Detecting Life Events in Feeds from Twitter

Barbara Di Eugenio
University of Illinois at Chicago
Chicago, Illinois, USA
Email: bdieugen@uic.edu

Nick Green
University of Illinois at Chicago
Chicago, Illinois, USA
Email: ngreen21@uic.edu

Rajen Subba
San Francisco Microsoft Office
San Francisco, CA, USA
Email: rsubba4@gmail.com

***Abstract*— Short posts on micro-blogs are characterized by high ambiguity and non-standard language. We focus on detecting life events from such micro-blogs, a type of event which have not been paid much attention so far. We discuss the corpus we assembled and our experiments. Simpler models based on unigrams perform better than models that include history, number of retweets and semantic roles.**

I. Introduction

Microblogging has become a popular tool for communication. Services like Twitter, Facebook and Y! Pulse are used by millions of users to share information and express their opinions on a variety of topics. Much attention has been paid to detecting events centered around celebrities [1] or real-world events such as earthquakes, floods, elections [2], [3], [4]. However, while blogging, users also divulge personal information on what we call "life events" – marriage, birth of a child, graduation, losing or getting a job. Interest is rising in recognizing events relevant to individuals who are otherwise "anonymous", such as bullying [5], in order to provide the posters with relevant, personalized services. For example, marketers know that people mostly shop based on habits, but that among the most likely times to break those habits is when a major life event happens [6].

All information extraction tasks on microblogs need to confront their peculiarities. Twitter users only have 140 characters to make a post, which results in extremely concise and ambiguous language. The informality of the posts leads to an abundance of spelling mistakes, organic acronyms and emoticons. However, detecting life events is more challenging than detecting events centered around celebrities or real-world events, which can take advantage of other features from the blogosphere, for example, the known entity "buzziness" for the day, or the temporal and geographical distributions of a large number of tweets all pertaining to the same event. Clearly, when an individual tweets about their own life or their friends' lives, all that is available is the tweets themselves and their structure as concerns replies, retweeting etc.

In this paper, we present our ongoing work on detecting life events from tweets – we have so far focused on *marriage* and *employment*. We first discuss the corpus we assembled, and then the experiments we ran on a variety of features. Our results show that features capturing either a deeper analysis of the tweets (semantic roles) or the larger context (tweet history and number of retweets) do not improve performance over a simple unigram model. Still, we believe semantic roles may be conducive to better performance, if we can improve the performance of the semantic role labeller itself.

II. Corpus

We gathered our corpus via the Twitter Streaming API [7], which allows downloading publicly available tweets. To reduce trending, we collected the data over a couple of weeks at roughly three hour intervals. We downloaded about 1 million tweets, restricted to originate in the United States only. This acts as a good initial filter of foreign languages since for the moment we are only interested in American English. It is important to note that no corpus previously existed for this sort of work, and that for the reasons outlined in Section I, neither is such a corpus easy to gather by focusing on certain dates, or hashtags.

Naturally, a corpus of 1M tweets is too large for manual inspection. We needed to identify a nucleus of relevant examples for the two life events of interest. We follow an approach similar to [5], in picking several keywords that reflect the domain(s) of interest and then filter out all tweets that do not contain such keywords. As they mention, this is at the cost of some sampling bias. Since one motivation for our work is targeted advertising, sampling bias is sometimes inherent in the approach itself: in practical use cases, clients provide only keywords to express what they are looking for and the objective is to mine social media for users who are or have been experiencing these life stages. For *employment*, we chose these keywords: *new job, laid off, interview, job offer*. For *marriage*, we went one step further than [5]. We selected the top three relevant keywords (after removal of obvious out-of-domain words such as country names) as ranked by using TF-IDF on multiple documents gathered by mining domain specific websites (brides.com; weddingstylemagazine.com; weddingsmagazine.com; theknot.com; insideweddings.com). The top three keywords turned out to be: *engaged, married, wedding*.

The result of applying the keyword filter reduced the dataset to approximately 0.5% in total, i.e., 4395 tweets. We then added a ranking mechanism to allow coding of the most relevant ones first. Ranking was based on well-formedness of the tweet and very simple spam detection. Firstly, some tweets contain very few intelligible words, for example `iWann Kno If We Still Gon Tew Tha #Wedding.*!?`. We calculated a spelling score using the dictionary library Hunspell [8]. Tweets that had a poor spelling rating (over half of words were not well-formed English) were assigned a lower ranking

Classification of Tweet	Example	Count
(e.a) Affecting the tweeter	Wearn a suit tomorrow for da interview ;+) dats how serious da money is	440
(e.b) Affecting another entity	@MRZLusHious I hope you get it! What time is your interview!?	58
(e.c) Relevant but general statement	Do you normally wear a headband to a job interview?	1
(e.neg) No relevance to the life event	Man im finna make these **** love me in this interview	691

TABLE I
TYPES OF CLASSIFICATION FOR THE EMPLOYMENT CATEGORY.

Classification of Tweet	Example	Count
(m.a) Affecting the tweeter	Omg I love that man we getting married	365
(m.b) Affecting another entity	Getting ready for the murder mystery wedding tonight!	126
(m.c) Affected the tweeter	But me and Amanda finally got married tho	42
(m.d) Affected another tweeter	HES HAWT .. but married.	82
(m.e) Negative of the life event	Is it crazy that I don't Eva wanna get married??	72
(m.f) Relevant but general statement	When I get married I definitely wanna be on bridezillas...	295
(m.neg) No relevance to the life event	married to the money...a #truelovestory	69

TABLE II
TYPES OF CLASSIFICATION FOR THE MARRIAGE CATEGORY.

score. This also reduces the priority of tweets that may not be in English. Secondly, a symptom of spam in social media is links to other websites (even if some links are of course legitimate). So we reduced the ranking of posts which included URLs. After these simple measures, about 20% of tweets were demoted in the dataset.

For the experiments we report here, we manually annotated 2250 of the highly ranked tweets. Initially we only envisioned a two or three-way classification: for example for marriage, *YES-Tweeter* (the tweeter is or is getting married), *YES-Other* (somebody else is or is getting married), *NO*. While performing the annotation, it became clear that a more fine-grained classification was needed, as detailed in Tables I and II. The datasets were double coded in their entirety. Intercoder agreement is acceptable on marriage ($\kappa = 0.72$) and excellent on employment ($\kappa = 0.88$) [9], [10].

III. EXPERIMENTS

We created several datasets which differ as concerns which type of tweets is considered as positive. In some cases, we added negative examples picked randomly from the earlier filtered out data, to ensure enough negative examples, especially for *marriage*, and to allow negative examples that did *not* include the keywords of interest.

For **employment**, we define the following datasets (see Table I for category definition):

• EmploymentA - *e.a* + *e.b*, i.e., someone seeking or having got employment (negative: *e.c* + *e.neg*)

• EmploymentB - *e.a*, i.e. tweeter is looking for/found employment (negative: *e.b* + *e.c* + *e.neg*)

• EmploymentC - Employment A with an extra 500 random negative tweets

• EmploymentD - Employment B with an extra 500 random negative tweets

For the **marriage** dataset (see Table II):

• MarriageA - *m.a to m.d*, i.e. someone is or is getting married (negative: *m.e* through *m.neg*)

• MarriageB - *m.a to m.d* + *m.f*, i.e. someone is or is getting married, plus general statements (negative: *m.e* + *m.neg* + 500 random negative tweets)

• MarriageC - Marriage A with an extra 500 random negative tweets.

Employment and Marriage datasets A and C mirror each other, but not Employment B/D and Marriage B. This is due to different distributions, e.g. as concerns general statements: one for employment, but 295 for marriage.

A. Unigram Models

Our simplest model is a unigram, bag-of-words model. It turns out it is also the most effective so far. We trained and tested models with 10-fold cross-validation. A large volume of algorithms were experimented with, along with variable parameters where available, as implemented in Weka [11]. These overall results are shown in Tables III and IV. Despite variation among the algorithms, the top two performing were Complement Naive Bayes (CNB) [12] and Support Vector Machine (SVM). More specifically for SVM, we experimented with numerous kernels and 10 different C parameters, ranging from 0.1 to 10.0. Based on this, we selected the Poly Kernel and a C parameter of 1.0 as our global parameter set for SVM due to it yielding the best overall performance over all datasets in our cross-validation experiments.

Weka's SVM implementation can return classification probabilities when the output is fitted to a logistic regression model. Given this, we analyzed the distribution of correct and incorrect classifications against this probability. If SVM gave a score close to 1.0, it had a high probability of correct classification, whereas its score was lower on the incorrectly classified results. We then investigated a back-off method composed of SVM and CNB: we disregard SVM classifications below a probability threshold of 0.9, and reclassify those data

Classifier	Set A	Set B	Set C	Set D
Baseline Maj. Class	0.580	0.620	0.700	0.730
Random Forest	0.767	0.765	0.852	0.847
Decision Table	0.714	0.715	0.790	0.802
Bayesian Network	0.760	0.771	0.817	0.837
Naive Bayes	0.800	0.802	0.839	0.844
CNB	0.836	0.861	0.841	0.860
SVM	0.788	0.820	0.857	0.862
SVM + CNB	**0.86**	**0.863**	**0.902**	**0.896**

TABLE III
ACCURACY ON EMPLOYMENT DATASETS

Tweet	**Wrong Class**
1. well unless its an interview, but other than that im not friendly !!	Positive
2. @YujChung interview attire shopping???	Positive
3. Bob Marley Interview	Positive
4. Don't Miss tomorrow my #Interview with #Radio538 at 9:00 !!! #trance	Positive
5. What do i wear for my Co-op interview?	Negative
6. I look the bomb for this interview.	Negative
7. interview at target at 11	Negative

TABLE V
INCORRECT CLASSIFICATIONS ON EMPLOYMENT

points via CNB. For employment, this composite classifier (bottom row of Table III) performs significantly better than any other algorithm other than SVM and CNB (χ^2 shows that differences ≥ 0.05 in accuracy are statistically significant). As concerns SVM or CNB, the results are better but not always significantly so.

The results on marriage are more mixed (Table IV). First, CNB does not perform as well on marriage as on employment, whereas SVM does. Not surprisingly then, the composite classifier does not gain as much, in fact, it shows some significant degradation in performance within set C with respect to SVM.

Classifier	Set A	Set B	Set C
Baseline Majority Class	0.580	0.630	0.580
Random Forest	0.694	0.769	0.723
Decision Table	0.644	0.933	0.777
Bayesian Network	0.643	0.867	0.741
Naive Bayes	0.694	0.826	0.724
CNB	0.715	0.764	0.693
SVM	0.729	**0.949**	**0.799**
SVM + CNB	**0.731**	0.935	0.757

TABLE IV
ACCURACY ON MARRIAGE DATASETS

To uncover potential misclassification regularities, we performed error analysis. The set of misclassified tweets shows that there was no single point of failure (see examples of misclassified tweets in Table V). Many tweets *(rows 1, 2, 5 and 6)* are ambiguous; the correct interpretation may be clarified by the larger context. Row 3 is intuitively negative given general knowledge about *Bob Marley*. Row 7 is yet another case of ambiguity that requires localized general knowledge; to many, a *target* could be considered something to aim for, however, it is also the name of a large US department store, which is what the tweeter is most likely describing.

Employment C, D, and Marriage B, C, reflect targeted advertising, namely, the need to classify a new tweet in real-time: accuracy on true negatives is more important than accuracy on true positive. For all sets other than Marriage B, accuracy on the negative class is higher than on the positive class, and always higher than 0.925. Marriage B and C differ on *m.f*, relevant general statements. We will investigate whether these general statements can be recognized separately.

B. Beyond unigrams

Since a unigram model is so simple, intuition is that more sophisticated features should help. To start with, we investigated a simple probabilistic bigram model using the text within the tweets. Our preliminary results showed little promise. We then proceeded to consider features that provide a deeper linguistic insight and features that relate to context.

a) Part of Speech: Whereas POS tagging is a staple for many natural language processing (NLP) tasks, the informal nature of tweets proves problematic for conventional POS taggers. Even Twitter specific taggers [13] have shortfalls, e.g., it tags hashtags as such. However, hashtags often function as linguistic POSs, e.g. see #Interview in Table V. Hence, we did not run experiments with POS tags.

b) Semantic Role Labeling (SRL): For targeted advertising, it is very important to know "who does what": SRL [14] assigns roles to the semantic arguments associated with the predicate. E.g., according to PropBank [15], for *marry* Arg0 is the *causer*, whereas Arg1 and Arg2 represent the two married people. The ClearNLP labeler [16] was applied to our tweet data and several features were used to build additional models: *Presence of Arg0/Arg1/Arg2; Number of roots; Number of conjunctions* (the latter two distinguish tweets that contain more than one main verb).

There was no improvement as concerns Employment. The few improvements in accuracy for Marriage (bold in Table VI) are not significant. Nonetheless, we believe these results are encouraging since they are obtained with extremely noisy SRL results: on a random sample of 58 tweets, SRL correctly recognized only 56% of the predicates, and for those, it assigned correct A0s and A1s 78% of the time. Whereas running the experiments with gold-standard SR labels is not feasible, since it would mean to tag all tweets by hand, in the future we intend to use a semi-automatic approach and a retrainable SRL such as SWIRL [17].

c) Adding context: History and Retweets.: We also explored features that relate to context, which is often considered essential for many disambiguating tasks, but they did not improve performance either. For tweets, one type of context

Classifier	Set A	Set B	Set C
Decision Table	**0.669**	0.835	0.729
Bayesian Network	0.637	0.831	0.685
Naive Bayes	0.689	0.815	0.680
CNB	0.711	**0.768**	**0.703**
SVM	0.714	0.874	0.750
SVM + CNB	0.712	0.883	**0.762**

TABLE VI
MARRIAGE: ACCURACY WITH SRL FEATURES

Classifier	No History	History	History + Prefix
Decision Table	0.790	0.787	0.794
Naive Bayes	0.839	0.851	0.837
CNB	0.841	0.858	0.856
SVM	0.857	0.844	0.841
SVM-CNB	**0.902**	0.880	0.828

TABLE VII
EMPLOYMENT C: PRIOR TWEETS INCLUDED

is whether a tweet is in reply to another tweet (history), or whether it is replied to. For us, tweet history is more interesting, since we need to perform live classification on the current tweet, rather than wait for replies to the current tweet. Additionally, the streaming API can supply the parent tweet for any tweet, but not the child tweets.

We built a tree of all target tweets for both categories: around 25% of all tweets were in reply to a prior tweet (27% for marriage and 26% for employment). We reran all our experiments with this additional information. We added the previous tweet, along with the tweet of interest, to the bag of words; in a second setting, we added to the bag of words each token tagged with a prefix, to distinguish between the token being part of the current or previous tweet.

Our experiments did not show improvement with respect to the simpler unigram models. To illustrate, Table VII shows results on Employment dataset C (the first column repeats entries from the C column from Table III). The addition of history had little impact on performance; rather, the only significant change was performance loss, when "history + prefix" was used with SVM+CNB. More sophisticated relationships between the current tweet and previous tweets, for example discourse relations, may be effective, but they require manual annotation, and it is not clear they help classification [18]. History looks back to the past of the tweet, whereas e.g. retweet counts (included in the tweet meta-data) look to the future. Retweet counts do not help either. When added to our unigram + history models, we observed degradation in accuracy, either minor or significant, for all algorithms.

IV. FUTURE WORK

We have presented experiments on detecting life-events of a user from micro-blog posts. Contrary to our expectations, the simplest bag-of-words approach performs with the highest accuracy: adding "deeper" information did not improve performance. However, as concerns SRL, results are encouraging given the noisy performance of the SR labeller. One possible direction is to normalize the tweets [19]. The other is to use a semi-automatic approach and a retrainable SRL such as SWIRL [17].

Acknowledgements: This work was initially supported by a Yahoo! Faculty Research and Engagement award. We thank Jayashree Khobarekar for her help with coding and performing SRL experiments.

REFERENCES

[1] A. Popescu, M. Pennacchiotti, and D. Paranjpe, "Extracting events and event descriptions from twitter," in *WWW11 - Proceedings of the 20th International Conference on the World Wide Web – Companion Volume*. ACM, 2011, pp. 105–106.
[2] T. Sakaki, M. Okazaki, and Y. Matsuo, "Earthquake shakes Twitter users: real-time event detection by social sensors," in *WWW10- Proceedings of the 19th International Conference on the World Wide Web*. ACM, 2010, pp. 851–860.
[3] H. Becker, M. Naaman, and L. Gravano, "Beyond trending topics: Real-world event identification on Twitter," in *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media (ICWSM11)*, 2011.
[4] E. Tjong Kim Sang and J. Bos, "Predicting the 2011 dutch senate election results with twitter," in *Proceedings of the Workshop on Semantic Analysis in Social Media*, Avignon, France, April 2012, pp. 53–60.
[5] J.-M. Xu, K.-S. Jun, X. Zhu, and A. Bellmore, "Learning from bullying traces in social media," in *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics*, 2012, pp. 656–666.
[6] C. Duhigg, "How companies learn your secrets," *New York Times*, February 16 2012, http://www.nytimes.com/2012/02/19/magazine/shopping-habits.html. Last accessed Dec 7, 2012.
[7] "Twitter streaming api," https://dev.twitter.com/docs/streaming-api, 2012.
[8] "Hunspell," http://hunspell.sourceforge.net/, 2011.
[9] J. Carletta, "Assessing agreement on classification tasks: the Kappa statistic," *Computational Linguistics*, vol. 22, no. 2, pp. 249–254, 1996.
[10] B. Di Eugenio and M. Glass, "The Kappa statistic: a second look," *Computational Linguistics*, vol. 30, no. 1, pp. 95–101, 2004, squib.
[11] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: An update," *SIGKDD Explorations*, vol. 11, no. 1, 2009.
[12] J. Rennie, L. Shih, J. Teevan, and D. Karger, "Tackling the poor assumptions of naive bayes text classifiers," in *In Proceedings of the Twentieth International Conference on Machine Learning*, 2003.
[13] 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," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, 2011, pp. 42–47.
[14] L. Màrquez, X. Carreras, K. C. Litkowski, and S. Stevenson, "Semantic role labeling: an introduction to the special issue," *Computational Linguistics*, vol. 34, no. 2, pp. 145–159, 2008.
[15] M. Palmer, D. Gildea, and P. Kingsbury, "The proposition bank: An annotated corpus of semantic roles," *Computational Linguistics*, vol. 31, no. 1, pp. 71–105, March 2005.
[16] J. D. Choi and M. Palmer, "Transition-based semantic role labeling using predicate argument clustering," in *Proceedings of the ACL Workshop on Relational Models of Semantics (RELMS'11)*, 2011, pp. 37–45.
[17] M. Surdeanu, L. Màrquez, X. Carreras, and P. Comas, "Combination strategies for semantic role labeling," *Journal of Artificial Intelligence Research*, vol. 29, no. 1, pp. 105–151, 2007.
[18] A. Wang, T. Chen, and M.-Y. Kan, "Re-tweeting from a linguistic perspective," in *Proceedings of the Second Workshop on Language in Social Media*, Montréal, Canada, June 2012, pp. 46–55.
[19] M. Kaufmann and J. Kalita, "Syntactic normalization of twitter messages," in *International Conference on Natural Language Processing, Kharagpur, India*, 2010.