Learning Multiview Embeddings of Twitter Users

Adrian Benton[*]
adrian@cs.jhu.edu

Raman Arora[*]
arora@cs.jhu.edu

Mark Dredze[*†]
mdredze@cs.jhu.edu

[*]Human Language Technology Center of Excellence
Center for Language and Speech Processing, Johns Hopkins University
Baltimore, MD 21218 USA
[†]Bloomberg LP, New York, NY 10022

Abstract

Low-dimensional vector representations are widely used as stand-ins for the text of words, sentences, and entire documents. These embeddings are used to identify similar words or make predictions about documents. In this work, we consider embeddings for social media users and demonstrate that these can be used to identify users who behave similarly or to predict attributes of users. In order to capture information from all aspects of a user's online life, we take a multiview approach, applying a weighted variant of Generalized Canonical Correlation Analysis (GCCA) to a collection of over 100,000 Twitter users. We demonstrate the utility of these multiview embeddings on three downstream tasks: user engagement, friend selection, and demographic attribute prediction.

1 Introduction

Dense, low-dimensional vector representations (embeddings) have a long history in NLP, and recent work on neural models have provided new and popular algorithms for training representations for word types (Mikolov et al., 2013; Faruqui and Dyer, 2014), sentences (Kiros et al., 2015), and entire documents (Le and Mikolov, 2014). These embeddings often have nice properties, such as capturing some aspects of syntax or semantics and outperforming their sparse counterparts at downstream tasks.

While there are many approaches to generating embeddings of *text*, it is not clear how to learn embeddings for social media *users*. There are several different types of data (views) we can use to build user representations: the text of messages they post, neighbors in their local network, articles they link to, images they upload, etc. We propose unsupervised learning of representations of users with a variant of Generalized Canonical Correlation Analysis (*GCCA*) (Carroll, 1968; Van De Velden and Bijmolt, 2006; Arora and Livescu, 2014; Rastogi et al., 2015), a multiview technique that learns a single, low-dimensional vector for each user best capturing information from each of their views. We believe this

is more appropriate for learning user embeddings than concatenating views into a single vector, since views may correspond to different modalities (image vs. text data) or have very different distributional properties. Treating all features as equal in this concatenated vector is not appropriate.

We offer two main contributions: (1) an application of *GCCA* to learning vector representations of social media users that best accounts for all aspects of a user's online life, and (2) an evaluation of these vector representations for a set of Twitter users at three different tasks: user engagement, friend, and demographic attribute prediction.

2 Twitter User Data

We begin with a description of our dataset, which is necessary for understanding the data available to our multiview model. We uniformly sampled 200,000 users from a stream of publicly available tweets from the 1% Twitter stream from April 2015. To include typical, English speaking users we removed users with verified accounts, more than 10,000 followers, or non-English profiles[1]. For each user we collected their 1,000 most recent tweets, and then filtered out non-English tweets. Users without English tweets in January or February 2015 were omitted, yielding a total of 102,328 users. Although limiting tweets to only these two months restricted the number of tweets we were able to work with, it also ensured that our data are drawn from a narrow time window, controlling for differences in user activity over time. This allows us to learn distinctions between users, and not temporal distinctions of content. We will use this set of users to learn representations for the remainder of this paper.

Next, we expand the information available about these users by collecting information about their social networks. Specifically, for each user mentioned in a tweet by one of the 102,328 users, we collect up to the 200 most recent English tweets for these users from January and February 2015. Similarly, we collected the 5,000 most recently added friends and followers of each of the 102,328 users. We then sampled 10 friends and 10 followers for each user and collected

---

[1]Identified with langid (Lui and Baldwin, 2012).

up to the 200 most recent English tweets for these users from January and February 2015. Limits on the number of users and tweets per user were imposed so that we could operate within Twitter's API limits. This data supports several of our prediction tasks, as well as the four sources for each user: their tweets, tweets of mentioned users, friends and followers.

3 User Views

Our user dataset provides several sources of information on which we can build user views: text posted by the user (*ego*) and people that are mentioned, friended or followed by the user and their posted text.

For each text source we can aggregate the many tweets into a single document, e.g. all tweets written by accounts mentioned by a user. We represent this document as a bag-of-words (*BOW*) in a vector space model with a vocabulary of the 20,000 most frequent word types after stopword removal. We will consider both count and TF-IDF weighted vectors.

A common problem with these high dimensional representations is that they suffer from the curse of dimensionality. A natural solution is to apply a dimensionality reduction technique to find a compact representation that captures as much information as possible from the original input. Here, we consider principal components analysis (PCA), a ubiquitous linear dimensionality reduction technique, as well as word2vec (Mikolov et al., 2013), a technique to learn nonlinear word representations.

We consider the following views for each user.

BOW: We take the bag-of-words (both count and TF-IDF weighted) representation of all tweets made by users in that view (ego, mention, friend, or follower) following the above pre-processing.

BOW-PCA: We run PCA and extract the top principal components for each of the above views. We also consider *all* possible combinations of views obtained by concatenating views before applying PCA, and concatenating PCA-projected views. By considering all possible concatenation of views, we ensure that this method has access to the same information as multi-view methods. Both the raw *BOW* and *BOW-PCA* representations have been explored in previous work for demographic prediction (Volkova et al., 2014; Al Zamal et al., 2012) and recommendation systems (Abel et al., 2011; Zangerle et al., 2013).

Word2Vec: *BOW-PCA* is limited to linear representations of *BOW* features. Modern neural network based approaches to learning word embeddings, including word2vec continuous bag of words and skipgram models, can learn nonlinear representations that also capture local context around each word (Mikolov et al., 2013). We represent each view as the simple average of the word embeddings for all tokens within that view (e.g., all words written by the ego user). Word embeddings are learned on a sample of 87,755,398 tweets and profiles uniformly sampled from the 1% Twitter stream in April 2015 along with all the tweets/profiles collected for our set of users – a total of over a billion tokens. We use the word2vec tool, select either skipgram or continuous bag-of-words embeddings on dev data for each prediction task, and train for 50 epochs. We use the default settings for all other parameters.

NetSim: An alternative to text based representations is to use the social network of users as a representation. We encode a user's social network as a vector by treating the users as a vocabulary, where users with similar social networks have similar vector representations (*NetSim*). An n-dimensional vector then encodes the user's social network as a bag-of-words over this user vocabulary. In other words, a user is represented by a summation of the one-hot encodings of each neighboring user in their social network. In this representation, the number of friends two users have in common is equal to the dot product between their social network vectors. We define the social network may be as one's followers, friends, or the union of both. The motivation behind this representation is that users who have similar networks may behave in similar ways. Such network features are commonly used to construct user representations as well as to make user recommendations (Lu et al., 2012; Kywe et al., 2012).

NetSim-PCA: The PCA-projected representations for each *NetSim* vector. This may be important for computing similarity, since users are now represented as dense vectors capturing linear correlations in the friends/followers a user has. *NetSim-PCA* is to *NetSim* as *BOW-PCA* is to *BOW*– we apply PCA directly to the user's social network as opposed to the *BOW* representations of users in that network.

Each of these views can be treated independently as a user representation. However, different downstream tasks may favor different views. For example, the friend network is useful at recommending new friends, whereas the ego tweet view may be better at predicting what content a user will post in the future. Picking a single view may ignore valuable information as views may contain complementary information, so using multiple views improves on a single view. One approach is to concatenate multiple views together, but this further increases the size of the user embeddings. In the next section, we propose an alternate approach for learning a single embedding from multiple views.

4 Learning Multiview User Embeddings

We use Generalized Canonical Correlation Analysis (*GCCA*) (Carroll, 1968) to learn a single embedding from multiple views. *GCCA* finds G, U_i that minimize:

$$\arg\min_{G,U_i} \sum_i \|G - X_i U_i\|_F^2 \quad \text{s.t. } G'G = I \quad (1)$$

where $X_i \in \mathbb{R}^{n \times d_i}$ corresponds to the data matrix for the ith view, $U_i \in \mathbb{R}^{d_i \times k}$ maps from the latent space to observed view i, and $G \in \mathbb{R}^{n \times k}$ contains all user representations (Van De Velden and Bijmolt, 2006).

Since each view may be more or less helpful for a downstream task, we do not want to treat each view equally in learning a single embedding. Instead, we weigh each view differently in the objective:

$$\arg\min_{G,U_i}\sum_i w_i\|G - X_iU_i\|_F^2 \text{ s.t. } G'G = I, w_i \geq 0 \quad (2)$$

where w_i explicitly expresses the importance of the ith view in determining the joint embedding. The columns of G are the eigenvectors of $\sum_i w_iX_i(X_i'X_i)^{-1}X_i'$, and the solution for $U_i = (X_i'X_i)^{-1}X_i'G$. In our experiments, we use the approach of Rastogi et al. (2015) to learn G and U_i, since it is more memory-efficient than decomposing the sum of projection matrices.

GCCA embeddings were learned over combinations of the views in §3. When available, we also consider *GCCA-net*, where in addition to the four text views, we also include the follower and friend network views used by *NetSim-PCA*. For computational efficiency, each of these views was first reduced in dimensionality by projecting its *BOW* TF-IDF-weighted representation to a 1000-dimensional vector through PCA.[2] We add an identity matrix scaled by a small amount of regularization, 10^{-8}, to the per-view covariance matrices before inverting, for numerical stability, and use the formulation of *GCCA* reported in Van De Velden and Bijmolt (2006), which ignores rows with missing data (some users had no data in the mention tweet view and some users accounts were private). We tune the weighting of each view i, $w_i \in \{0.0, 0.25, 1.0\}$, discriminatively for each task, although the *GCCA* objective is unsupervised once the w_i are fixed.

We also consider a minor modification of *GCCA*, where G is scaled by the square-root of the singular values of $\sum_i w_iX_iX_i'$, *GCCA-sv*. This is inspired by previous work showing that scaling each feature of multiview embeddings by the singular values of the data matrix can improve performance at downstream tasks such as image or caption retrieval (Mroueh et al., 2015). Note that if we only consider a single view, X_1, with weight $w_1 = 1$, then the solution to *GCCA-sv* is identical to the PCA solution for data matrix X_1, without mean-centering.

When we compare representations in the following tasks, we sweep over embedding width in $\{10, 20, 50, 100, 200, 300, 400, 500, 1000\}$ for all methods. This applies to *GCCA*, *BOW-PCA*, *NetSim-PCA*, and *Word2Vec*. We also consider concatenations of vectors for every possible subset of views: singletons, pairs, triples, and all views. We tried applying PCA directly to the concatenation of all 1000-dimensional *BOW-PCA* views, but this did not perform competitively in our experiments.

---

[2] We excluded count vectors from the *GCCA* experiments for computational efficiency since they performed similarly to TF-IDF representations in initial experiments.

5 Experimental Setup

We selected three user prediction tasks to demonstrate the effectiveness of the multi-view embeddings: user engagement prediction, friend recommendation and demographic characteristics inference. Our focus is to show the performance of multiview embeddings compared to other representations, not on building the best system for a given task.

User Engagement Prediction The goal of user engagement prediction is to determine which topics a user will likely tweet about, using hashtag as a proxy. This task is similar to hashtag recommendation for a tweet based on its contents (Kywe et al., 2012; She and Chen, 2014; Zangerle et al., 2013). Purohit et al. (2011) presented a supervised task to predict if a hashtag would appear in a tweet using features from the user's network, previous tweets, and the tweet's content.

We selected the 400 most frequently used hashtags in messages authored by our users and which first appeared in March 2015, randomly and evenly dividing them into dev and test sets. We held out the first 10 users who tweeted each hashtag as exemplars of users that would use the hashtag in the future. We ranked all other users by the cosine distance of their embedding to the average embedding of these 10 users. Since embeddings are learned on data pre-March 2015, the hashtags cannot impact the learned representations. Performance is measured using precision and recall at k, as well as mean reciprocal rank (MRR), where a user is marked as correct if they used the hashtag. Note that this task is different than that reported in Purohit et al. (2011), since we are making recommendations at the level of users, not tweets.

Friend Recommendation The goal of friend recommendation/link prediction is to recommend/predict other accounts for a user to follow (Liben-Nowell and Kleinberg, 2007).

We selected the 500 most popular accounts – which we call celebrities – followed by our users, randomly, and evenly divided them into dev and test sets. We randomly select 10 users who follow each celebrity and rank all other users by cosine distance to the average of these 10 representations. The tweets of selected celebrities are removed during embedding training so as not to influence the learned representations. We use the same evaluation as user engagement prediction, where a user is marked as correct if they follow the given celebrity.

For both user engagement prediction and friend recommendation we z-score normalize each feature, subtracting off the mean and scaling each feature independently to have unit variance, before computing cosine similarity. We select the approach and whether to z-score normalize based on the development set performance.

Demographic Characteristics Inference Our final task is to infer the demographic characteristics of a user (Al Zamal et al., 2012; Chen et al., 2015).

Model	Dim	P@1000	R@1000	MRR
BOW	20000	0.009/0.005	0.241/0.157	0.006/0.006
BOW-PCA	500	0.011/0.008	0.312/0.29	0.007/0.009
NetSim	NA	0.006/0.006	0.159/0.201	0.004/0.004
NetSim-PCA	300	0.010/0.008	0.293/0.299	0.006/0.006
Word2Vec	100	0.009/0.007	0.254/0.217	0.005/0.004
GCCA	100	0.012/0.009	0.357/0.325	0.011/0.008
GCCA-sv	500	**0.012/0.010**	0.359/0.334	**0.010/0.011**
GCCA-net	200	0.013/0.009	**0.360/0.346**	**0.011/0.011**
NetSize	NA	0.001/0.001	0.012/0.012	0.000/0.000
Random	NA	0.000/0.000	0.002/0.008	0.000/0.000

Table 1: Macro performance at user engagement prediction on dev/test. Ranking of model performance was consistent across metrics. Precision is low since few users tweet a given hashtag. Values bolded by best test performance per metric. Baselines (bottom): *NetSize*: a ranking of users by the size of their local network; *Random* randomly ranks users.
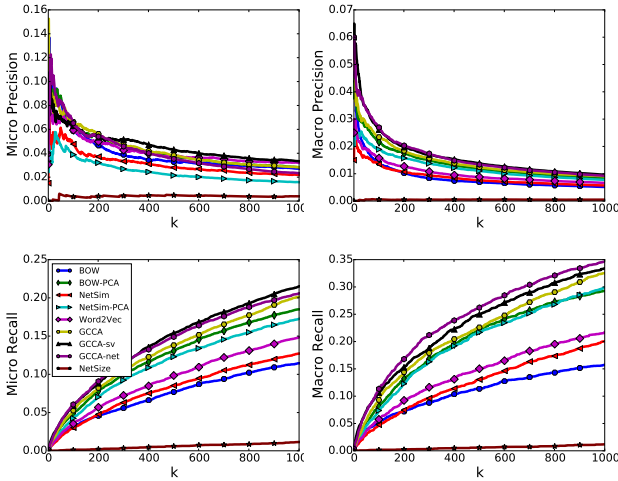

Figure 1: The best-performing approaches on user engagement prediction as a function of k (number of recommendations). The ordering of methods is consistent across k.

We use the dataset from Volkova et al. (2014; Volkova (2015) which annotates 383 users for age (old/young), 383 for gender (male/female), and 396 political affiliation (republican/democrat), with balanced classes. Predicting each characteristic is a binary supervised prediction task. Each set is partitioned into 10 folds, with two folds held out for test, and the other eight for tuning via cross-fold validation. The provided dataset contained tweets from each user, mentioned users, friends and follower networks. It did not contain the actual social networks for these users, so we did not evaluate *NetSim*, *NetSim-PCA*, or *GCCA-net* at these prediction tasks.

Each feature was z-score normalized before being passed to a linear-kernel SVM where we swept over $10^{-4}, \ldots, 10^4$ for the penalty on the error term, C.

6 Results

User Engagement Prediction Table 1 shows results for user engagement prediction and Figure 1 the precision and recall curves as a function of number of recommendations. *GCCA* outperforms other methods for precision and recall at 1000, and does close to the best in terms of MRR. Including network views (*GCCA-*
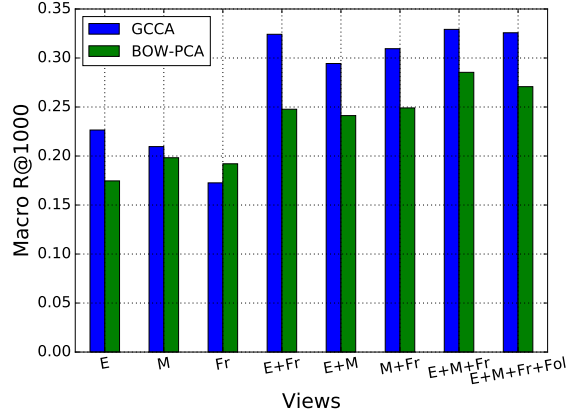

Figure 2: Macro recall@1000 on user engagement prediction for different combinations of text views. Each bar shows the best-performing model swept over dimensionality. *E*: ego, *M*: mention, *Fr*: friend, *Fol*: followertweet views.

Model	Dim	P@1000	R@1000	MRR
BOW	20000	0.133/0.153	0.043/0.048	0.000/0.001
BOW-PCA	20	0.311/0.314	0.101/0.102	0.001/0.001
NetSim	NA	0.406/0.420	0.131/0.132	**0.002/0.002**
NetSim-PCA	500	**0.445/0.439**	**0.149/0.147**	**0.002/0.002**
Word2Vec	200	0.260/0.249	0.084/0.080	0.001/0.001
GCCA	50	0.269/0.276	0.089/0.091	0.001/0.001
GCCA-sv	500	**0.445/0.439**	**0.149/0.147**	**0.002/0.002**
GCCA-net	20	0.376/0.364	0.123/0.120	0.001/0.001
NetSize	NA	0.033/0.035	0.009/0.010	0.000/0.000
Random	NA	0.034/0.036	0.010/0.010	0.000/0.000

Table 2: Macro performance for friend recommendation. Performance of *NetSim-PCA* and *GCCA-sv* are identical since the view weighting for *GCCA-sv* only selected solely the friend view. Thus, these methods learned identical user embeddings.

Model	age	gender	politics
BOW	0.771/0.740	0.723/0.662	0.934/0.975
BOW-PCA	0.784/0.649	0.719/0.662	0.908/0.900
BOW-PCA + BOW	0.767/0.688	0.660/0.714	**0.937/0.9875**
GCCA	0.725/0.740	0.742/0.714	0.899/0.8125
GCCA + BOW	0.764/0.727	0.657/0.701	0.940/0.9625
GCCA-sv	0.709/0.636	0.699/0.714	0.871/0.850
GCCA-sv + BOW	0.761/0.688	0.647/0.675	0.937/0.9625
Word2Vec	**0.790/0.753**	**0.777/0.766**	0.927/0.938

Table 3: Average CV/test accuracy for inferring demographic characteristics.

net and *GCCA-sv*) improves the performance further. The best performing *GCCA* setting placed weight 1 on the ego tweet view, mention view, and friend view, while *BOW-PCA* concatenated these views, suggesting that these were the three most important views but that *GCCA* was able to learn a better representation. Figure 2 compares performance of different view subsets for *GCCA* and *BOW-PCA*, showing that *GCCA* uses information from multiple views more effectively for predicting user engagement.

Friend Recommendation Table 2 shows results for friend prediction and Figure 3 similarly shows that performance differences between approaches are consistent across k (number of recommendations.) Adding network views to *GCCA*, *GCCA-net*, improves performance, although it cannot contend with *NetSim* or
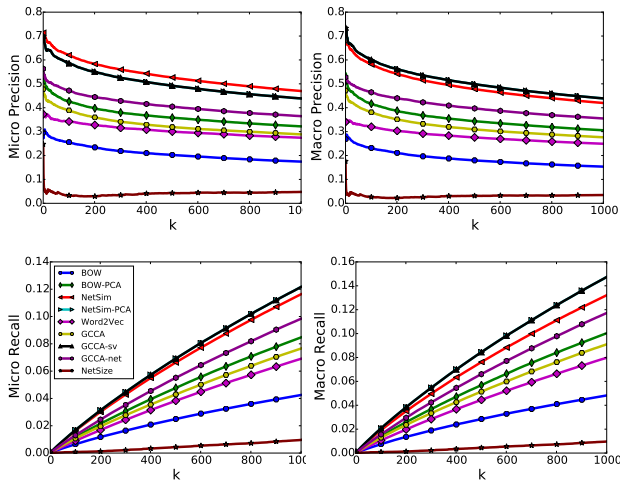
Figure 3: Performance on friend recommendation varying k.

NetSim-PCA, although *GCCA-sv* is able to meet the performance of *NetSim-PCA*. The best *GCCA* placed non-zero weight on the friend tweets view, and *GCCA-net* only places weight on the friend network view; the other views were not informative. *BOW-PCA* and *Word2Vec* only used the friend tweet view. This suggests that the friend view is the most important for this task, and multiview techniques cannot exploit additional views to improve performance. *GCCA-sv* performs identically to *GCCA-net*, since it only placed weight on the friend network view, learning identical embeddings to *GCCA-net*.

Demographic Characteristics Prediction Table 3 shows the average cross-fold validation and test accuracy on the demographic prediction task. *GCCA + BOW* and *BOW-PCA + BOW* are the concatenation of *BOW* features with *GCCA* and *BOW-PCA*, respectively. The wide variation in performance is due to the small size of the datasets, thus it's hard to draw many conclusions other than that *GCCA* seems to perform well compared to other linear methods. *Word2Vec* surpasses other representations in two out of three datasets.

It is difficult to compare the performance of the methods we evaluate here to that reported in previous work, (Al Zamal et al., 2012). This is because they report cross-fold validation accuracy (not test), they consider a wider range of hand-engineered features, different subsets of networks, radial basis function kernels for SVM, and find that accuracy varies wildly across different feature sets. They report cross-fold validation accuracy ranging from 0.619 to 0.805 for predicting age, 0.560 to 0.802 for gender, and 0.725 to 0.932 for politics.

7 Conclusion

We have proposed several representations of Twitter users, as well as a multiview approach that combines these views into a single embedding. Our multiview embeddings achieve promising results on three different prediction tasks, making use of both what a user writes as well as the social network. We found that each task relied on different information, which our method successfully combined into a single representation.

We plan to consider other means for learning user representations, including comparing nonlinear dimensionality reduction techniques such as kernel PCA (Schölkopf et al., 1997) and deep canonical correlation analysis (Andrew et al., 2013; Wang et al., 2015). Recent work on learning user representations with multitask deep learning techniques (Li et al., 2015), suggests that learning a nonlinear mapping from observed views to the latent space can learn high quality user representations. One issue with *GCCA* is scalability: solving for G relies on an SVD of a large matrix that must be loaded into memory. Online variants of *GCCA* would allow this method to scale to larger training sets and incrementally update representations. The PCA-reduced views for all 102,328 Twitter users can be found here: `http://www.dredze.com/datasets/multiview_embeddings/`.

References

Fabian Abel, Qi Gao, Geert-Jan Houben, and Ke Tao. 2011. Analyzing user modeling on twitter for personalized news recommendations. In *User Modeling, Adaption and Personalization - 19th International Conference, UMAP 2011, Girona, Spain, July 11-15, 2011. Proceedings*, pages 1–12.

Faiyaz Al Zamal, Wendy Liu, and Derek Ruths. 2012. Homophily and latent attribute inference: Inferring latent attributes of twitter users from neighbors. In *Internation Conference on Weblogs and Social Media (ICWSM)*.

Galen Andrew, Raman Arora, Jeff Bilmes, and Karen Livescu. 2013. Deep canonical correlation analysis. In *International Conference on Machine Learning (ICML)*, pages 1247–1255.

Raman Arora and Karen Livescu. 2014. Multi-view learning with supervision for transformed bottleneck features. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 2499–2503. IEEE.

J Douglas Carroll. 1968. Generalization of canonical correlation analysis to three or more sets of variables. In *Convention of the American Psychological Association*, volume 3, pages 227–228.

Xin Chen, Yu Wang, Eugene Agichtein, and Fusheng Wang. 2015. A comparative study of demographic attribute inference in twitter. In *Conference on Weblogs and Social Media (ICWSM)*.

Manaal Faruqui and Chris Dyer. 2014. Improving vector space word representations using multilingual correlation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2014, April 26-30, 2014, Gothenburg, Sweden*, pages 462–471.

Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2015. Skip-thought vectors. *CoRR*, abs/1506.06726.

Su Mon Kywe, Tuan-Anh Hoang, Ee-Peng Lim, and Feida Zhu. 2012. On recommending hashtags in twitter networks. In *Social Informatics - 4th International Conference, SocInfo 2012, Lausanne, Switzerland, December 5-7, 2012. Proceedings*, pages 337–350.

Quoc V Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Internation Conference on Machine Learning (ICML)*.

Jiwei Li, Alan Ritter, and Dan Jurafsky. 2015. Learning multi-faceted representations of individuals from heterogeneous evidence using neural networks. *arXiv preprint arXiv:1510.05198*.

David Liben-Nowell and Jon Kleinberg. 2007. The link-prediction problem for social networks. *Journal of the American society for information science and technology*, 58(7):1019–1031.

Chunliang Lu, Wai Lam, and Yingxiao Zhang. 2012. Twitter user modeling and tweets recommendation based on wikipedia concept graph. In *IJCAI Workshop on Intelligent Techniques for Web Personalization and Recommender Systems*.

Marco Lui and Timothy Baldwin. 2012. langid. py: An off-the-shelf language identification tool. In *Association for Computational Linguistics (ACL): system demonstrations*, pages 25–30.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems (NIPS)*, pages 3111–3119.

Youssef Mroueh, Etienne Marcheret, and Vaibhava Goel. 2015. Asymmetrically weighted cca and hierarchical kernel sentence embedding for multimodal retrieval. *arXiv preprint arXiv:1511.06267*.

Hemant Purohit, Yiye Ruan, Amruta Joshi, Srinivasan Parthasarathy, and Amit Sheth. 2011. Understanding user-community engagement by multi-faceted features: A case study on twitter. In *WWW Workshop on Social Media Engagement (SoME)*.

Pushpendre Rastogi, Benjamin Van Durme, and Raman Arora. 2015. Multiview lsa: Representation learning via generalized cca. In *North American Association for Computational Linguistics (NAACL)*.

Bernhard Schölkopf, Alexander J. Smola, and Klaus-Robert Müller. 1997. Kernel principal component analysis. In *Artificial Neural Networks - ICANN '97, 7th International Conference, Lausanne, Switzerland, October 8-10, 1997, Proceedings*, pages 583–588.

Jieying She and Lei Chen. 2014. Tomoha: Topic model-based hashtag recommendation on twitter. In *International conference on World wide web (WWW)*, pages 371–372. International World Wide Web Conferences Steering Committee.

Michel Van De Velden and Tammo HA Bijmolt. 2006. Generalized canonical correlation analysis of matrices with missing rows: a simulation study. *Psychometrika*, 71(2):323–331.

Svitlana Volkova, Glen Coppersmith, and Benjamin Van Durme. 2014. Inferring user political preferences from streaming communications. In *Association for Computational Linguistics (ACL)*, pages 186–196.

Svitlana Volkova. 2015. *Predicting Demographics and Affect in Social Networks*. Ph.D. thesis, Johns Hopkins University, October.

Weiran Wang, Raman Arora, Karen Livescu, and Jeff Bilmes. 2015. On deep multi-view representation learning. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 1083–1092.

Eva Zangerle, Wolfgang Gassler, and Günther Specht. 2013. On the impact of text similarity functions on hashtag recommendations in microblogging environments. *Social Network Analysis and Mining*, 3(4):889–898.