Putting Things in Context: Community-specific Embedding Projections for Sentiment Analysis

Yi Yang and **Jacob Eisenstein**
School of Interactive Computing
Georgia Institute of Technology
Atlanta, GA 30308
{yiyang+jacobe}@gatech.edu

Abstract

Variation in language is ubiquitous, and is particularly evident in newer forms of writing such as social media. Fortunately, variation is not random, but is usually linked to social factors. By exploiting linguistic homophily — the tendency of socially linked individuals to use language similarly — it is possible to build models that are more robust to variation. In this paper, we focus on social network *communities*, which make it possible to generalize sociolinguistic properties from authors in the training set to authors in the test sets, without requiring demographic author metadata. We detect communities via standard graph clustering algorithms, and then exploit these communities by learning community-specific projections of word embeddings. These projections capture shifts in word meaning in different social groups; by modeling them, we are able to improve the overall accuracy of Twitter sentiment analysis by a significant margin over competitive prior work.

1 Introduction

Words can mean different things to different people. Fortunately, these differences are rarely idiosyncratic, but are usually linked to social factors, such as age (Rosenthal and McKeown, 2011), gender (Eckert and McConnell-Ginet, 2003), race (Green, 2002), geography (Trudgill, 1974), and more ineffable characteristics such as political and cultural attitudes (Fischer, 1958; Labov, 1963). In natural language processing (NLP), the recent surge of interest in social media has brought variation to the fore, with papers presenting computational techniques for *characterizing* variation in the lexicon (Eisenstein et al., 2010; Gouws et al., 2011), orthography (Eisenstein, 2015), and syntax (Stewart, 2014; Johannsen et al., 2015). However, aside from attempts to normalize spelling variants (Sproat et al., 2001; Aw et al., 2006; Han and Baldwin, 2011; Yang and Eisenstein, 2013), there have been few attempts to address the problems posed by variation for the accuracy of NLP systems.

One recent exception is the work on Hovy (2015), who shows that the accuracy of sentiment analysis and topic classification can be improved by the inclusion of coarse-grained author demographics such as age and gender. However, such demographic information is not directly available in most datasets, and it is not yet clear whether predicted age and gender offers any improvements. On the other end of the spectrum are attempts to create *personalized* language technologies, as are often employed in information retrieval (Shen et al., 2005), recommender systems (Basilico and Hofmann, 2004), and language modeling (Federico, 1996). But personalization requires annotated data for each individual user

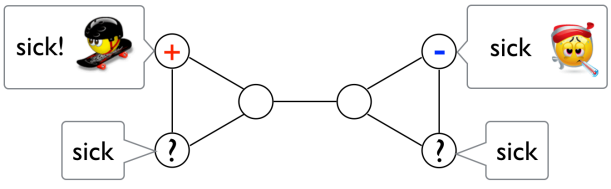

Figure 1: Words such as 'sick' can express opposite sentiment polarities depending on the user. We account for this variation by generalizing across the social network.

— something that may be possible in interactive settings such as information retrieval, but is not typically feasible in natural language processing.

We propose a middle ground between group-level demographic characteristics and personalization, by exploiting social network structure. The sociological theory of *homophily* asserts that individuals are usually similar to their friends (McPherson et al., 2001). This property has been demonstrated both for language (Puniyani et al., 2010; Bryden et al., 2013) as well as for the demographic properties targeted by Hovy (2015), which are more likely to be shared by friends than by random pairs of individuals (Thelwall, 2009; Al Zamal et al., 2012). Social network information is available in a wide range of contexts, from social media (Huberman et al., 2008) to political speech (Thomas et al., 2006) to historical texts (Winterer, 2012). Thus, social network homophily has the potential to provide a general and effective way to account for linguistic variation in NLP.

Figure 1 gives a schematic of the motivation for our approach. The word 'sick' typically has a negative sentiment, e.g., 'I would like to believe he's sick rather than just mean and evil'.[1] However, in some communities the word can have a positive sentiment, e.g., the lyric 'this sick beat', recently trademarked by the musician Taylor Swift. Given labeled examples of 'sick' in use by individuals in a social network, we assume that the word will have a similar sentiment meaning for their near neighbors — an assumption of *linguistic homophily* that is the basis for this research. Note that this differs from the assumption of *label homophily*, which entails that neighbors in the network will hold similar opinions, and will therefore produce similar document-level labels (Thomas et al., 2006; Tan et al., 2011; Hu et al., 2013). Linguistic homophily is a more generalizable claim, which could in principle be applied to any language processing task where author network information is available.

To scale this idea to datasets with tens of thousands of unique authors, we compress the social network, using algorithms for *community detection* (Fortunato, 2010). Community detection algorithms place each social network node into one of a finite number of communities; the goal is typically to maximize *modularity*, which rewards link density within each community, and penalizes link density across communities. A community detection algorithm applied to Figure 1 would likely identify the two triads as separate communities, as they are each completely connected, with only one edge between them. Applying this idea to language, we assume that linguistic meaning is relatively consistent within communities, and potentially distinct across communities — an assumption we call *linguistic modularity*.

To exploit the community structure of the author social network, we induce community-specific projection matrices, which project word embeddings into community-specific spaces. We apply this idea to Twitter sentiment classification, gathering social network metadata for Twitter users in the SemEval Twitter sentiment analysis tasks (Nakov et al., 2013; Rosenthal et al., 2015). Results are positive; building on the competitive NLSE system of Astudillo et al. (2015), we achieve significant improvements on the 2014 and 2015 tasks.

2 Community Detection

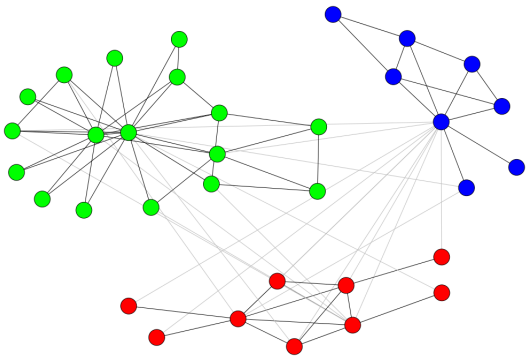

Figure 2: Zachary's Karate Club network (Zachary, 1977) with color indicating the three communities detected by the Fast-Greedy algorithm, described in Section 2.1.

Community detection in networks aims at identifying clusters or/and their hierarchical organization, by only using the information encoded in the network topology (as shown in Figure 2). This problem is of considerable interest to many areas, with applications to food webs, citation networks, social networks, and the web; in the NLP context, community

[1] Charles Rangel, describing Dick Cheney

detection has previously been used for word sense induction, by identifying communities of senses in a co-occurrence graph (Jurgens, 2011). In practice, community detection is typically treated as an optimization problem, where the objective function captures the intuition of a network community as set of nodes with better internal connectivity than external connectivity. Specifically, many of the best-known community detection algorithms seek to maximize *modularity*, which compares the density of edges within communities to the density of edges between communities (Newman, 2006).

In an undirected network over n nodes, let us write $a_{ij} \in \{0, 1\}$ to indicate the presence of an edge between nodes i and j, with $m \triangleq \frac{1}{2}\sum_i^n \sum_j^n a_{ij}$, the total number of edges. Then the modularity can be defined as,

$$Q = \frac{1}{2m} \sum_{i=1,j=1}^{n} (\mathbf{a}_{ij} - E[\mathbf{a}_{ij}])\delta(c_i, c_j), \quad (1)$$

with $c_i \in \{1, 2, \ldots, K\}$ indicating the community of node i, and $\delta(c_i, c_j)$ indicating whether i and j are placed in the same community. The term $E[\mathbf{a}_{ij}]$ represents the expected value of a_{ij} under a random "rewiring" of the network that preserves the degree of nodes i and j. The absolute value of the modularity is strictly less than one, where positive modularity indicates that the edge density within communities is more than would be expected by chance. In the limiting cases of $K = 1$ and $K = N$, the modularity is zero. The problem of finding a community partitioning that maximizes the modularity is NP-complete, regardless of whether the number of communities is specified (Brandes et al., 2008). We therefore consider two well-known approximate algorithms.

2.1 Community Detection Algorithms

Fast-Greedy *Fast-Greedy* is a hierarchical approach that detects communities in a bottom-up fashion (Clauset et al., 2004). Initially, every node belongs to a separate community, and communities are merged iteratively such that each merge is locally optimal. Specifically, in each iteration, the algorithm scans over every edge that joins two separate communities in the current configuration, and computes the increase in the current value of modularity ΔQ by merging the two communities together. The edge that yields the largest positive ΔQ would be adopted to merge the two communities at the end of this iteration. The algorithm stops when it is not possible to increase the modularity any more, and results in the community structure as well as the dendrogram describing the community structure. By utilizing efficient data structures, the running time of Fast-Greedy on a network with n nodes and m edges is $\mathcal{O}(md \log n)$, where d is the depth of the dendrogram.

Multi-Level *Multi-Level* is also a hierarchical approach that optimizes modularity in a greedy manner (Blondel et al., 2008). Initially, the algorithm starts with each node in its own community. Different from Fast-Greedy, which makes the decision to merge communities until the end of each iterations, Multi-Level re-assigns communities in a local, faster way. In particular, in each step, the algorithm sweeps over all communities in the current configuration sequentially, where each community would be merged with one of its neighbor community (i.e., there exists edges between nodes of the two communities), as long as the resulting merged community corresponds to the largest positive increase of modularity. The process stops when there is only a single community left or when the modularity cannot be increased any more. This procedure resembles how we perform Brown clustering (Brown et al., 1992) for NLP problems, where the clustering starts with each word in its own cluster, and the algorithm merges the two word clusters that maximizes the quality of the resulting clustering in each iteration. The distinction is that we are able to merge any two word clusters in Brown clustering, while we can only merge two neighbor communities in Multi-Level algorithm. The complexity of Multi-Level is approximating $\mathcal{O}(m)$, as most of the computation is concentrated on the first iteration.

2.2 Linguistic Modularity in Social Network Communities

The hypothesis of *linguistic homophily*, as presented in the introduction, is that socially connected individuals are likely to be more linguistically similar than a randomly selected pair of individuals. Having focused on community detection as a summary

of social network structures, we now refine this hypothesis to *linguistic modularity*: linguistic similarity should be higher for pairs of individuals who are in the same community than for pairs of individuals in different communities.

To formalize linguistic modularity, we follow the traditional social network definition of modularity closely. Suppose we are given a social network G, on which we induce a community structure $C = \{c_i\}$. We now create an alternative network $G^{(L)}$, representing linguistic similarity. In $G^{(L)}$, each node has an edge to the K most linguistically similar nodes, where linguistic similarity is measured by TF-IDF weighted cosine similarity. We can then compute modularity of $G^{(L)}$ with respect to the community structure C, which was computed from the original network G. This measure will quantify whether the communities identified from the social network structure tend to include linguistically-similar pairs of nodes. Linguistic modularity inherits all the properties of traditional social network modularity: the absolute value is less than one; the modularity is zero for trivial community structures of a single community or one community per node; the expected modularity is zero for a randomly-chosen community structure.

3 Testing Linguistic Modularity

We evaluate the linguistic modularity of the Fast-Greedy and Multi-Level community detection algorithms on a social network constructed from the SemEval Twitter Sentiment Analysis evaluation. Specifically, we start with the users in the SemEval datasets, and then use the Twitter API to crawl their friend links as undirected edges (in Twitter terminology, a individual's "friends" are accounts that the individual follows). The statistics of this network are shown in the first line of Table 1. As shown, the average degree for each node is less than three, and more than 25% of authors are *isolates*, meaning that they have no social network connections. For these users, the social network can provide no useful information.

One approach to increase the density of the social network would be to crawl additional nodes by snowball sampling (Goodman, 1961): specifically, we would expand the set of nodes to include all friends of nodes in the original SemEval network. However, there are more than seven million such individuals, which would be too many to crawl, given the limitations of the Twitter API. We therefore focus on adding nodes who will do the most to densify the author network, adding all individuals who are followed by at least 100 SemEval authors if they have less than ten thousand friends. We call the resulting network SEMEVAL+, and its statistics are shown in the second line of Table 1; the proportion of isolated nodes has fallen to 5%. Note that all data was acquired in April 2015; by this time, the authorship information of 11.4% of the tweets in our SemEval datasets was no longer available.

Network	# Author	# Relation	# Isolates
SEMEVAL	14,087	40,110	3,633
SEMEVAL+	17,417	1,050,369	689

Table 1: Statistics of social networks studied in this work. *Isolates* are individual nodes with no social network ties.

Why the follower network? Huberman et al. (2008) argue that Twitter's follower network is not a true social network because many users have an unreasonably large number of followers and friends, exceeding a supposed cognitive upper limit on the size of human social networks known as Dunbar's number (Dunbar, 1992; Dunbar, 1998). Huberman et al. (2008) instead advocate for a *mention network*, in which edges are drawn between each pair of individual who mention each others' usernames in their posts. Later work showed that the mention network has strong linguistic properties: it is better correlated with each author's distribution over latent topics as induced by latent Dirichlet allocation (Blei et al., 2003) than the follower network (Puniyani et al., 2010). However, the mention network is considerably sparser than the follower network, and contains a much larger number of isolated individuals, as many authors simply do not mention other usernames in their tweets — and many others are correspondingly never mentioned. Thus, while the mention network is an intriguing resource for future work, we do not employ it here.

Network	Algo.	# Cmty	$Q^{(L)}$
SEMEVAL	Fast-Greedy	19	0.1549
SEMEVAL+		5	0.2102
SEMEVAL	Multi-Level	19	0.1462
SEMEVAL+		7	0.2526

Table 2: Linguistic modularities on the SemEval data

3.1 Quantifying linguistic modularity

We measure linguistic modularity as described in subsection 2.2 on the SEMEVAL and SEMEVAL+ networks, using the Fast-Greedy and Multi-Level community detection algorithms. Linguistic information was computed from the most recent 3,200 tweets from each author, and the linguistic network $G^{(L)}$ was constructed by building edges to the five nearest neighbors. We merge all the communities with less than ten users into a single community, which mostly consists of isolates. The number of communities is selected automatically by the algorithm in all cases.

Results are presented in Table 2. All results are significantly higher than zero by a bootstrap resampling test, $p \ll 0.001$, indicating that the hypothesis of linguistic modularity holds for this data and for both community detection algorithms. The linguistic modularity is substantially higher for SEMEVAL+, indicating that crawling additional users has improved the quality of the detected communities from a linguistic standpoint. As an upper bound, we also computed the linguistic modularity from a text-based K-means clustering on the same authors, with the same numbers of clusters (5, 7, and 19), obtaining values of $Q^{(L)}$ between 0.43 and 0.47. Recall that the social network community detection algorithms form communities *without considering the text*, so it is unsurprising that the modularity values for K-means are higher. The fact that we are able to obtain linguistic modularities that are significantly higher than zero on purely social network data shows that social network communities can provide useful complementary information for language processing tasks. Finally, we note that the results do not demonstrate a clear winner between Fast-Greedy and Multi-Level, so we consider both approaches in the supervised sentiment analysis task that follows.

4 Community-specific Word Embeddings

In this section, we propose to leverage social network community structures for improving NLP tasks. We build on the successful non-linear subspace embedding (NLSE) model, which learns a *task-specific* projection matrix to apply to pre-trained word embeddings (Astudillo et al., 2015),

$$\mathbf{h}_j = \text{Sigmoid}(\mathbf{S}\mathbf{w}_j), \tag{2}$$

where $\mathbf{w}_j \in \mathbb{R}^d$ is a pre-trained word embedding for word j, and $\mathbf{S} \in \mathbb{R}^{k \times d}$ is the task-specific projection matrix, with d and k indicating the sizes of the pre-trained and projected word embeddings respectively. The advantage of this approach is that the pre-trained word embeddings can be learned from large amounts of unlabeled text, while supervised training is needed only to learn the $s \times d$ matrix $\mathbf{S}$, and the final classification weights. By simply adding the projected embeddings across all words in a Tweet, Astudillo et al. (2015) obtain a Tweet representation that performs very well on sentiment analysis: a Softmax classifier applied on top of this representation outperforms all alternatives on the SemEval 2014 and 2015 Twitter sentiment analysis tasks.

We incorporate social network communities into the NLSE model, in a method we call COMMSEM, for **Comm**ity-**S**pecific **Em**beddings. Specifically, we induce additional projection matrices per community, so that for document i in community c_i, we have,

$$\mathbf{h}_{i,j} = \text{Sigmoid}(\left[\mathbf{S}^{(0)} + \mathbf{S}^{(c_i)}\right] \mathbf{w}_j), \tag{3}$$

where $\mathbf{S}^{(0)}$ is the community independent projection matrix, $\mathbf{S}^{(c_i)}$ is the community-specific projection matrix associated with the community c_i, and $\mathbf{h}_{i,j}$ is the embedding of word j in community c_i. Intuitively, $\mathbf{S}^{(0)}$ encourages COMMSEM to take advantage of the statistical power given by modeling all the training data, and $\mathbf{S}^{(c_i)}$ provides the flexibility to capture community-specific information. To avoid overfitting, we penalize the community-specific projections with L_2 regularization. The overall setup is illustrated in Figure 3.

Because the community detection problem is NP-complete, both Fast-Greedy and Multi-Level are heuristic approximations, and the specific local optima that they obtain depends on the initialization

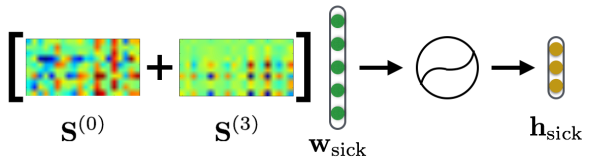

Figure 3: An illustration of obtaining community-specific embedding via COMMSEM for the word "sick" in community 3.

Dataset	# Positive	# Negative	# Neutral	Total
Development	3,230	1,265	4,109	8,604
Test 2013	1,572	601	1,640	3,813
Test 2014	982	202	669	1,853
Test 2015	1,038	365	987	2,390

Table 3: Statistics of SemEval Twitter sentiment datasets.

and ordering of nodes in the graph. By running these algorithms over multiple orderings, we can obtain multiple community structures, and by averaging over these structures, we may obtain more reliable results. This idea can be easily incorporated in COMMSEM. Suppose we obtain L distinct community structures by running the community detection algorithms multiple times. Then we can compute multi-community word embeddings for instance i as,

$$\mathbf{h}_{i,j} = \text{Sigmoid}\left(\left[\mathbf{S}^{(0)} + \sum_{k=1}^{L} \mathbf{S}^{(c_i^{(k)})}\right] \mathbf{w}_j\right), \quad (4)$$

where $c_i^{(k)}$ is the community for instance i in community structure k.

5 Experiments

We test the utility of community-specific projections on the 2013–2015 SemEval Twitter sentiment analysis tasks, in which the system classifies whether a given message is of positive, negative, or neutral sentiment. Following Astudillo et al. (2015), we train and tune our system using only the development data, which is the SemEval Twitter 2013 training data, and evaluate on the SemEval Twitter test sets. The statistics of these datasets are shown in Table 3. The datasets adopted in this work are from (Astudillo et al., 2015), which consist of entire tweets of the original test sets, but only a subset of development tweets. Furthermore, we fail to identify the authors of about 11.4% of the tweets in the datasets, as the tweets were not available by the time we started to crawl social relations.

Experimental Settings We employ the best performing pre-trained word embeddings used by Astudillo et al. (2015), which are trained with a corpus of 52 million tweets. The embeddings are learned using the structured skip-gram model (Ling et al., 2015), and the embedding dimension is set as 600. We also pre-process the SemEval Twitter messages using the same pre-processing procedure for the embedding training corpus. We report the same evaluation metric as the SemEval challenge: the Average F1 score of positive and negative classes. For the social network, we use only the SEMEVAL+ network, since the linguistic modularity was considerably higher for communities extracted from this network than from the original SEMEVAL network (Table 2).

Competitive systems We consider two competitive Twitter sentiment classification methods based on NLSE, the sub-space projection method described in section 4. The *pooled* setting ignores the community structure information, and only estimates one NLSE model with all the training data. The *per-community* setting leverages the community information by training multiple independent NLSE models, one per community, and then apply each model to the test set messages from the same community. Aside from NLSE, we also compare against the three top-performing systems in the SemEval 2015 Twitter sentiment analysis challenge (Rosenthal et al., 2015): WEBIS (Hagen et al., 2015), UNITN (Severyn and Moschitti, 2015) and LSISLIF (Hamdan et al., 2015). Our implementation of NLSE model performs slightly worse than the results reported in the original paper (Astudillo et al., 2015), so we also republish these results. Our implementation of COMMSEM is built on top of our re-implementation of NLSE, so any improvements to the NLSE implementation may also improve COMMSEM.

Parameter tuning We adopt the same training and validation data splits as Astudillo et al. (2015), where 80% of the development data is used for pa-

rameter learning and 20% for model selection.[2] We choose the subspace size from {5, 10, 15}, and the L_2 regularization penalty from {1.0, 3.0, 5.0, 8.0, 10.0, 20.0} for COMMSEM after preliminary search. As described in section 4, it is possible to apply COMMSEM to multiple overlapping community structures by running the community detection algorithms multiple times. We compare a range of numbers of community structures, {1, 3, 5, 10}.

Results Evaluation results are presented in Table 4. COMMSEM achieves similar results as pooled NLSE on the SemEval Twiter 2013 test set, but it significantly outperforms pooled NLSE on the 2014 and 2015 test sets, $p < .05$ ($\chi^2 \approx 4.67$ and 4.12, $dof = 1$) by McNemar's test. One way in which the 2014 and 2015 test sets are more difficult than the 2013 test set is that they are drawn from different time periods than the training data. NLSE performs considerably worse when trained separately within each community, probably due to the smaller amount of available training data. Based on dev set tuning, the optimal number of community structures used by COMMSEM is one for Fast-Greedy community detection algorithm, and three for Multi-Level community detection algorithm. Multi-Level is more sensitive to initialization and node ordering than Fast-Greedy, and the resulting community structures contain more additional useful information for COMMSEM.

5.1 Analysis

We investigate what information has been captured by the community-specific embeddings. We simply compute the Euclidean distance between the community-specific embedding $\mathbf{h}_{i,j}$ and the embedding obtained from the shared projection $\mathbf{h}_{0,j} = \text{Sigmoid}(\mathbf{S}^{(0)}\mathbf{w}_j)$. The top ten words that with highest Euclidean distances corresponding to each community are presented in Table 5. We exclude the special community merged from small communities as well as one community with only 62 tweets in the training data. It is interesting to note that the metric helps identify many community-specific sentiment words, which are mostly related to positive sentiment for community 1 and 3 and negative sentiment for community 2. Although the ratios of positive and negative tweets posted by users in these communities are generally similar, the words are dominated by one particular sentiment polarity in each community, which implies that users in one community may prefer to use sentiment words differently with respect to only one sentiment.

6 Related Work

Domain adaptation and personalization Community-specific NLP can be viewed as a form of domain adaptation, where the domains are determined automatically by a graph clustering algorithm. Our approach is then a form of supervised domain adaptation, since we assume some labeled data within each community (Søgaard, 2013). Early approaches to supervised domain adaptation focused on adapting the classifier weights across communities, using enhanced feature spaces (Daumé III, 2007) or Bayesian priors (Chelba and Acero, 2006; Finkel and Manning, 2009).[3] In contrast, our goal is to learn a transformation of the input representation for each community. In this way, our work is more similar in spirit to unsupervised domain adaptation, which typically works by transforming the input feature space so as to overcome domain differences (Blitzer et al., 2006; Ben-David et al., 2010; Chen et al., 2012; Glorot et al., 2011). In our case, the transformation is learned discriminatively; an interesting possibility for future work would be the exploration of new training objectives that combine domain generality (Ben-David et al., 2010) with the label log-likelihood as considered here.

Personalization has been an active research topic in many other areas, such as speech recognition (speaker adaptation) and information retrieval (personalized search). Standard techniques for these tasks include linear transformation of model parameters (Leggetter and Woodland, 1995) and collaborative filtering (Breese et al., 1998). These methods have been adopted to personalized sentiment analysis recently (Tang et al., 2015; Al Boni et al., 2015). Supervised personalization typically requires labeled training examples for every individual user,

[2] We obtained these splits through communication with the authors.

[3] We adopted Easy Adaptation approach (Daumé III, 2007) in our preliminary experiments, and observed marginal improvements.

System	Communities	T2013	T2014	T2015	Avg
Our method					
COMMSEM	Fast-Greedy	72.01	73.70	65.56	70.42
COMMSEM	Multi-Level	72.10	**73.78***	**66.13***	**70.67**
NLSE (Our implementation)					
Pooled		71.96	72.72	65.19	69.96
Per-community	Fast-Greedy	66.49	67.92	61.61	65.34
Per-community	Multi-Level	61.92	63.23	57.56	60.90
Reported results					
NLSE		72.09	73.64	65.21	70.31
WEBIS		68.49	70.86	64.84	68.06
UNITN		**72.79**	73.60	64.59	70.33
LSISLIF		71.34	71.54	64.27	69.05

Table 4: Average F1 score on the SemEval test sets. Results are marked with * if they are significant better than pooled NLSE at $p < 0.05$.

Community	# Positive	# Negative	Top words with highest Euclidean distances
1	467	190	brighter$^{(+)}$ blessings$^{(+)}$ celebrate$^{(+)}$ shine$^{(+)}$ brightest$^{(+)}$ blessing$^{(+)}$ gbu$^{(+)}$ glowing$^{(+)}$ celebrating$^{(+)}$ hwaiting$^{(+)}$
2	845	340	mistakes$^{(-)}$ stfu$^{(-)}$ ass$^{(-)}$ shutup$^{(-)}$ bitch$^{(-)}$ #ohwell$^{(-)}$ retard$^{(-)}$ regret$^{(-)}$ dgaf$^{(-)}$ idgaf$^{(-)}$
3	868	302	enjoyin$^{(+)}$ #stillkidrauhl$^{(+)}$ funny$^{(+)}$ brighter$^{(+)}$ happiest$^{(+)}$ catchy$^{(+)}$ sweeter$^{(+)}$ jealous$^{(+)}$ relaxing$^{(+)}$ smartest$^{(+)}$

Table 5: Top ten words with highest Euclidean distances between the community-specific embeddings and the embeddings obtained from the shared projection corresponding to each community. Positive sentiment words are in red$^{(+)}$ and negative sentiment words are in blue$^{(-)}$.

which is generally not possible in NLP scenarios. We therefore leverage social network community structures to generalize across individual users.

Sentiment analysis with social relations Previous work on incorporating social relations for sentiment classification mainly relies on the label consistency assumption, where the existence of social connections between users is considered as a clue that the sentiment polarities of all messages from the users should be similar. Speriosu et al. (2011) construct a heterogeneous network with tweets, users, and n-grams as nodes, and each node is associated with a sentiment label distribution obtained from a maximum entropy classifier or sentiment lexicons. The label distributions of tweets is then refined by performing label propagation over social relations. Hu et al. (2013) model social relations using the graph Laplacian of the adjacency graph representation of the social network, which they employ as a source of regularization, so that socially-similar users are encouraged to have similar labels. Tan et al. (2011) leverage a similar intuition, using a factor graph based approach in which the labels of targets belonging to socially connected users are treated as factors in a joint probabilistic model. Our work is based on a different intuition: rather than assuming labels will tend to be similar for socially-connected users, we assume that similar usage of language. These assumptions are complementary; if both hold for a specific setting, then label consistency and linguistic consistency could in principle be applied to improve performance.

Customized word representations Our approach is based on community-specific projections of word embeddings. Similarly, Bamman et al. (2014) learn geographically-specific word embeddings, with the

goal of capturing geographical differences in meaning. Kulkarni et al. (2015) are interested in the changing meaning of words over time, and learn separate embeddings for different temporal epochs. Yang and Eisenstein (2015) learn multiple *feature* embeddings for a set of domain attributes, such as author, genre, and temporal epoch. All of these approaches differ from our work in that they learn multiple embeddings, rather than learning multiple *projections* of a single underlying embedding. This enables our approach to exploit very large, unlabeled data to learn the original word embeddings, and then to learn the much smaller community-specific projections using labeled data.

7 Conclusion

This paper presents a new method for learning to overcome language variation, leveraging the tendency of individuals with similar linguistic patterns to share social network connections — the phenomenon of *linguistic homophily*. By learning separate projections of word embeddings for each community, our approach is able to capture subtle shifts in meaning for individual words across social communities. We have formulated this model by building on prior work which learns task-specific projections of word embeddings; we go further by learning projections that are both task-specific and community-specific. A key question for future work is whether the task and community dimensions can be decoupled: can we learn a community-based projection that is useful across multiple tasks? Answering this question requires ground truth annotations for multiple tasks for a set of socially linked authors. This could perhaps be obtained by adding a layer of annotations on top of the SemEval Twitter Sentiment dataset, or by identifying some form of "found" annotations such as retweets or favorites. We plan on pursuing this direction in future work. Another direction for future research is to try to generalize linguistic homophily from its implementation in social network communities, using some form of continuous representation for individual author nodes, which should capture the node's social properties; very recent work by Li et al. (2015) suggests one possible approach.

8 Acknowledgments

We thank Duen Horng "Polo" Chau for discussions about community detection and Ramon Astudillo for sharing data and helping us to reproduce the NLSE results. This research was supported by the National Science Foundation under award RI-1452443, by the National Institutes of Health under award number R01GM112697-01, and by the Air Force Office of Scientific Research. The content is solely the responsibility of the authors and does not necessarily represent the official views of these sponsors.

References

Mohammad Al Boni, Keira Qi Zhou, Hongning Wang, and Matthew S Gerber. 2015. Model adaptation for personalized opinion analysis. In *Proceedings of the Association for Computational Linguistics (ACL)*, Beijing, China.

Faiyaz Al Zamal, Wendy Liu, and Derek Ruths. 2012. Homophily and latent attribute inference: Inferring latent attributes of Twitter users from neighbors. In *Proceedings of the International Conference on Web and Social Media (ICWSM)*, pages 387–390, Menlo Park, California. AAAI Publications.

Ramon F Astudillo, Silvio Amir, Wang Lin, Mário Silva, and Isabel Trancoso. 2015. Learning word representations from scarce and noisy data with embedding subspaces. In *Proceedings of the Association for Computational Linguistics (ACL)*, Beijing, China.

AiTi Aw, Min Zhang, Juan Xiao, and Jian Su. 2006. A phrase-based statistical model for SMS text normalization. In *Proceedings of ACL*, pages 33–40.

David Bamman, Chris Dyer, and Noah A. Smith. 2014. Distributed representations of geographically situated language. In *Proceedings of the Association for Computational Linguistics (ACL)*, pages 828–834, Baltimore, MD.

Justin Basilico and Thomas Hofmann. 2004. Unifying collaborative and content-based filtering. In *Proceedings of the twenty-first international conference on Machine learning*, page 9. ACM.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. 2010. A theory of learning from different domains. *Machine learning*, 79(1-2):151–175.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022.

John Blitzer, Ryan McDonald, and Fernando Pereira. 2006. Domain adaptation with structural correspondence learning. In *Proceedings of Empirical Methods for Natural Language Processing (EMNLP)*, pages 120–128.

Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008.

Ulrik Brandes, Daniel Delling, Marco Gaertler, Robert Görke, Martin Hoefer, Zoran Nikoloski, and Dorothea Wagner. 2008. On modularity clustering. *Knowledge and Data Engineering, IEEE Transactions on*, 20(2):172–188.

John S Breese, David Heckerman, and Carl Kadie. 1998. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pages 43–52. Morgan Kaufmann Publishers Inc.

Peter F Brown, Peter V Desouza, Robert L Mercer, Vincent J Della Pietra, and Jenifer C Lai. 1992. Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–479.

John Bryden, Sebastian Funk, and Vincent Jansen. 2013. Word usage mirrors community structure in the online social network twitter. *EPJ Data Science*, 2(1).

Ciprian Chelba and Alex Acero. 2006. Adaptation of maximum entropy capitalizer: Little data can help a lot. *Computer Speech & Language*, 20(4):382–399.

Minmin Chen, Z. Xu, Killian Weinberger, and Fei Sha. 2012. Marginalized denoising autoencoders for domain adaptation. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Aaron Clauset, Mark EJ Newman, and Cristopher Moore. 2004. Finding community structure in very large networks. *Physical review E*, 70(6):066111.

Hal Daumé III. 2007. Frustratingly easy domain adaptation. In *Proceedings of the Association for Computational Linguistics (ACL)*, Prague.

Robin I.M. Dunbar. 1992. Neocortex size as a constraint on group size in primates. *Journal of Human Evolution*, 22(6):469–493.

Robin I.M. Dunbar. 1998. The social brain hypothesis. *Evolutionary Antropology*, 6(178).

Penelope Eckert and Sally McConnell-Ginet. 2003. *Language and Gender*. Cambridge University Press, New York.

Jacob Eisenstein, Brendan O'Connor, Noah A. Smith, and Eric P. Xing. 2010. A latent variable model for geographic lexical variation. In *Proceedings of Empirical Methods for Natural Language Processing (EMNLP)*, pages 1277–1287, Stroudsburg, Pennsylvania. Association for Computational Linguistics.

Jacob Eisenstein. 2015. Systematic patterning in phonologically-motivated orthographic variation. *Journal of Sociolinguistics*, 19:161–188.

Marcello Federico. 1996. Bayesian estimation methods for n-gram language model adaptation. In *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*, volume 1, pages 240–243. IEEE.

Jenny R. Finkel and Christopher Manning. 2009. Hierarchical bayesian domain adaptation. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 602–610, Boulder, CO.

John L Fischer. 1958. Social influences on the choice of a linguistic variant. *Word*, 14:47–56.

Santo Fortunato. 2010. Community detection in graphs. *Physics Reports*, 486(3):75–174.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the International Conference on Machine Learning (ICML)*, Seattle, WA.

Leo A Goodman. 1961. Snowball sampling. *The annals of mathematical statistics*, pages 148–170.

Stephan Gouws, Donald Metzler, Congxing Cai, and Eduard Hovy. 2011. Contextual bearing on linguistic variation in social media. In *Proceedings of the ACL Workshop on Language and Social Media*.

Lisa J. Green. 2002. *African American English: A Linguistic Introduction*. Cambridge University Press, Cambridge, U.K., September.

Matthias Hagen, Martin Potthast, Michael Büchner, and Benno Stein. 2015. Webis: An ensemble for twitter sentiment detection. In *Proceedings of the 9th International Workshop on Semantic Evaluation*.

Hussam Hamdan, Patrice Bellot, and Frederic Bechet. 2015. lsislif: Feature extraction and label weighting for sentiment analysis in twitter. In *Proceedings of the 9th International Workshop on Semantic Evaluation*.

Bo Han and Timothy Baldwin. 2011. Lexical normalisation of short text messages: Makn sens a# twitter. In *Proceedings of ACL*, volume 1.

Dirk Hovy. 2015. Demographic factors improve classification performance. In *Proceedings of the Association for Computational Linguistics (ACL)*, pages 752–762, Beijing, China.

Xia Hu, Lei Tang, Jiliang Tang, and Huan Liu. 2013. Exploiting social relations for sentiment analysis in microblogging. In *Proceedings of the sixth ACM international conference on Web search and data mining (WSDM)*, pages 537–546.

Bernardo Huberman, Daniel M. Romero, and Fang Wu. 2008. Social networks that matter: Twitter under the microscope. *First Monday*, 14(1).

Anders Johannsen, Dirk Hovy, and Anders Søgaard. 2015. Cross-lingual syntactic variation over age and gender. In *Proceedings of the Conference on Natural Language Learning (CoNLL)*.

David Jurgens. 2011. Word sense induction by community detection. In *Proceedings of TextGraphs-6: Graph-based Methods for Natural Language Processing*, pages 24–28. Association for Computational Linguistics.

Vivek Kulkarni, Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. 2015. Statistically significant detection of linguistic change. In *Proceedings of the Conference on World-Wide Web (WWW)*, pages 625–635.

William Labov. 1963. The social motivation of a sound change. *Word*, 19(3):273–309.

Christopher J Leggetter and Philip C Woodland. 1995. Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models. *Computer Speech & Language*, 9(2):171–185.

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*.

Wang Ling, Chris Dyer, Alan Black, and Isabel Trancoso. 2015. Two/too simple adaptations of word2vec for syntax problems. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*, Denver, CO, May.

Miller McPherson, Lynn Smith-Lovin, and James M Cook. 2001. Birds of a feather: Homophily in social networks. *Annual review of sociology*, pages 415–444.

Preslav Nakov, Zornitsa Kozareva, Alan Ritter, Sara Rosenthal, Veselin Stoyanov, and Theresa Wilson. 2013. Semeval-2013 task 2: Sentiment analysis in twitter. In *Proceedings of the 7th International Workshop on Semantic Evaluation, SemEval*.

Mark EJ Newman. 2006. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582.

Kriti Puniyani, Jacob Eisenstein, Shay Cohen, and Eric P. Xing. 2010. Social links from latent topics in microblogs. In *Proceedings of NAACL Workshop on Social Media*, Los Angeles.

Sara Rosenthal and Kathleen McKeown. 2011. Age prediction in blogs: A study of style, content, and online behavior in pre- and Post-Social media generations. In *Proceedings of the Association for Computational Linguistics (ACL)*, pages 763–772, Portland, OR.

Sara Rosenthal, Preslav Nakov, Svetlana Kiritchenko, Saif M Mohammad, Alan Ritter, and Veselin Stoyanov. 2015. Semeval-2015 task 10: Sentiment analysis in twitter. In *Proceedings of the 9th International Workshop on Semantic Evaluation, SemEval*.

Aliaksei Severyn and Alessandro Moschitti. 2015. Unitn: Training deep convolutional neural network for twitter sentiment classification. In *Proceedings of the 9th International Workshop on Semantic Evaluation*.

Xuehua Shen, Bin Tan, and ChengXiang Zhai. 2005. Implicit user modeling for personalized search. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 824–831. ACM.

Anders Søgaard. 2013. Semi-supervised learning and domain adaptation in natural language processing. *Synthesis Lectures on Human Language Technologies*, 6(2):1–103.

Michael Speriosu, Nikita Sudan, Sid Upadhyay, and Jason Baldridge. 2011. Twitter polarity classification with label propagation over lexical links and the follower graph. In *Proceedings of Empirical Methods for Natural Language Processing (EMNLP)*, pages 53–63.

R. Sproat, A.W. Black, S. Chen, S. Kumar, M. Ostendorf, and C. Richards. 2001. Normalization of non-standard words. *Computer Speech & Language*, 15(3):287–333.

Ian Stewart. 2014. Now we stronger than ever: African-american syntax in twitter. In *Proceedings of the Student Research Workshop at the 14th Conference of the European Chapter of the Association for Computational Linguistics*.

Chenhao Tan, Lillian Lee, Jie Tang, Long Jiang, Ming Zhou, and Ping Li. 2011. User-level sentiment analysis incorporating social networks. In *Proceedings of Knowledge Discovery and Data Mining (KDD)*, pages 1397–1405.

Duyu Tang, Bing Qin, and Ting Liu. 2015. Learning semantic representations of users and products for document level sentiment classification. In *Proceedings of the Association for Computational Linguistics (ACL)*, Beijing, China.

Mike Thelwall. 2009. Homophily in MySpace. *J. Am. Soc. Inf. Sci.*, 60(2):219–231.

Matt Thomas, Bo Pang, and Lillian Lee. 2006. Get out the vote: Determining support or opposition from Congressional floor-debate transcripts. In *Proceedings of Empirical Methods for Natural Language Processing (EMNLP)*, pages 327–335.

Peter Trudgill. 1974. Linguistic change and diffusion: Description and explanation in sociolinguistic dialect geography. *Language in Society*, 3(2):215–246.

Caroline Winterer. 2012. Where is america in the republic of letters? *Modern Intellectual History*, 9(03):597–623.

Yi Yang and Jacob Eisenstein. 2013. A log-linear model for unsupervised text normalization. In *Proceedings of Empirical Methods for Natural Language Processing (EMNLP)*.

Yi Yang and Jacob Eisenstein. 2015. Unsupervised multi-domain adaptation with feature embeddings. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*, Denver, CO, May.

Wayne W Zachary. 1977. An information flow model for conflict and fission in small groups. *Journal of anthropological research*, pages 452–473.