# Automated Social Group Modeling for Characterizing and Forecasting Elections

Aravindan Mahendiran

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Computer Science and Applications

Naren Ramakrishnan, Chair
Cal Ribbens
Aditya Prakash

January 10, 2014
Blacksburg, Virginia

Keywords: Forecasting, Twitter, Query Expansion, Elections, Probabilistic Soft Logic

# Automated Social Group Modeling for Characterizing and Forecasting Elections

Aravindan Mahendiran

(ABSTRACT)

Twitter mining over the last few years has garnered a lot of attention from the research community. Strong correlations have been shown between Twitter *mentions* and stock markets, book sales and flu outbreaks which is then used for forecasting. Even though such methodologies are accurate in forecasting the trends to a great extent, their performance is dictated by the domain specific vocabulary is used to track the relevant tweets. Such a vocabulary is usually provided by subject matter experts but is not exhaustive. The language used in Twitter is drastically different from other forms of writing like those in news articles or even web blogs. It constantly evolves with time as users adopt popular hash-tags to express their opinion. Thus, the vocabulary used by the forecasting algorithms needs to be dynamic in nature and should capture the rising trends of the domain. Otherwise, the prediction algorithms miss out on capturing the some of the most informative documents.

We propose a novel unsupervised learning algorithm builds a vocabulary through modeling user preferences by exploiting the explicit and latent structure in such data sets. We use Probabilistic Soft Logic, a framework for probabilistic reasoning over relational domains, to develop a query expansion algorithm that learns such a dynamic vocabulary for any given domain. Using 7 presidential elections from Latin America we show how such a query expansion methodology improves the recall and accuracy of two state of the art election prediction algorithms. Through this approach we achieve close to 2x increase in the number of documents used for predictions and 16.13% reduction in the prediction error.

# Dedication

*To my awesomest brother, mother and friends*

# Acknowledgments

First and foremost I would like to thank my adviser Dr.Naren Ramakrishnan. He was not only instrumental in kindling my interest in machine learning and data mining but is also my mentor in many ways.

I would like to thank my fellow graduate students Nabeel Mohammed, Rupinder Paul Khandpur and Sathappan Muthiah. Their help and support were invaluable through my graduate studies.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

The last decade has seen a massive explosion of on-line data in all forms be it news articles, blogs or social media like Twitter, Facebook and MySpace. Twitter is a novel micro-blogging service and was launched in 2006. Twitter users post messages called tweets on a public message board and these tweets are limited to 140 characters. Originally the tweets were meant to be personal status updates but over the years these tweets have evolved into much more. Now apart from simple status updates, tweets can be URLs websites or even directed messages to particular individuals. Due to the short nature of the messages users often combine multiple words into *hash-tags* to convey their views. Therefore, these hash-tags become the most important part of a tweet as often the entire essence of the tweet is captured in single hash-tag. These hash-tags evolve over time and gain more traction as users adopt the popular ones. This makes the language used in Twitter very different from other textual web content like blogs and articles.

Today twitter has grown so big[1] that it has come to be looked at

[1] As of May 7,2013 twitter has 555 million active registered users with 135000 new users signing up everyday and approximately 1 billion tweets created every 5 days

as a treasure trove of mine-able data. With official APIs that are open to public, the easy access to large volumes of data has piqued the interest of scientists in the data mining community. Researchers have studied various real world phenomenon like book sales [1], box office earnings [2] and even stock prices [3] and have not only shown strong correlations to the chatter on Twitter but were also able to make forecasts about future trends too.

The more curious research though is whether the on-line chatter can be used to model the social, economic or political landscape of a country. Political leaders have started using Twitter as a channel to mobilize supporters for their ideologies. For instance, during the 2008 US presidential election Barack Obama used social media and specifically Twitter extensively in his campaign. His victory established Twitter as a channel to garner support for a particular ideology be it political or otherwise. Bollen et al. in their research [4] used a version of the well-established psychometric instrument- Profile of Mood States(POMS) to model the mood of Twitter traffic and found correlations to a number of social and economic events that occurred during the same time period. The results from this research instigated more researchers to study and quantify the political sentiment through social media and if possible even forecast elections.

However, there is a constant debate among political scientists on whether Twitter can be used as a surrogate for political opinion of the masses. Some believe twitter indeed is an indicator of political opinions while others question the validity of such results.. In this work we aim to answer these questions by modeling online social groups and improving the performance of election prediction algorithms that use Twitter. In the following section we review the current state of the art approaches to election prediction using Twitter..

## 1.1 Related Work

We divide the literature review into three parts. First we look at a selection of volume based approaches to predict elections i.e., models that predict election results by merely counting the number of times a particular candidate is mentioned on Twitter. Then we review more sophisticated approaches that model the demographics of an election to make more informed predictions. Lastly, we shall summarize a quite prevalent pessimistic view on such methodologies' capability to predict elections.

### 1.1.1 Volume based approaches

In one of the most cited papers in this space, [5] the authors claim that *"The mere number of tweets reflect voter preferences and comes close to traditional polls.."* while predicting the 2010 German federal election. They go on to strongly conclude that Twitter can indeed be a valid indicator of political opinion. This was followed by [6, 7, 8, 9] all of which use volume based approaches combined with sentiment analysis. Both [6, 8] fit a regression model to opinion polls with volume of mentions and sentiment as independent variables and the opinion polls as the dependent variable. They conclude that

sentiment is a weak predictor compared to share of volume.
In general the methodologies described in these publications count the occurrence of certain hand crafted keywords from the ”Twittersphere” and classify such tweets as positive or negative using a classifier trained on human annotated lexicons.

### 1.1.2 Profile Modelling

More sophisticated approaches are adapted in [10, 11, 12]. The authors either model the candidates or the voters in the elections rather than compute the aggregated sentiment of the mass. In [11] the authors build a Support Vector Machine classifier trained on manually labeled tweets and classify users into ’left’ and ’right’ aligned. Through latent semantic analysis they claim to have identified the hidden structure in the data that is strongly associated with the users’ political affiliations. Livne et al. in [10] analyze the Twitter profiles of candidates who contesting in the 2010 mid-term elections in the U.S. They identify topics specific to groups of candidates, split according to their known political orientations and use the features obtained as inputs to a regression model to predict the elections. In a similar technique Diaz-Aviles in [12] model the candidates by building a emotional vector for each candidate by using the mentions of that candidate and sentiments associated with each mention learned using the NRC EmotionLexicon(EmoLex). They then use these profiles learned to predict the rise and fall of a candidate’s popularity.
In another research, Mustafaraj et al. [13] model the distribution of political content among Twitter users. They divide the users into two groups the ”vocal minority” and the ”silent majority”. They observe

that these two groups engage in different ways in social media. The vocal minority aim to broaden the impact of tweets by re-tweeting and linking to other web-content whereas the silent majority who tweet significantly lesser are more inclined to share their personal view points. Though they do not make any predictions about elections, they make very valid observations such as *"Because of this differences between content generated by different groups , one should be aware of aggregating data and building models upon them, without verifying the underlying model that has generated the data."*.

### 1.1.3 Flaws in current state of the art

Of late there has been a lot of studies showing how such models that predict elections using social media feeds are flawed [14, 15, 16, 17]. These publications not only list the obvious issues in using Twitter to predict elections but also detail recommendations on how to make such methodologies better.

Daniel Gayo-Avello surveys almost all the state of the art approaches in predicting elections in his paper [15] most of which are detailed above. According to him post-hoc analysis of elections in retrospect must not count as valid predictions and that researchers do not report negative results leading to what is called the *file drawer* effect. His major points of argument against such models are:

- The models are tailor made to fit a particular election and that they need to be generic enough to reproduce similar results when run on other elections. In particular Metaxas et al in [14] state that any method claiming predictive power on the basis of Twitter data should be a clearly defined algorithm and should be "explainable" i.e., black box approaches should be avoided.

- There is no predefined notion of "vote" that has been used to predict the elections. Most of the models aim to predict elections merely by counting the tweets related to a candidate.

- Biases in Twitter are ignored. Twitter is not a representative sample of the electorate demographic as not every age gender or social group is represented. He also notes that since people tweet on a voluntary basis the data produced is only by those who are politically active. Another point of contention is the credibility of tweets i.e., whether the tweets are rumors, campaign propaganda or contain misleading information just to maliciously attack candidate's on-line popularity.

- Since in 2008 and 2010 , 91.6% and 84% of elections were won by the the incumbent candidate respectively, Gayo-Avello argues that incumbency should be the baseline rather than just chance. He also notes that most of the methodologies are only slightly better than chance.

- Lastly he states even though sentiment classifiers are highly researched space in Natural Language Processing, the accuracy of such methods are only slightly better than random classifiers. Further, these classifiers do not detect humor and sarcasm which in his opinion plays a major role in political discussions.

## 1.2 Motivation

The one thing that the previously described prediction methodologies have in common is the process of filtering tweets that match a vocabulary to obtain those that pertain to the particular election. Usually this is done by a process of querying the Twitter API for a set of keywords such as the names of the candidate and political parties contesting the elections. Given that the language used on Twitter is completely different from the language in newspapers and magazines, this process gives a very low recall.[*Aravind: Recall is used as a place holder. How to word this better?*] For example, for the 2012 presidential election in Venezuela users preferred the hash-tags *#elmundocochavez* and *#hayuncamino* to show their support for Hugo Chavez and Henrique Capriles respectively. Such hash-tags are not known a priori and gain more traction and adaptation closer to the election. Querying just for "chavez" or "capriles" would result in missing out on a huge chunk of tweets that are indicative of a user's political preference. Thus, it becomes vital that any methodology that predicts elections accounts for such memes that become popular during the time period leading up to the election.
In this work we build on our earlier work [18] and address this issue by:

- Designing and implementing a new dynamic query expansion algorithm using Probabilistic Soft Logic to obtain a exhaustive vocabulary.
- Showing how the vocabulary obtained from the Dynamic Query Expansion exercise improves the recall[*Aravind: again place-*

*holder*] and accuracy of the prediction algorithms.

## 1.3 Document Overview

The rest of the document is organized as follows:
In the next chapter, we outline the Probabilistic Soft Logic framework and how we take advantage of this domain specific language to build our dynamic query expansion algorithm.
In the third chapter, we detail two state of the art algorithms used to predict elections.
We then detail our experiments and present the results confirming our hypothesis in the fourth chapter.
In the final chapter, we make our final conclusions and present the road map for future work.

# Chapter 2

# Dynamic Query Expansion

## 2.1 Query Expansion

In most document corpora, a single concept can be referred using multiple terms. In information retrieval (IR) this is called *synonymy* and has a huge impact on the recall of documents pertaining to the concept. Researchers address this problem by creating as exhaustive a query as possible. But when exploring the Twitter corpora it becomes almost impossible to hand craft such a expansive query as the meme and hash-tag adaptations are not known a priori.
To address this issue IR experts use *query expansion* or reinforcement learning. These are iterative algorithms that are initialized with a small set of query terms. When the documents matching the query terms are returned, after basic processing such as tokenization, stop-word removal and stemming a richer vocabulary is obtained by ranking the terms in these documents by their frequency counts. The top words from this list is then used to query the documents again. This process goes on until no new terms are added to the vocabulary. We design and implement such an pipeline using Probabilistic Soft

Logic to build our vocabulary for a given election. We first review the PSL framework before proceeding to detail our methodology.

## 2.2 Probabilistic Soft Logic

Probabilistic Soft Logic [19] is a framework for collective probabilistic reasoning on relational domains. PSL models have been developed in various domains, including collective classification [20], ontology alignment [21], personalized medicine [22], opinion diffusion [23] , trust in social networks [24], and graph summarization [25]. PSL represents the domain of interest as logical atoms. It uses first order logic rules to capture the dependency structure of the domain, based on which it builds a joint probabilistic model over all atoms. Instead of hard truth values of 0 (false) and 1 (true), PSL uses soft truth values relaxing the truth values to the interval $[0, 1]$. The logical connectives are adapted accordingly. This makes it easy to incorporate similarity or distance functions.

User defined *predicates* are used to encode the relationships and attributes and *rules* capture the dependencies and constraints. Each rule's antecedent is a conjunction of atoms and its consequent is a dis-junction. The rules can also labeled with non negative weights which are used during the inference process. The set of predicates and weighted rules thus make up a PSL program where known truth values of ground atoms are set from observed data and unknown truth values for the remaining atoms are learned using the PSL inference mechanism.

Given a set of atoms $\ell = \{\ell_1, \ldots, \ell_n\}$, an interpretation defined as

$I : \ell \rightarrow [0,1]^n$ is a mapping from atoms to soft truth values. PSL defines a probability distribution over all such interpretations such that those that satisfy more ground rules are more probable. *Lukasiewicz t-norm* and its corresponding co-norm are used for defining relaxations of the logical AND and OR respectively to determine the degree to which a ground rule is satisfied. Given an interpretation $I$, PSL defines the formulas for the relaxation of the logical conjunction ($\wedge$), dis-junction ($\vee$), and negation ($\neg$) as follows:

$$\begin{aligned} \ell_1 \tilde{\wedge} \ell_2 &= \max\{0, I(\ell_1) + I(\ell_2) - 1\}, \\ \ell_1 \tilde{\vee} \ell_2 &= \min\{I(\ell_1) + I(\ell_2), 1\}, \\ \tilde{\neg} l_1 &= 1 - I(\ell_1), \end{aligned}$$

where we use ˜to indicate the relaxation of the Boolean domain. The interpretation $I$ determines whether the rules is satisfied, if not, the *distance to satisfaction*. A rule $r \equiv r_{body} \rightarrow r_{head}$ is satisfied if and only if the truth value of head is atleast that of the body. The rule's distance to satisfaction measures the degree to which this condition is violated.

$$d_r(I) = \max\{0, I(r_{body} - I(r_{head}\}$$

PSL then induces a probability distribution over possible interpretations $I$ over the given set of ground atoms $l$ in the domain. If $R$ is the set of all ground rules that are instances of a rule from the system and uses only the atoms in $I$ then, the probability density function

$f$ over $I$ is defined as

$$f(I) = \frac{1}{Z}\text{exp}[-\sum_{r \in R} \lambda_r (d_r(I))^p] \tag{2.1}$$

$$Z = \int_I \text{exp}[-\sum_{r \in R} \lambda_r (d_r(I))^p] \tag{2.2}$$

where $\lambda_r$ is the weight of the rule $r$, $Z$ is the continuous version of the normalization constant used in discrete Markov random fields, and $p \in \{1, 2\}$ provides a choice between two different loss functions, linear and quadratic. The values of the atoms can be further restricted by providing linear equality and inequality constraints allowing one to encode functional constraints from the domain.

PSL provides for two kinds of inferences (a)most probable explanation and (b)calculation of the marginal distributions. In the MPE inference given a partial interpretation with grounded atoms based on observed evidence, the PSL program infers the truth values for the unobserved atoms satisfying the most likely interpretation. In the second setting, given ground truth data for all atoms we can learn the weights for the rules in our PSL program. In our work we leverage the MPE inference.

## 2.3 Dynamic Query Expansion using PSL

In [18], we used PSL to model user affiliations within groups. Specifically we built a PSL program for a social network where we have a set of users, their posts, messages to other users and the various groups we want to model. The rules we defined captured the dynamics of group affiliations through the various interactions. Through the MPE inference we classified users into different groups based on their hash-tag usage and their interactions with other users.

In this paper we extend our earlier work to achieve what we call dynamic query expansion through PSL. Similar to the query expansion methodology described earlier we start with an initial set of hash-tags which we believe are indicative of the affinity of a particular user to a candidate contesting in the election. We refer to these hash-tags as seed words. Instead of a single inference, we iteratively perform the inference over successive time windows such that the inference from window $w_t$ is used as a prior to window $w_{t+1}$ and the inference from that is used for window $w_{t+2}$ and so on.

Figure2.1 illustrates the design of the iterative algorithm for dynamic query expansion. The initial pre-processing starts with the tweet input stream which is filtered by the date range specified by the window size. For each election, tweets from a month leading up to the election were used. After extensive analysis it was determined that the most optimal window size was 3 days. Smaller window sizes resulted in sub-optimal inferences as there were not enough data points feeding into the PSL stage. Larger window sizes lead to memory issues as the PSL optimization has a time complexity of $O(n^{3.5})$ where $n$ is the number of relevant rule groundings, i.e., those with non-zero distance to satisfaction. The tweets passing the date filter are then geo-coded using a geo-location algorithm that infers the

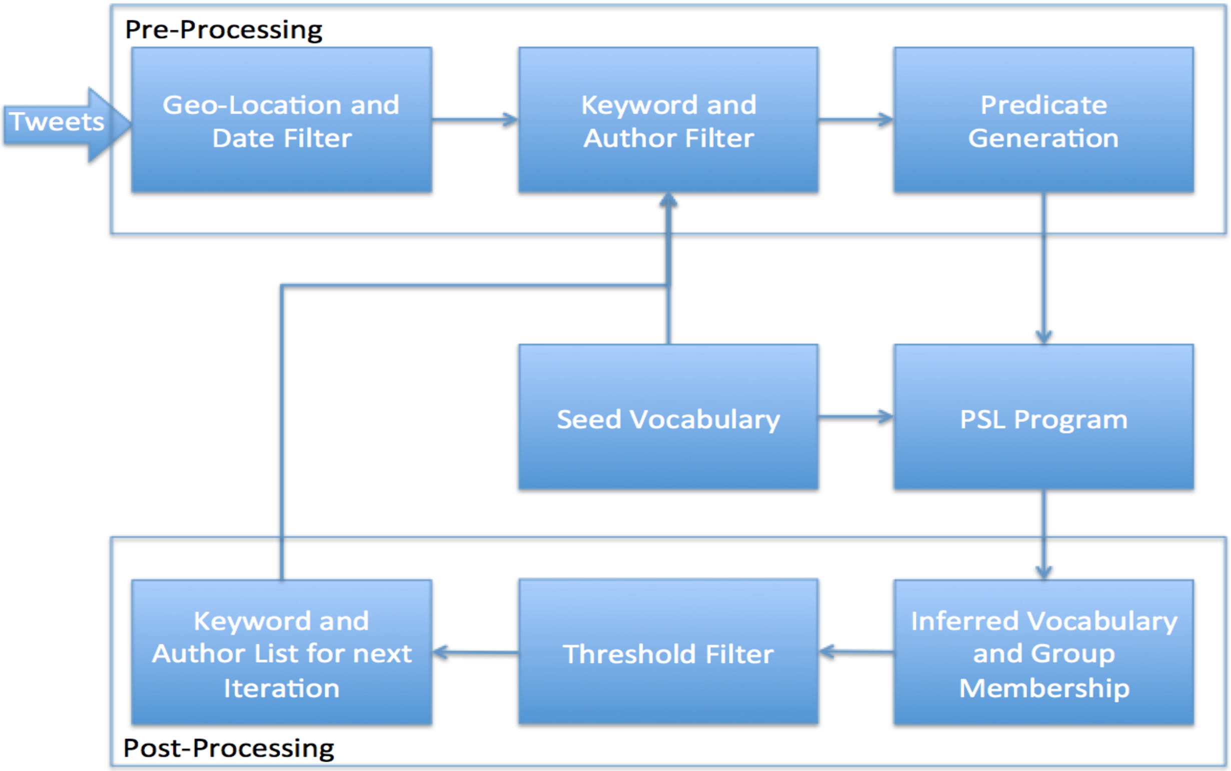

Pre-Processing
Tweets
Geo-Location and Date Filter
Keyword and Author Filter
Predicate Generation
Seed Vocabulary
PSL Program
Keyword and Author List for next Iteration
Threshold Filter
Inferred Vocabulary and Group Membership
Post-Processing

Figure 2.1: Design of the query expansion pipeline.

location of a tweet. This ensures that only tweets originating from the country of interest are used. The geo-location algorithm tags the tweets with a location by looking at the GPS coordinates of the tweet if available or landmarks and locations mentioned in the tweet or author's profile. For tweets that do not have either of these information it uses a label propagation algorithm to infer the author's location through his/her network.

The geo-tagged tweets are then tracked for the presence of a hashtag from the vocabulary for that particular iteration. In addition to filtering tweets using the vocabulary the authors whose affiliations are already inferred by the system are also used as a filtering criteria. The tweets are then converted into PSL predicates and fed into the inference process. The PSL program infers the hash-tags and tweeters that are mostly associated with a particular candidate. Each author and hash-tag's association with a candidate is measured using the truth value of the predicate grounding. In the post-processing step, these truth values are filtered by a threshold value to identify the hash-tags and authors strongly associated to a candidate. These hash-tags become a part of the vocabulary of the candidate and along with the users identified are used as a filter criterion for the next iteration. This iterative process proceeds until the day before the election when we obtain the final vocabulary which are strongly associated with a candidate.

Within the PSL program we define predicates to encode the network. The predicates $Tweeted(U,T)$ and $Contains(T,W)$ capture the fact that a user $U$ tweeted a tweet $T$ and tweet $T$ contains hash-tag $W$ respectively. Similarly, the belief that a user $U$ or hash-tag $W$ is affiliated/associated to the group $G$ is encoded as $Is\_Member(U,G)$ and $Belongs(W,G)$ respectively. In order to capture the temporal connectivity between the iterations, in addition to the initiating the inference process with the rule

$$Seed\_Word(W,G) \Rightarrow Belongs(W,G)$$

we define additional rules such as

$$Was\_Member(A,G) \Rightarrow Is\_Memeber(A,G)$$

$$Belonged(W,G) \Rightarrow Belongs(W,G)$$

where the predicates $Was\_Member$ and $Belonged$ are inferences from the previous time window and are loaded in as priors for the current iteration. These rules are weighted slightly lower than the recursive rules below so that the system overcomes the bias it had learned in light of new more convincing evidence. This way hash-tags that are more indicative of a user's affiliation are assigned stronger truth values or weights for every successive iteration and the weights hash-tags that aren't are reduced. The same reasoning applies to the user-candidate affiliations(memberships) too. Below we outline the recursive PSL rules that grows the hash-tag preferences and the user affiliations.

$$\begin{aligned} Tweeted(A,T)\,\tilde{\wedge}\,Contains(T,W)\,\tilde{\wedge}\,Belongs(W,G) \\ \tilde{\wedge}\,Positive(T) \Rightarrow Is\_Member(A,G) \end{aligned}$$

$$\begin{aligned} Tweeted(A,T)\,\tilde{\wedge}\,Contains(T,W)\,\tilde{\wedge}\,Belongs(W,G) \\ \tilde{\wedge}\,Negative(T) \Rightarrow\sim Is\_Member(A,G) \end{aligned}$$

$$\begin{aligned} Is\_Member(A,G)\,\tilde{\wedge}\,Tweeted(A,T)\,\tilde{\wedge}\,Contains(T,W) \\ \tilde{\wedge}\,Positive(T) \Rightarrow Belongs(W,G) \end{aligned}$$

$$\begin{aligned} Is\_Member(A,G)\,\tilde{\wedge}\,Tweeted(A,T)\,\tilde{\wedge}\,Contains(T,W) \\ \tilde{\wedge}\,Negative(T) \Rightarrow\sim Belongs(W,G) \end{aligned}$$

Here $Positive$ and $Negative$ are predicates whose truth values are calculated from the sentiment of the tweet such that the highly positive tweets get a truth value closer to 1.0 for the predicate $Positive$ and highly negative tweets are assigned a truth value of 1.0 for the predicate $Negative$. Since PSL works under the close world assumption, we do not need to specify the groundings that are false

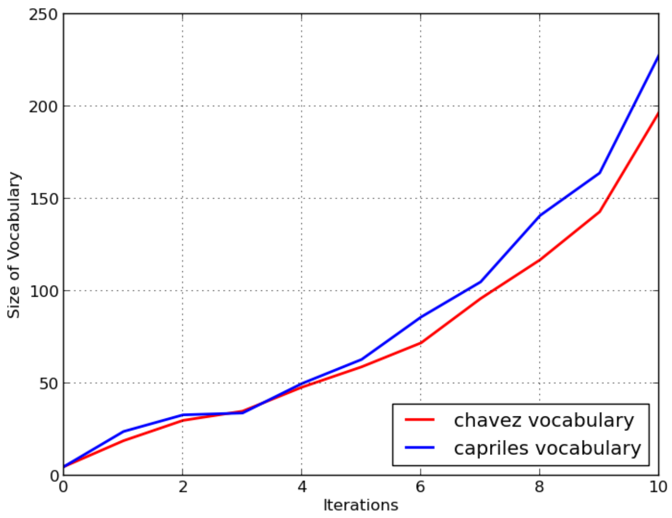


Figure 2.2: Vocabulary growth with each iteration.

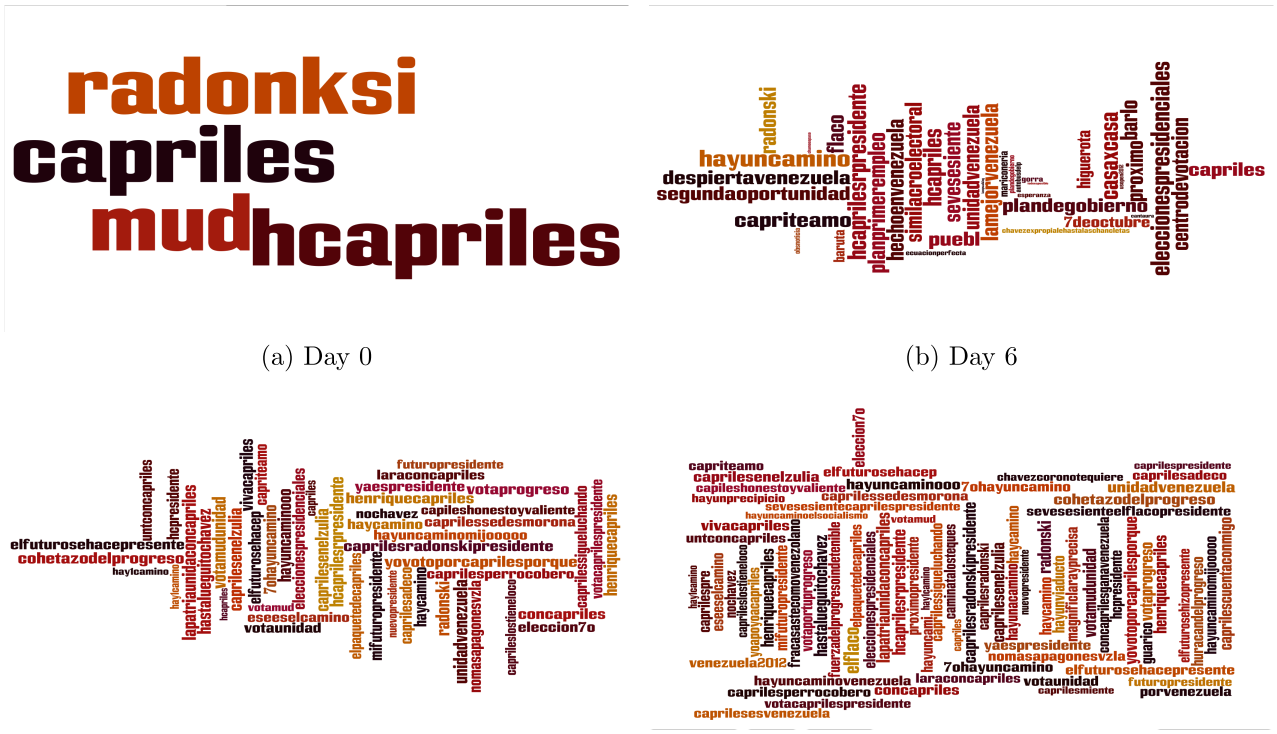
i.e., positive tweets are not assigned 0.0 for the predicate $Negative$ and vice-versa. For tweets that do not have a positive or negative orientation we assign a truth value of 0.5 for both the $Positive$ and $Negative$ predicates.

The last two rules defined below encode the assumption that when two hash-tags co-occur and one is a name of a candidate then the other hash-tag is bound to be about the candidate too.

$$Contains(T, W1) \tilde{\wedge} Contains(T, W2) \tilde{\wedge} Seed\_Word(W1, G) \\ \tilde{\wedge} Positive(T) \Rightarrow Belongs(W2, G)$$

$$Contains(T, W1) \tilde{\wedge} Contains(T, W2) \tilde{\wedge} Seed\_Word(W1, G) \\ \tilde{\wedge} Negative(T) \Rightarrow \sim Belongs(W2, G)$$

Once all the tweets are loaded into the PSL program as predicates, we start the inference process by closing all the predicates except $Is\_Member$ and $Belongs$. This way, only their truth values are inferred.



(a) Day 0

(b) Day 6

(c) Day 15

(d) Day 30

Figure 2.3: Evolution of hash-tags for Henrique Capriles

## 2.4 Results

Figure2.2 shows how the vocabulary grows with each iteration for the two candidates who contested the Venezuelan presidential election on October 7th 2012.

Figure2.3 shows how the hash-tags for Henrique Capriles evolved during the month leading up to the election. Initially in Figure2.3a the system starts with only a few hand picked hash-tags that constitute the seed vocabulary. After a few iterations Figure2.3b shows how the vocabulary has grown. However, not all the words identified until now remain in the final vocabulary as the system drops certain words in successive iterations. At the same time it is also

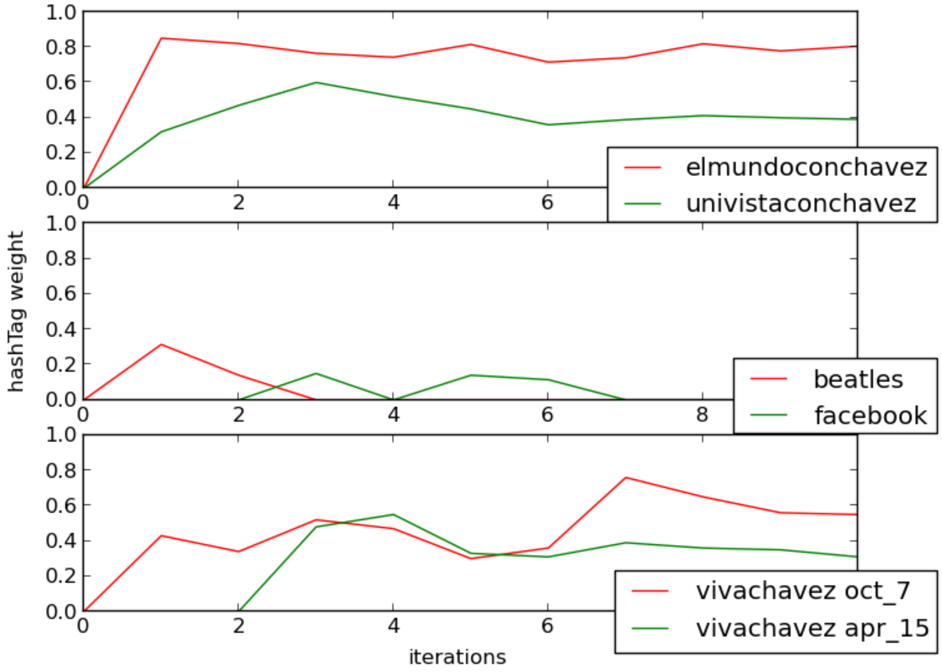


Figure 2.4: Time series comparison for different hash-tags identified for Hugo Chavez.

noticed that hash-tags like ”capriles” and ”hayuncamino” which are very strongly associated with Capriles consistently remain as the top ranked hash-tags even after ten iterations(Figure2.3d). It is also interesting to note that the algorithm identified hash-tags like ”nochavez”(Figure2.3c) and attributed it rightly to Hugo Chavez's primary contender- Capriles.

In figure 2.4, the first plot elucidates how hash-tags like *”elmnduconchavez”* and *”univistaconchavez”* remain highly associated with Hugo Chavez for the October 7th Presidential election. These hash-tags remain indicative of a user's affiliation throughout the month leading up to the election. Meanwhile hash-tags such as *”beatles”* and *”facebook”* ( in second plot) show spikes in their time series primarily because users affiliated with Chavez used them during that time window. But as iterative process continues the system drops these non-informative words. The third plot presents another interesting observation The hash-tag *"vivachavez"* is part of both the Venezuelan elections despite the fact that Hugo Chavez did not contest the second election on April 15th 2013. It is picked up as a phrase commonly used by supporters of Nicholas Maduro whose election campaign was strategized around the death of Hugo Chavez to garner sympathy and mobilize support. Similarly variations of the hash-tags *"hayuncamino"* and *"unidadvenuzela"* was returned for Henrique Capriles for both these elections.

# Chapter 3

# Prediction Models

In this section we review two prediction models we adapted from current literature to test our hypothesis. The first one is a naive model that forecasts elections based on the counts of mentions of a candidate. We dub this as *"unique visitor model"* and is adapted from [7] and [5]. The second model uses a regression fit to regress from tweet features to opinion polls and then predicts election. This we dub as the *"regression model"* and is adapted from [8] and [6].

## 3.1 Unique Visitor Model

Without any loss of generality, it can be assumed that large parties that are more popular will have a larger social media foot print than smaller and less popular parties. This model takes advantage of this assumption and predicts elections by calculating the relative popularity of candidates contesting the election. We first define a vocabulary for each candidate. This vocabulary is crafted by hand and includes the candidate's names and aliases, the name and acronyms for his/her political party and the official Twitter handle of the candidate and is same as the seed vocabulary that was used to initialize the PSL pipeline. For the given time period, the tweets from the country in question are tracked for the occurrence of the terms in the vocabulary. We then build a time series of sentiment and Klout scores from the tweets returned. Klout score is a value provided by Klout.com that quantifies the impact each user has on social media. We use the sentiment scores provided as a part of the meta-data of the tweet. Once a time series of the Klout and sentiment scores are built, we calculate the absolute popularity of a candidate $C_d$ as:

$$C_d = \sum_i K_i * UCS_{id} \tag{3.1}$$

where $K_i$ is the Klout score for user $i$, and $UCS_{id}$ is User Candidate Score, the average of sentiment scores for all tweets from user $i$ about candidate $d$. We then normalize the popularity scores across all candidates so that they sum to 1. This gives us the relative popularity of each candidate $P_d$ using which we predict the elections.

$$P_d = \sum_i \frac{C_d}{C_i} \tag{3.2}$$

From the above equations, it is noticeable that each user contributes only once to the popularity score of a candidate. This was preferred to merely counting the mentions of a candidate since we wanted to remove the bias of bot generated tweets from election campaigns that boosted the number of times a candidate is mentioned on Twitter.

## 3.2 Regression Model

In this model, in addition to Twitter data, we leverage the opinion polls available for the elections to make our predictions. Like the earlier model we track the tweets that contain a word from the vocabulary defined for each candidate. We then define a linear regression fit that uses the opinion polls as dependent variable and features generated from these tweets as independent variable. We use a total of 6 features based on Klout scores, number of unique users, total number of mentions, sentiment and incumbency. We normalize each of these features across all candidates to get the relative share of the volume. For example for the we define share of positive mentions($SoPM$) as:

$$SoPM(x) = \frac{\#PositiveMentions(x)}{\sum_i \#PositiveMentions(i)} \tag{3.3}$$

and share of negative users($SoNU$) as:

$$SoNU(x) = \frac{\sum_j K_j}{\sum_i \sum_j K_j} \tag{3.4}$$

where $K_j$ is the Klout score of user $j$ who tweeted negatively about a candidate. Similarly we define share of sentiment ($SoS$) as the sum of all sentiment scores normalized across all candidates. We use a binary variable for incumbency. We then build a time-line of opinion polls. For each of the polling dates we calculate these features by using tweets created during the 10 day window leading up to the polling date. When we have more than one polling house publishing its opinion poll for the same date we take the average of the polls. Once we create a feature set for all the polling dates, we fit a simple

| Feature | Coefficient Value |
|---|---:|
| $SoPU$ | 0.4622 |
| $SoNU$ | -0.443 |
| $SoPM$ | 0.1158 |
| $SoNM$ | -0.065 |
| $SoS$ | 0.156 |
| $Incumbency$ | 0.0 |

Table 3.1: Regression coefficients learned for features

least square regression as :

$$\begin{aligned} Popularity(x) = \alpha_1 * SoPM(x) + \alpha_2 * SoNM(x) \\ +\beta_1 * SoPU(x) + \beta_2 * SoNU(x) \\ +\gamma * SoS(x) + \delta * Incumbency(x) + \epsilon \end{aligned} \tag{3.5}$$

Table3.1 details the coefficients learned for each feature averaged over all the candidates from all the elections. The values confirm our hypothesis that the number of unique users and sentiment have more predictive power than total number of mentions. Intuitively it is also seen that the coefficients for share of negative users and negative mentions carry a negative weight. Another interesting observation is the fact that the incumbency binary variable is not very predictive which is contradictory to the popular opinion. After learning the regression fit, we make a prediction by building such features using the same 10 day window leading up to the prediction date.

## 3.3 Performance

The Unique Visitor Model and the Regression Model were tested exhaustively on a total of 36 elections from Latin America during 2012 and 2013 ranging from local mayor elections to presidential

| Election Type | Number of Elections | Number of Correct Predictions | Accuracy |
|---|---:|---:|---:|
| President/Prime Minister | 8 | 8 | 100% |
| Governor | 4 | 3 | 75% |
| Mayor | 24 | 12 | 50% |
| Overall | 36 | 23 | 63.88% |

Table 3.2: Track Record of Prediction Algorithms

elections at the country level. It is important to note that every single election was predicted ahead of time and not in retrospect. The tweets were purchased from DataSift, an infoveilence service that resells Twitter data. On an average we collected close to 2 million unique tweets a day from over 21 countries in Latin and South America. Then these tweets were geo-coded using a geo-location algorithm we developed to obtain tweets from the country of interest. Only tweets from the locations pertaining to elections were used to make the predictions. For example, for the Rio De Janeiro Mayor elections only tweets from the city of Rio De Janeiro were used and similarly for state level Governor elections only tweet originating from that particular state was used. Once the tweets were filtered by location the time series of Klout and sentiment scores were calculated by tracking the tweets for the mentions of candidate.

Table3.2 below shows the over all performance of the two models. It can be noticed that the accuracy drops as the granularity of the elections reduces. This is primarily due to the fact that, opinion polls were available only for the country level elections. Therefore, we could not use the Regression Model for the state or city level elections. This increased the error as the predictions were generated only from the naive Unique Visitor Model.

Also, from the tweets collected it was noticed that there wasn't much chatter on Twitter about these smaller local elections. This skewed the results as the model tracking the names of the candidates was using tweets that mentioned the candidate's name but wasn't about the candidate contesting in the election but was about some other person having the same name as the candidate. If the city level elections were ignored as outliers the over all accuracy of the models improves to 91.6%.

# Chapter 4

# Evaluations and Results

To evaluate our hypothesis about the vocabularies we test our models on 8 different presidential elections from Latin America using both the seed vocabulary and the vocabulary generated by the query expansion exercise. We use the results obtained using the seed vocabulary detailed in the previous section as a baseline score. We then use the same vocabulary to seed our PSL learning algorithm. The prediction algorithms are then run again, now by using the expanded vocabulary obtained through the query expansion. In order to remain consistent with predicting ahead of time, we use track only the hash-tags identified by the query expansion pipeline until that particular date. Figure 4.1 shows the increase in the number of documents that were used by the algorithms to make a predictions. It is noticed when averaged across all the 8 elections we notice close to a 2X increase in the number of tweets that were used by these models. This is a substantial increase in the recall[*Aravind: placeHolder*] of relevant tweets for the domain.

To further illustrate the fact that the vocabulary used by such algorithms plays a vital role, we compare the performance of the models

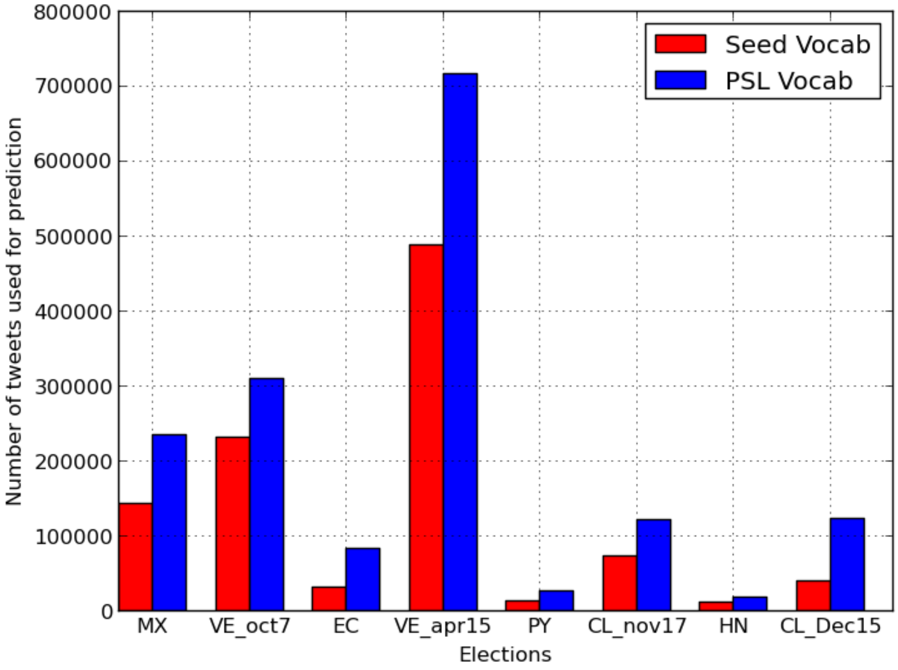


Figure 4.1: Recall of seed vocabulary vs PSL vocabulary

using the two different vocabularies. The Mean Absolute Percentage Error (MAPE) was used as a metric to measure the performance of the models. To reduce the effect of outliers we track the popularity of only the major candidates and ignore the ones who obtained less than 10% of the total votes. Table 4.1 shows the performance of each vocabulary in different elections. On an average it is seen that the mean absolute percentage error is reduced by 15.58%. We see greater and more consistent improvement with the regression model as the model weighs each window of tweets differently depending upon the opinion poll time series whereas the unique visitor model values them equally. Therefore, when the algorithm uses the 'not-so-informative' hash-tags identified during the earlier iterations, the sentiment value and the counts of these mentions bring down the

| Election | UniVis+Seed | UniVis+PSL | Improvement | Reg+Seed | Reg+PSL | Improvement |
|---:|---:|---:|---:|---:|---:|---:|
| Mexico | 0.353 | 0.368 | -4.2% | 0.123 | 0.07 | 43.09% |
| Venezuela_Oct7 | 0.069 | 0.077 | -11.59% | 0.158 | 0.109 | 31.01& |
| Ecuador | 0.531 | 0.547 | -3.01% | 0.263 | 0.244 | 7.22% |
| Venezuela_Apr15 | 0.198 | 0.178 | 10.10% | 0.142 | 0.112 | 21.126& |
| Paraguay | 0.34 | 0.288 | 15.29% | 0.2 | 0.18 | 10% |
| Chile_Nov17 | 0.56 | 0.42 | 25% | 0.245 | 0.207 | 15.51% |
| Honduras | 0.563 | 0.527 | 6.39% | 0.293 | 0.184 | 37.20% |
| Chile_Dec17 | 0.096 | 0.061 | 36.45% | 0.409 | 0.369 | 9.77% |

Table 4.1: Improvement in prediction accuracy

accuracy of the model even though at a later stage hash-tags that are strongly indicative of a user's preference is picked up. So words such as "facebook" which occur commonly dominate the counts and therefore skew the results even though they are dropped from the vocabulary at a later point.

# Chapter 5

# Conclusions and Future Work

[*Aravind: I am having trouble structuring the this section and also with the language to be used to detail the future work. Can you help me out here?*] In this work we built two prediction algorithms to forecast elections and showed how Twitter in addition to standard opinion polls can be used to pulse the political opinion of a country. We also established the reproducibility of election predictions using Twitter by forecasting more than thirty elections from Latin America. We then implemented a novel query expansion methodology using Probabilistic Soft Logic. We showed how vocabulary has a direct impact on the recall of documents and the accuracy of prediction algorithms. Using the query expansion methodology we were able to improve the accuracy of the prediction algorithms by upto 16%. It is also important to note that though we used elections to show performance gains, the query expansion system is generic and can be used to learn a vocabulary for any given domain.

Further, this work was motivated towards a future goal to model the electorate demographics. With more fine grained data about the gender, age and exact location of a user it is possible to infer the preferences at a group level rather than at a user level. This would enable us to study the various interactions between groups and individual users in more detail and thus make more informed election predictions.

While modeling the interactions between users using PSL we used hard coded rule weights. With a labeled data set providing ground truths about user affiliations it is possible to study the various forms of interactions by using the weight learning mechanism of PSL to understand which theories about social group modeling are more probable and which aren't.

Also, while tracking the candidates using the seed vocabulary we noticed that using the names of candidates introduced a lot of noise as some of the names such as 'Jose' are very common. To avoid this we propose to use more sophisticated named entity recognition to improve the accuracy of the tweets returned about a particular candidate.

In addition to Twitter, we also propose to use other data sources such as Google Search Trends, Facebook and web blogs to track the popularity of candidates online.

# Bibliography

[1] Daniel Gruhl, Ramanathan Guha, Ravi Kumar *et al.*, “The predictive power of online chatter,” in *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining.* ACM, 2005, pp. 78–87.

[2] Sitaram Asur and Bernardo A Huberman, “Predicting the future with social media,” in *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on*, vol. 1. IEEE, 2010, pp. 492–499.

[3] Johan Bollen, Huina Mao, and Xiaojun Zeng, “Twitter mood predicts the stock market,” *Journal of Computational Science*, vol. 2, no. 1, pp. 1–8, 2011.

[4] Johan Bollen, Huina Mao, and Alberto Pepe, “Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena.” in *ICWSM*, 2011.

[5] Andranik Tumasjan, Timm Oliver Sprenger, Philipp G Sandner *et al.*, “Predicting elections with twitter: What 140 characters reveal about political sentiment.” *ICWSM*, vol. 10, pp. 178–185, 2010.

[6] Brendan O'Connor, Ramnath Balasubramanyan, Bryan R Routledge *et al.*, "From tweets to polls: Linking text sentiment to public opinion time series." *ICWSM*, vol. 11, pp. 122–129, 2010.

[7] Diego Saez-Trumper, Wagner Meira, and Virgilio Almeida, "From total hits to unique visitors model for elections forecasting," in *International Conference on Web Science*, 2011.

[8] Adam Bermingham and Alan F Smeaton, "On using twitter to monitor political sentiment and predict election results," 2011.

[9] Gianluca Demartini, Stefan Siersdorfer, Sergiu Chelaru *et al.*, "Analyzing political trends in the blogosphere." in *ICWSM*, 2011.

[10] Avishay Livne, Matthew P Simmons, Eytan Adar *et al.*, "The party is over here: Structure and content in the 2010 election." in *ICWSM*, 2011.

[11] Michael D Conover, Bruno Gonçalves, Jacob Ratkiewicz *et al.*, "Predicting the political alignment of twitter users," in *Privacy, security, risk and trust (passat), 2011 ieee third international conference on and 2011 ieee third international conference on social computing (socialcom).* IEEE, 2011, pp. 192–199.

[12] Ernesto Diaz-Aviles, Claudia Orellana-Rodriguez, and Wolfgang Nejdl, "Taking the pulse of political emotions in latin america based on social web streams," in *Web Congress (LA-WEB), 2012 Eighth Latin American.* IEEE, 2012, pp. 40–47.

[13] Eni Mustafaraj, Samantha Finn, Carolyn Whitlock *et al.*, “Vocal minority versus silent majority: Discovering the opionions of the long tail,” in *Privacy, security, risk and trust (passat), 2011 ieee third international conference on and 2011 ieee third international conference on social computing (socialcom).* IEEE, 2011, pp. 103–110.

[14] Panagiotis Takis Metaxas, Eni Mustafaraj, and Daniel Gayo-Avello, “How (not) to predict elections,” in *Privacy, security, risk and trust (PASSAT), 2011 IEEE third international conference on and 2011 IEEE third international conference on social computing (SocialCom).* IEEE, 2011, pp. 165–171.

[15] Daniel Gayo-Avello, “” i wanted to predict elections with twitter and all i got was this lousy paper”–a balanced survey on election prediction using twitter data,” *arXiv preprint arXiv:1204.6441*, 2012.

[16] ——, “Don’t turn social media into another’literary digest’poll,” *Communications of the ACM*, vol. 54, no. 10, pp. 121–128, 2011.

[17] Daniel Gayo-Avello, Panagiotis Takis Metaxas, and Eni Mustafaraj, “Limits of electoral predictions using twitter.” in *ICWSM*, 2011.

[18] Bert Huang, Stephen H Bach, Eric Norris *et al.*, “Social group modeling with probabilistic soft logic,” in *NIPS Workshop on Social Network and Social Media Analysis: Methods, Models, and Applications*, 2012.

[19] Angelika Kimmig, Stephen Bach, Matthias Broecheler *et al.*, "A short introduction to probabilistic soft logic," in *Proceedings of the NIPS Workshop on Probabilistic Programming: Foundations and Applications*, 2012, pp. 1–4.

[20] Matthias Broecheler and Lise Getoor, "Computing marginal distributions over continuous markov networks for statistical relational learning," in *Advances in Neural Information Processing Systems*, 2010, pp. 316–324.

[21] Matthias Brocheler, Lilyana Mihalkova, and Lise Getoor, "Probabilistic similarity logic," *arXiv preprint arXiv:1203.3469*, 2012.

[22] Stephen H Bach, Matthias Broecheler, Stanley Kok *et al.*, "Decision-driven models with probabilistic soft logic," 2010.

[23] Stephen Bach, Matthias Broecheler, Lise Getoor *et al.*, "Scaling mpe inference for constrained continuous markov random fields with consensus optimization," in *Advances in Neural Information Processing Systems*, 2012, pp. 2663–2671.

[24] Bert Huang, Angelika Kimmig, Lise Getoor *et al.*, "Probabilistic soft logic for trust analysis in social networks," in *International Workshop on Statistical Relational AI*, 2012.

[25] Alex Memory, Angelika Kimmig, Stephen Bach *et al.*, "Graph summarization in annotated data using probabilistic soft logic," *status: accepted*, 2012.