a), authors  $a_1$  and  $a_2$  share a NP in tweet 2 and tweet 3 respectively and this NP has been classified to a category. From that classified NP, in-

formation of the category can propagate to other NPs that  $a_1$  and  $a_2$  have written, thus allowing us to infer the other unclassified NPs. Although author  $a_3$  does not have any classified NP, the information from the classified NP of  $a_1$  and  $a_2$  can propagate through the common NP of  $a_2$  and  $a_3$  from tweets 4 and 5 respectively. If most NPs are unigrams, dimension reduction algorithms such as Singular Valued Decomposition (SVD), Non-Negative Matrix Factorization (NNMF) or Latent Dirichlet Allocation (LDA) can model the co-occurrences of words with different authors. Since most NPs are not unigrams, we propose an extension of LDA called NP+LDA to handle n-grams. These dimension reduction algorithms use such co-occurrences to reduce the authors and NPs into a latent space with significantly lower dimension.

The structure in Figure 1(b) shows author  $a_3$  who has no common NP with either  $a_1$  or  $a_2$ . In this case, the typical dimension reduction algorithms will not be able to capture the necessary information for inferring the categories of NPs from  $a_3$ . But information from the classified NP could propagate through the social link which connects  $a_2$  and  $a_3$ . So then the dimension reduction algorithm to find the feature vector should consider the social link between  $a_2$  and  $a_3$ . The Role Author Recipient Topic Model (RART) [7] which models unigrams from conversation based corpus does consider the authors' social dimensions. Similar to LDA, we propose an extension NP+RART to handle n-grams of NPs. Instead of having a topic distribution for each author, we could first find the communities each author belongs to and define a topic distribution for the communities.

**Challenges:** When modeling n-grams of NPs using topic models such as LDA, it is tempting to assume that we can treat each different NP as a single word token. But that causes an inflated number of unique "words" in the model. For example, given that an NP has  $n$  unique words, there are a total of  $n!$  different variations of that NP. Too many unique words in the model causes the data to be extremely sparse because of low occurrences for each variation of the NP. So we need a model that handles n-grams of NP which also considers the individual word tokens in the NP. We use NP+LDA, which is previously proposed by Wallach [11]. NP+LDA considers n-grams instead of individual word tokens for inferring the latent topics for each NP.

Much of traditional text analysis has focused on exploiting language semantics and grammar structure in text for extracting information. But tweets often have the following problems, 1) Tweet length is usually short with less than 30 words. This makes it harder to infer the context of the NPs in tweets. 2) Spelling and grammar are usually wrong despite the high literary rates in modern societies. The variation in spelling makes it hard to compare against a dictionary. The grammar idiosyncrasy makes it hard to use linguistic rules to infer where the NPs are. 3) Many tweets offer little information and impact on society, for example, the last tweet in the examples earlier does not tell us very much about the author. As a result, we may waste unnecessary time on processing uninformative tweets.

Instead of using the well-explored semantics in Natural Language Processing, Twitter provides additional features for text analysis. In Twitter, users follow other users to track and subscribe to the tweets other users write. Users also retweet (cite other users' tweets) from other users. These follow and retweet relationships result in formation of com-

munities of Twitter users who focused on discussing certain categories of topics. But no prior work has ever attempted to use the social features or community-based information for classifying NPs.

We propose a method of identifying NPs in tweets using its Part-of-Speech (POS) tags. After we have identified the boundaries of where the NPs are, classification is formulated as a set of binary classification problems to infer whether these NPs belong to a set of categories that we want to find (e.g. politics, sports). We obtain the feature vectors for classification using the authors' topic preferences and their communities membership.

Our contributions are as follows, 1) We show that finding the NPs can be performed easily by using Regular Expressions on the Part-of-Speech (POS) tags of the words in tweets. We POS-tagged a total of 2.9 million tweets using the annotated Twitter data set by Gimpel et al. [4]. 2) We use a sentence reader trained on an existing knowledge base, the Never Ending Language Learner (NELL) and ClueWeb corpus for classifying these NPs. 3) Because the sentence reader is only able to classify a small proportion of the NPs with high confidence, we use a second classifier which derives feature vectors from two models NP+LDA and NP+RART. 4) Using NP+LDA, we learn the topic distributions of the authors to find the topics of words that authors frequently tweet about. Then using the topic distributions of authors and n-grams as feature vectors for supervised learning, we perform classification on these NPs using the categories of a partially known set of NPs from the Never Ending Language Learner (NELL)'s knowledge base [2]. 5) From NP+LDA and Role Author Recipient Topic Model (RART) [7], we further extend and propose NP+RART to show that we can improve the classification of NPs using the community information in Twitter.

We discuss some related and prior work in the area of noun phrase and named entity recognition (NER) in Section 2. Section 3 describes our methods and models in detail. Then we evaluate our work in Section 4. Finally we end our paper in Section 5 with a conclusion of what we have done and some possible future work.

## 2. RELATED WORK

Ritter et al. built an NLP pipeline for performing Named Entity Recognition (NER) in tweets [9]. The pipeline begins with Part of Speech (POS) tagging of tweets using a supervised learning approach, followed by text chunking then classification using an annotated set of tweets.

Liu et al. proposed an alternating two-step approach to perform the NER task in tweets [6]. The two-step approach alternates between the KNN classifier and CRF labeler. The KNN classifier models global features which span over long range of words. The CRF models the localized features among consecutive words. To train these two learning algorithms, 12,245 tweets were manually annotated by two independent annotators.

The work of Michelson and Macskassy is closely related to ours [8]. Michelson and Macskassy proposed a two step process to understand a tweet  $T$  content in order to profile the authors' topic interests. The first step identifies and disambiguates possible named entities as candidates in tweet  $T$ . The second step uses the hierarchical classification in Wikipedia to profile the authors' topic interests.

Li et al. [5] proposed an unsupervised method of finding

named entities in Twitter using Wikipedia and Microsoft N-gram data as knowledge-base. Li et al. approach is to find the optimal segments of word as named entities using a dynamic programming approach. However, unlike our approach, they do not classify the named entities into its respective categories.

Our use of topic distributions as feature vectors for supervised learning is inspired by previous work on Supervised Topics Models by Wang et al. [1, 12]. The topic distributions are used as feature vectors for a Logistic Regression classifier without the bias intercept. However, Wang et al. use the topic distributions for classifying documents while we use it for a more fine grain classification of noun phrases in the documents.

### 3. NOUN PHRASE (NP) RECOGNITION

Let us first examine the list of features in the data set and the tools we have. 1) Retweets in Twitter data set give us a hint that the retweeted tweets contain meaningful information for extraction. 2) Gimpel et al. has a set of annotated POS tweets [4], which we can use for training a POS Tagger 3) Each tweet is associated with at least an author and the author is likely to focus on a few specific topics. 4) Replies and Retweets give us a hint of the active and implicit social network beneath the passive and explicit follow relationships. 5) Knowledge bases such as NELL provide entities disambiguation subjected to whether the entity and its context pattern exist in NELL’s knowledge base.

In summary, we used POS tagging and regular expressions to segment the boundaries of NPs in every tweet. We automatically classify a small set of NPs using NELL’s knowledge base (KB). Then we derive feature vectors on the KB-classified and KB-unclassified NPs using two models; NP+LDA and NP+RART. We train a classifier using the categories and feature vectors of KB-classified NPs, then use the classifier on the feature vectors of KB-unclassified NPs to obtain broad-level categories.

#### 3.1 Extraction of Noun Phrases using Part-of-Speech (POS) Tags

To extract NPs, we use the POS Tagger provided by Gimpel et al. [4] to tag the tweets. From the POS tags of the words in each tweet, we use a lexical analysis program *lex*, to recognize the regular expression for obtaining NPs. The following regular expressions are used to obtain the NPs,

$$\begin{aligned} \text{Base\_NP} &:= \text{determiner?} \text{ adjectives} * \text{ nouns} + \\ \text{Conj\_NP} &:= \text{Base\_NP}(\text{of Base\_NP}) * \end{aligned}$$

The regular expression rules we described here can be interpreted as follows: a base noun phrase (BASE\_NP) can have zero or one determiner, zero or non-zero adjectives and at least one or more nouns. A conjunctive noun phrase (Conj\_NP) is made up of one or more base NPs, e.g. “the student of computer science”.

Once we have the NPs, we can derive a feature vector to represent each NP. The feature vector represents the author’s writing preferences or her belonging to a community base on her social activity in Twitter. If we also have known information for the categories of some NPs, we can use these partial information to classify the categories of other unknown NPs.

### 3.2 Never Ending Language Learner (NELL) for Supervised Learning

NELL<sup>1</sup> is a knowledge base that stores two pairs of information for NP classification [2, 3]. 1) The *noun phrase* and 2) words that come before and after the NP known as the *contextual patterns*.

To utilize NELL’s knowledge base for classifying noun phrases, we obtained a sentence-level semantic category recognizer (SentReg) for many different categories. We follow the basic approach described by Whitelaw et al. with a few variations [14]. SentReg was trained using NELL’s knowledge base on the ClueWeb corpus<sup>2</sup>. However, SentReg does not have high recall on Twitter data for these reasons. 1) SentReg was trained on ClueWeb, which has better grammatical structure and language semantics than Twitter. 2) Many Twitter tweets are related to very recent events and NELL’s knowledge base is unlikely to have the contextual patterns for the new NPs. 3) Twitter data does not have many NP contextual pattern for SentReg to recognize. Therefore, we only use NPs classified by SentReg with high confidence and this results in a very small set of NPs as compared to all the NPs we have.

#### 3.3 Noun Phrases for Latent Dirichlet Allocation (NP+LDA)

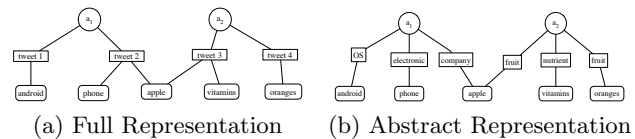


Figure 2: Representation

After SentReg classifies a small set of NPs, we need to represent these NPs in the form of a feature vector to be used in a supervised classification algorithm. We can either have a feature vector which represents the NP, or the edge connecting the NP and the author.

Figure 2(a) shows two authors  $a_1$  and  $a_2$ . Author  $a_1$  writes tweet 1 which contains the NP *android*. Tweet 2 written by  $a_1$  has the NPs *phone* and *apple*. Similarly,  $a_2$  writes tweet 3 that contains *apple* and *vitamins*. Tweet 4 written by  $a_2$  contains only one NP *oranges*. We can proceed to learn the meanings of the NPs by classifying each word to a specific category. However, observe that the NP *apple* can belong to the category *company* or *fruit* depending on which author wrote the NP. The NP *apple* written by  $a_1$  is inferred as *company* while the NP *apple* written by  $a_2$  should be classified as *fruit*. Therefore, it is more specific to classify the edge which connects the author and NP instead of the NP itself. Figure 2(b) shows the result of classifying the author - NP edges. This is the goal of our proposed models, to find feature vectors connecting the author and NP, then classify the edge to some category.

Since most tweets are short and contain only one to three NPs, it will be easier to learn the topic distribution of the authors by aggregating all the NPs tweeted by the authors as opposed to treating each tweet as an individual document, this approach was previously applied in TwitterRank [13].

<sup>1</sup><http://rtw.ml.cmu.edu>

<sup>2</sup><http://lemurproject.org/clueweb09.php/>

Formally, given a set of tweets  $T_a$  written by an author  $a$ , for each tweet  $t \in T_a$ , suppose we have the noun phrases  $NP_t$  in tweet  $t$ . For each  $n \in NP_t$ , obtain the feature vector  $\psi_{t,n}$  representing the noun phrase from tweet  $t$  written by author  $a$  and associate a category  $c_{t,n}$  if the noun phrase has been previously recognized by SentReg.

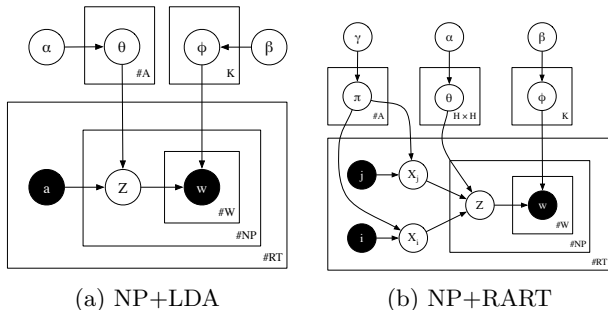


Figure 3: Topic Models with Noun Phrases (NP)

Figure 3(a) shows the graphical model of NP+LDA model. We denote the tweets written by author  $a$  as  $T_a$ . Each tweet is represented as  $t$ , where  $t \in T_a$ . Each NP of  $t$  as  $n$  and words in the NP as  $w_{t,n,i}$ . Instead of generating each word from each latent variable  $z$ , we generate a noun phrase which consists of a set of words. This way, we can ensure that all words from a noun phrase belong to the same topic. This assumption should be reasonable because a noun phrase is too short to contain a mixture of topics. Interested readers may refer to Wallach [11] for more details.

### 3.4 Noun Phrases for Role Author Recipient Topic Model (NP+RART)

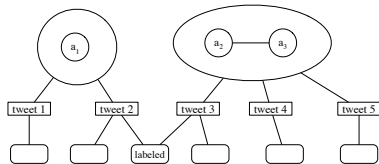


Figure 4: Communities in Twitter

Recall that in Figure 1(b), we wanted to use the social links between pair of authors as an additional piece of information to infer the category of NPs. Since Twitter is a community-based type of social media platform we can group different users into communities as shown in Figure 4. Each community talks about a certain topic (e.g. politics, sports). Based on the community membership of the users, and the topic preferences of the communities, we can derive a feature vector for the author - community - NP triplet that contains more information than the author - NP feature vector by NP+LDA.

Refer to Figure 3(b) for the NP+RART model modified for noun phrases. NP+RART is a hybrid of our NP+LDA and Role Author Recipient Topic Model (RART) from McCallum et al. [7]. Similar to NP+LDA, each noun phrase is generated by a single latent variable  $z_{t,n}$ . The NP+RART model has another set of latent variables  $x_i$  and  $x_j$ . The latent variable  $x_i$  denotes the community membership of the

author who retweets and  $x_j$  represents the community membership of the author who is retweeted. Given a tweet, the  $x_i$  and  $x_j$  are constrained to be the same for all the noun phrases in the same tweet. This helps to ensure that authors always act in the same “role” when writing or retweeting. Based on the small average number of 3.7 noun phrases for each tweet, it should be reasonable to assume that authors always act in the same “role” in a single tweet. The generative process is as follows, 1) Each topic  $z$  samples the word distribution from a symmetric Dirichlet distribution. 2) For each Retweet  $t$ , the retweeter  $i$  and the original author  $j$  samples their community membership from a symmetric Dirichlet distribution. 3) For each Retweet, the retweeter  $i$  and the original author  $j$  samples a latent community membership variable  $x_{t,i}$  and  $x_{t,j}$  from their own community membership distribution  $\pi_i$  and  $\pi_j$  respectively. The pair of sampled community membership refers to a block out of  $H \times H$  number of blocks in the model. 4) Each block (g,h) samples their topic distribution from a symmetric Dirichlet distribution. 5) For each noun phrase  $n$  in retweet  $t$ , the block (g,h) samples a single latent variable  $z_{t,n}$  for the noun phrase  $n$  from the topic distribution of the block (g,h),  $\theta_{g,h}$ . 6) The block (g, h) then samples each noun phrase  $n$  for tweet  $t$  using a single latent variable  $z_{t,n} = k$  and the topic word distribution  $\phi_{k,v}$ .

### 3.5 Feature Vectors for Classification

Given noun phrase  $n$  from tweet  $t$ , we can derive the feature vector  $\psi_n$  to represent  $n$  in the latent space.  $\psi_n$  is the  $K$  dimensional feature vector which we can derive from NP+LDA or NP+RART, where  $K$  is the number of topics used in both models. In  $\psi_n$ , each element  $\psi_{n,k}$  corresponds to the probability of  $n$  belonging to the topic of  $k$ . In other words,  $\psi_n$  is a normalized probability vector which sums to 1, derived from the parameters of NP+LDA ( $\theta, \phi$ ) and NP+RART ( $\pi, \theta, \phi$ ).

Given that the classification step is separate from the inference process of NP+LDA and NP+RART, we can use any classifier such as Support Vector Machines, Decision Trees, K-Nearest Neighbor etc. We have chosen to use Logistic Regression because its probabilistic nature could allow for joint inference with the probabilistic topic models of NP+LDA/NP+RART in future work.

## 4. EXPERIMENTS

### 4.1 Data set

We developed our Noun Phrase recognition on a small collection of tweets which we crawled from Twitter. We first selected a core set of 16 Twitter users mainly consisting of American politicians from Democratic Party and Republican Party. We expanded this core set of users by including the users who follow at least 3 out of these 16 core users. In total, we have 171,384 users from this expanded set. Then starting from September 28th 2011 (09/28/2011), we obtained 3,200 tweets that were published before 09/28/2011 and continue crawling their tweets daily until December 31th 2011 (12/31/2011). Over a period of approximately 3 months, we ended our crawl with a total of 110,667,693 tweets.

As many of such tweets describe mundane activities and gibberish text, we only retained tweets that have been retweeted and whose original authors are from the expanded set. Af-

ter filtering, we obtained a small sample of 7,414,570 tweets with an average of 18.0 words per tweet.

## 4.2 Noun Phrases from NP+RART

We derive the feature vectors from NP+RART for the NPs by using 50 topics and 100 communities.

**Table 1: List of Ten Most Probable Political Noun Phrases From NP+RART**

Noun Phrase	Political Prob
passage of balanced budget amdt	0.952
passage of a balanced budget amdt	0.952
gop co-chair of deficit commission decries inability of washington	0.952
boehner debt deal plan	0.952
balanced budget amdt	0.952
a balanced budget amdt	0.952
support reid debt limit	0.952
short-term debt deal	0.952
oppose balanced budget	0.952
house rejects reids debt ceiling proposal	0.952

Table 1 shows the top ten list of NPs returned by the classifier with features from NP+RART. The list of NPs from Table 1 do indicate that NP+RART is able to classify political NPs. To compare the performance of NP+RART against NP+LDA, we sampled 200 NPs on which NP+RART and NP+LDA disagree. We manually annotate the disagreed set of NPs (disagreement set). The annotation results showed that NP+RART has a better accuracy than NP+LDA in classifying political NPs. We find that whenever NP+LDA or NP+RART classifies a NP as political, the probability of NP+LDA being correct is in the range of [0.604, 0.8], while NP+RART has a probability range of [0.725, 0.921].

## 5. CONCLUSION

We proposed a noun phrase (NP) classification method using the authors' behavior and social information in Twitter. We showed that our first model NP+LDA could classify NPs into political categories more accurately than state of the art tagger (SentReg) based on a large knowledge base. By incorporating community information in NP+RART, we further improve the accuracy of our classifier. Finally, we show that we are also able to classify sports NPs in a political dominated Twitter data set.

There are many possible extensions from here. Joint inference between the classification and topic models is one area to explore. Another possibility is to find categories that are likely to co-occur in the same tweet, then use collective classification to classify the NPs base on the categories of neighboring NPs in the same tweet.

## 6. ACKNOWLEDGEMENTS

This research/project is supported by the Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Programme Office. The research is also partly funded by Google. Freddy will like to thank Nelman Lubis Ibrahim for his help in crawling the data, Ramnath Balasubramanyan, Bryan Kisiel, Partha Talukdar, Kathryn Rivard,

Dana Movshovitz-Attias, Brendan O'Connor and all other members of the Read the Web team for their help during his stay at CMU.

## 7. REFERENCES

- [1] D. Blei and J. McAuliffe. Supervised topic models. In *Advances in Neural Information Processing Systems 20*, pages 121–128. MIT Press, Cambridge, MA, 2008.
- [2] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. H. Jr., and T. M. Mitchell. Toward an architecture for never-ending language learning. In *AAAI 2010*, 2010.
- [3] A. Carlson, J. Betteridge, R. C. Wang, E. R. Hruschka, Jr., and T. M. Mitchell. Coupled semi-supervised learning for information extraction. *WSDM '10*, pages 101–110, New York, NY, USA, 2010. ACM.
- [4] K. Gimpel, N. Schneider, B. O'Connor, D. Das, D. Mills, J. Eisenstein, M. Heilman, D. Yogatama, J. Flanigan, and N. A. Smith. Part-of-speech tagging for twitter: annotation, features, and experiments. *HLT '11*, pages 42–47, Stroudsburg, PA, USA, 2011. ACL.
- [5] C. L. Li, J. Weng, Q. He, Y. Yao, A. Datta, A. Sun, and B.-S. Lee. Twiner: Unsupervised named entity recognition in targeted twitter stream. *SIGIR '12*, New York, NY, USA, 2012. ACM.
- [6] X. Liu, S. Zhang, F. Wei, and M. Zhou. Recognizing named entities in tweets. *HLT '11*, pages 359–367, Stroudsburg, PA, USA, 2011. ACL.
- [7] A. McCallum, X. Wang, and A. Corrada-Emmanuel. Topic and role discovery in social networks with experiments on enron and academic email. *J. Artif. Int. Res.*, 30:249–272, October 2007.
- [8] M. Michelson and S. A. Macskassy. Discovering users' topics of interest on twitter: a first look. *AND '10*, pages 73–80, New York, NY, USA, 2010. ACM.
- [9] A. Ritter, S. Clark, Mausam, and O. Etzioni. Named entity recognition in tweets: an experimental study. *EMNLP '11*, pages 1524–1534, Stroudsburg, PA, USA, 2011. ACL.
- [10] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. *WWW '10*, pages 851–860, New York, NY, USA, 2010. ACM.
- [11] H. M. Wallach. Topic modeling: beyond bag-of-words. In *Proceedings of the 23rd international conference on Machine learning*, ICML '06, New York, NY, USA, 2006. ACM.
- [12] C. Wang, D. M. Blei, and F.-F. Li. Simultaneous image classification and annotation. In *CVPR'09*, pages 1903–1910, 2009.
- [13] J. Weng, E.-P. Lim, J. Jiang, and Q. He. Twitterrank: finding topic-sensitive influential twitterers. *WSDM '10*, pages 261–270, New York, NY, USA, 2010. ACM.
- [14] C. Whitelaw, A. Kehlenbeck, N. Petrovic, and L. Ungar. Web-scale named entity recognition. *CIKM '08*, pages 123–132, New York, NY, USA, 2008. ACM.