Broadly Improving User Classification via Communication-Based Name and Location Clustering on Twitter

Shane Bergsma, Mark Dredze, Benjamin Van Durme, Theresa Wilson, David Yarowsky
Department of Computer Science and Human Language Technology Center of Excellence
Johns Hopkins University
Baltimore, MD 21218, USA
shane.a.bergsma@gmail.com, mdredze@cs.jhu.edu, vandurme@cs.jhu.edu, taw@jhu.edu, yarowsky@cs.jhu.edu

Abstract

Hidden properties of social media users, such as their ethnicity, gender, and location, are often reflected in their observed attributes, such as their first and last names. Furthermore, users who communicate with each other often have similar hidden properties. We propose an algorithm that exploits these insights to cluster the observed attributes of hundreds of millions of Twitter users. Attributes such as user names are grouped together if users with those names communicate with other similar users. We separately cluster millions of unique first names, last names, and user-provided locations. The efficacy of these clusters is then evaluated on a diverse set of classification tasks that predict hidden users properties such as ethnicity, geographic location, gender, language, and race, using only profile names and locations when appropriate. Our readily-replicable approach and publicly-released clusters are shown to be remarkably effective and versatile, substantially outperforming state-of-the-art approaches and human accuracy on each of the tasks studied.

1 Introduction

There is growing interest in automatically classifying users in social media by various hidden properties, such as their gender, location, and language (e.g. Rao et al. (2010), Cheng et al. (2010), Bergsma et al. (2012)). Predicting these and other properties for users can enable better advertising and personalization, as well as a finer-grained analysis of user opinions (O'Connor et al., 2010), health (Paul and Dredze, 2011), and sociolinguistic phenomena (Eisenstein et al., 2011). Classifiers for user properties often rely on information from a user's social network (Jernigan and Mistree, 2009; Sadilek et al., 2012) or the textual content they generate (Pennacchiotti and Popescu, 2011; Burger et al., 2011).

Here, we propose and evaluate classifiers that better exploit the *attributes* that users explicitly provide in their user profiles, such as names (e.g., first names like *Mary*, last names like *Smith*) and locations (e.g., *Brasil*). Such attributes have previously been used as "profile features" in supervised user classifiers (Pennacchiotti and Popescu, 2011; Burger et al., 2011; Bergsma et al., 2012). There are several motivations for exploiting these data. Often the only information available for a user is a name or location (e.g. for a new user account). Profiles also provide an orthogonal or complementary source of information to a user's social network and textual content; gains based on profiles alone should therefore add to gains based on other data. The decisions of profile-based classifiers could also be used to bootstrap training data for other classifiers that use complementary features.

Prior work has encoded profile attributes via lexical or character-based features (e.g. Pennacchiotti and Popescu (2011), Burger et al. (2011), Bergsma et al. (2012)). Unfortunately, due to the long-tailed distribution of user attributes, a profile-based classifier will encounter many examples at test time that were not observed during training. For example, suppose a user *wassim hassan* gives their location as *tanger*. If the attribute tokens *wassim*, *hassan*, and *tanger* do not occur in training (nor indicative sub-

Proceedings of NAACL-HLT 2013, pages 1010–1019,
Atlanta, Georgia, 9–14 June 2013.

strings), then a classifier can only guess at the user's ethnicity and location. In social media, the prevalence of fake names and large variations in spelling, slang, and language make matters worse.

Our innovation is to enhance attribute-based classifiers with new data, derived from the communications of Twitter users with those attributes. Users with the name tokens *wassim* and *hassan* often talk to users with Arab names like *abdul* and *hussein*. Users listing their location as *tanger* often talk to users from *morocco*. Since users who communicate often share properties such as ethnicity and location (§8), the user *wassim hassan* might be an Arab who uses the French spelling of the city *Tangier*.

Our challenge is to encode these data in a form readily usable by a classifier. Our approach is to represent each unique profile attribute (e.g. *tanger* or *hassan*) as a vector that encodes the communication pattern of users with that attribute (e.g. how often they talk to users from *morocco*, etc.); we then cluster the vectors to discover latent groupings of similar attributes. Based on transitive (third party) connections, *tanger* and *tangier* can appear in the same cluster, even if no two users from these locations talk directly. To use the clusters in an attribute-based classifier, we add new features that indicate the cluster memberships of the attributes. Clustering thus lets us convert a high-dimensional space of all attribute pairs to a low-dimensional space of cluster memberships. This makes it easier to share our data, yields fewer parameters for learning, and creates attribute groups that are interpretable to humans.

We cluster names and locations in a very large corpus of 168 million Twitter users (§2) and use a distributed clustering algorithm to separately cluster millions of first names, last names, and user-provided locations (§3). We evaluate the use of our cluster data as a novel feature in supervised classifiers, and compare our result to standard classifiers using character and token-level features (§4). The cluster data enables significantly improved performance in predicting the gender, location, and language of social media users, exceeding both existing state-of-the-art machine and human performance (§6). Our cluster data can likewise improve performance in other domains, on both established and new NLP tasks as further evaluated in this paper (§6). We also propose a way to enhance a geolocation system by using communication patterns, and show strong improvements over a hand-engineered baseline (§7). We share our clusters with the community to use with other tasks. The clusters, and other experimental data, are available for download from `www.clsp.jhu.edu/~sbergsma/TwitterClusters/`.

First names: maria, david, ana, daniel, michael, john, alex, jessica, carlos, jose, chris, sarah, laura, juan
Last names: silva, santos, smith, garcia, oliveira, rodriguez, jones, williams, johnson, brown, gonzalez
Locations: brasil, indonesia, philippines, london, jakarta, são paulo, rio de janeiro, venezuela, brazil

Table 1: Most frequent profile attributes for our collection of 168 million Twitter users, in descending order

2 Attribute Associations on Twitter

Data and Processing Our raw Twitter data comprises the union of 2.2 billion tweets from 05/2009 to 10/2010 (O'Connor et al., 2010), 1.8 billion tweets collected from 07/2011 to 08/2012, and 80 million tweets collected from followers of 10 thousand location and language-specific Twitter feeds.

We implemented each stage of processing using MapReduce (Dean and Ghemawat, 2008). The total computation (from extracting profiles to clustering attributes) was 1300 days of wall-clock CPU time.

Attribute Extraction Tweets provide the name and self-reported location of the tweeter. We find 126M unique users with these attributes in our data. When tweets mention other users via an @*user* construction, Twitter also includes the profile name of the mentioned user; we obtain a further 42M users from these cases. We then normalize the extracted attributes by converting to lower-case, deleting symbols, numbers, and punctuation, and removing common honorifics and suffixes like *mr*/*mrs* and *jr*/*sr*. Common prefixes like *van* and *de la* are joined to the last-name token.[1] This processing yields 8.3M

[1] `www.clsp.jhu.edu/~sbergsma/TwitterClusters/` also provides our scripts for normalizing attributes. The scripts can be used to ensure consistency/compatibility between arbitrary datasets and our shared cluster data. Note we use no special processing for the companies, organizations, and spammers among our users, nor for names arising from different conventions (e.g. 1-word names, reversed first/last names).

henrik: fredrik 5.87, henrik 5.82, anders 5.73, johan 5.69, andreas 5.59, martin 5.54, magnus 5.41
courtney: taylor 8.03, ashley 7.92, courtney 7.92, emily 7.91, lauren 7.82, katie 7.72, brittany 7.69
ilya: sergey 5.85, alexey 5.62, alexander 5.59, dmitry 5.51, Александр 5.46, anton 5.44, andrey 5.40

Table 2: Top associates and PMIs for three first names.

unique locations, 7.4M unique last names, and 5.5M unique first names. These three sets provide the target attributes that we cluster in §3. Table 1 shows the most frequent names in each of these three sets.

User-User Links We extract each user mention as an undirected communication link between the user tweeting and the mentioned user (including self-mentions but not retweets). We consider each user-user link as a single event; we count it once no matter how often two specific users interact. We extract 436M user-user links in total.

Attribute-Attribute Pairs We use our profile data to map each user-user link to an attribute-attribute pair; we separately count each pair of first names, last names, and locations. For example, the first-name pair (*henrik*, *fredrik*) occurs 181 times. Rather than using the raw count, we calculate the association between attributes a_1 and a_2 via their pointwise mutual information (PMI), following prior work in distributional clustering (Lin and Wu, 2009):

$$\text{PMI}(a_1, a_2) = \log \frac{\text{P}(a_1, a_2)}{\text{P}(a_1)\text{P}(a_2)}$$

PMI essentially normalizes the co-occurrence by what we would expect if the attributes were independently distributed. We smooth the PMI by adding a count of 0.5 to all co-occurrence events.

The most highly-associated name attributes reflect similarities in ethnicity and gender (Table 2). The most highly-ranked associates for locations are often nicknames and alternate/misspellings of those locations. For example, the locations *charm city*, *bmore*, *balto*, *westbaltimore*, *b a l t i m o r e*, *baltimoreee*, and *balitmore* each have the U.S. city of *baltimore* as their highest-PMI associate. We show how this can be used to help geolocate users (§7).

3 Attribute Clustering

Representation We first represent each target attribute as a feature vector, where each feature corresponds to another attribute of the same type as the target and each value gives the PMI between this attribute and the target (as in Table 2).[2] To help cluster the long-tail of infrequent attributes, we also include orthographic features. For first and last names, we have binary features for the last 2 characters in the string. For locations, we have binary features for (a) any ideographic characters in the string and (b) each token (with diacritics removed) in the string. We normalize the feature vectors to unit length.

Distributed K-Means Clustering Our approach to clustering follows Lin and Wu (2009) who used k-means to cluster tens of millions of phrases. We also use cosine similarity to compute the closest centroid (i.e., we use the *spherical* k-means clustering algorithm (Dhillon and Modha, 2001)). We keep track of the average cosine similarity between each vector and its nearest centroid; this average is guaranteed to increase at each iteration.

Like Lin and Wu (2009), we parallelize the algorithm using MapReduce. Each mapper finds the nearest centroids for a portion of the vectors, while also computing the partial sums of the vectors assigned to each centroid. The mappers emit the centroid IDs as keys and the partial sums as values. The Reducer aggregates the partial sums from each partition and re-normalizes each sum vector to unit length to obtain the new centroids. We also use an inverted index at each iteration that, for each input feature, lists which centroids each feature belongs to. Using this index greatly speeds up the centroid similarity computations.

Clustering Details We cluster with nine separate configurations: over first names, last names, and locations, and each with 50, 200, and 1000 cluster centroids (denoted C^{50}, C^{200}, and C^{1000}). Since k-

[2]We decided to restrict the features for a target to be attributes of the same type (e.g., we did not use *last name* associations for a *first name* target) because each attribute type conveys distinct information. For example, first names convey gender and age more than last names. By separately clustering representations using first names, last names, and locations, each clustering can capture its own distinct latent-class associations.

Cluster 463 (Serbian): pavlović, jovanovic, jovanović, stanković, srbija, marković, petrović, radovic, nenad, milenkovic, nikolic, sekulic, todorovic, stojanovic, petrovic, aleksic, ilic, markovic
Cluster 544 (Black South African): ngcobo, nkosi, dlamini, ndlovu, mkhize, mtshali, sithole, mathebula, mthembu, khumalo, ngwenya, shabangu, nxumalo, buthelezi, radebe, mabena, zwane, mbatha, sibiya
Cluster 449 (Turkish): şahin, çelik, öztürk, koç, çakır, karataş, aktaş, güngör, özkan, balcı, gümüş, akkaya, genç, sarı, yüksel, güneş, yiğit, yalçın, orhan, sağlam, güler, demirci, küçük, yavuz, bayrak, özcan, altun
Cluster 656 (Indonesian): utari, oktaviana, apriani, mustika, septiana, febrianti, kurniawati, indriani, nurjanah, septian, cahya, anggara, yuliani, purnamasari, sukma, wijayanti, pramesti, ningrum, yanti, wulansari

Table 3: Example C^{1000} last-name clusters

Cluster 56 [sim=0.497]: gregg, bryn, bret, stewart, lyndsay, howie, elyse, jacqui, becki, rhett, meaghan, kirstie, russ, jaclyn, zak, katey, seamus, brennan, fraser, kristie, stu, jaimie, kerri, heath, carley, griffin
Cluster 104 [sim=0.442]: stephon, devonte, deion, demarcus, janae, tyree, jarvis, donte, dewayne, javon, destinee, tray, janay, tyrell, jamar, iesha, chyna, jaylen, darion, lamont, marquise, domonique, alexus
Cluster 132 [sim=0.292]: moustafa, omnya, mennatallah, إسلام, shorouk, ragab, لؤي, radwa, moemen, mohab, hazem, yehia, حرية, اسراء, mennah, مصري, abdelrahman, مصطفي, حزب, تامر, nermeen, hebatallah
...

Table 4: C^{200} soft clustering for first name *yasmeen*

means is not guaranteed to reach a global optimum, we use ten different random initializations for each configuration, and select the one with the highest average similarity after 20 iterations. We run this one for an additional 30 iterations and take the output as our final set of centroids for that configuration.

The resulting clusters provide data that could help classify hidden properties of social media users. For example, Table 3 shows that last names often cluster by ethnicity, even at the sub-national level (e.g. Zulu tribe surnames *nkosi*, *dlamini*, *mathebula*, etc.). Note the Serbian names include two entries that are not last names: *srbija*, the Serbian word for *Serbia*, and *nenad*, a common Serbian first name.

Soft Clustering Rather than assigning each attribute to its single highest-similarity cluster, we can assign each vector to its N most similar clusters. These soft-cluster assignments often reflect different social groups where a name or location is used. For example, the name *yasmeen* is similar to both common American names (Cluster 56), African American names (Cluster 104), and Arabic names (Cluster 132) (Table 4). As another example, the C^{1000} assignments for the location *trujillo* comprise separate clusters containing towns and cities in Peru, Venezuela, Colombia, etc., reflecting the various places in the Latin world with this name. In general, the soft cluster assignment is a low-dimensional representation of each of our attributes. Although it can be interpretable to humans, it need not be in order to be useful to a classifier.

4 Classification with Cluster Features

Our motivating problem is to classify users for hidden properties such as their gender, location, race, ethnicity, and language. We adopt a discriminative solution. We encode the relevant data for each instance in a feature vector and train a (linear) support vector machine classifier (Cortes and Vapnik, 1995). SVMs represent the state-of-the-art on many NLP classification tasks, but other classifiers could also be used. For multi-class classification, we use a one-versus-all strategy, a competitive approach on most multi-class problems (Rifkin and Klautau, 2004).

The input to our system is one or more observed user attributes (e.g. name and location fields from a user profile). We now describe how features are created from these attributes in both state-of-the-art systems and via our new cluster data.

Token Features (*Tok*) are binary features that indicate the presence of a specific attribute (e.g., *first-name=bob*). Burger et al. (2011) and Bergsma et al. (2012) used *Tok* features to encode user profile features. For multi-token fields (e.g. location), our *Tok* features also indicate the specific position of each token (e.g., loc_1=*são,* loc_2=*paulo,* loc_N=*brasil*).

Character N-gram Features (*Ngm*) give the count of all character n-grams of length 1-to-4 in the input. *Ngm* features have been used in user classification (Burger et al., 2011) and represent the state-

of-the-art in detecting name ethnicity (Bhargava and Kondrak, 2010). We add special begin/end characters to the attributes to mark the prefix and suffix positions. We also use a smoothed log-count; we found this to be most effective in preliminary work.

Cluster Features (*Clus*) indicate the soft-cluster memberships of the attributes. We have features for the top-2, 5, and 20 most similar clusters in the C^{50}, C^{200}, and C^{1000} clusterings, respectively. Like Lin and Wu (2009), we "side-step the matter of choosing the optimal value k in k-means" by using features from clusterings at different granularities. Our feature dimensions correspond to cluster IDs; feature values give the similarity to the cluster centroid. Other strategies (e.g. hard clustering, binary features) were less effective in preliminary work.

5 Classification Experiments

5.1 Methodology

Our main objective is to assess the value of using cluster features (*Clus*). We add these features to classifiers using *Tok+Ngm* features, which represents the current state-of-the-art. We compare these feature settings on both Twitter tasks (§5.2) and tasks not related to social-media (§5.3). For each task, we randomly divide the gold standard data into 50% train, 25% development and 25% test, unless otherwise noted. As noted above, the gold-standard datasets for all of our experiments are available for download. We train our SVM classifiers using the LIBLINEAR package (Fan et al., 2008). We optimize the classifier's regularization parameter on development data, and report our final results on the held-out test examples. We report *accuracy*: the proportion of test examples classified correctly. For comparison, we report the accuracy of a majority-class baseline on each task (*Base*).

Classifying hidden properties of social media users is challenging (Table 5). Pennacchiotti and Popescu (2011) even conclude that "profile fields do not contain enough good-quality information to be directly used for user classification." To provide insight into the difficulty of the tasks, we had two humans annotate 120 examples from each of the test sets, and we average their results to give a "*Human*" performance number. The two humans are experts in this domain and have very wide knowledge of global names and locations.

Country: 53 possible countries		
United States	courtland dante	cali baby
United States	tinas twin	on the court
Brazil	thamires gomez	macapá ap
Denmark	marte clason	NONE
Lang. ID: 9 confusable languages		
Bulgarian	valentina getova	NONE
Russian	borisenko yana	edinburgh
Bulgarian	NONE	blagoevgrad
Ukrainian	andriy kupyna	ternopil
Farsi	kambiz barahouei	NONE
Urdu	musadiq sanwal	jammu
Ethnicity: 13 European ethnicities		
German	dennis hustadt	
Dutch	bernhard hofstede	
French	david coste	
Swedish	mattias bjarsmyr	
Portuguese	helder costa	
Race: black or white		
black	kerry swain	
black	darrell foskey	
white	ty j larocca	
black	james n jones	
white	sean p farrell	

Table 5: Examples of class (left) and input (names, locations) for some of our evaluation tasks.

5.2 Twitter Applications

Country A number of recent papers have considered the task of predicting the geolocation of users, using both user content (Cheng et al., 2010; Eisenstein et al., 2010; Hecht et al., 2011; Wing and Baldridge, 2011; Roller et al., 2012) and social network (Backstrom et al., 2010; Sadilek et al., 2012).

Here, we first predict user location at the level of the user's location *country*. To our knowledge, we are the first to exploit user locations *and* names for this prediction. For this task, we obtain gold data from the portion of Twitter users who have GPS enabled (geocoded tweets). We were able to obtain a very large number of gold instances for this task, so selected only 10K for testing, 10K for development, and retained the remaining 782K for training.

Language ID Identifying the language of users is an important prerequisite for building language-specific social media resources (Tromp and Pech-

enizkiy, 2011; Carter et al., 2013). Bergsma et al. (2012) recently released a corpus of tweets marked for one of nine languages grouped into three confusable character sets: Arabic, Farsi, and Urdu tweets written in Arabic characters; Hindi, Nepali, and Marathi written in Devanagari, and Russian, Bulgarian, and Ukrainian written in Cyrillic. The tweets were marked for language by native speakers via Amazon Mechanical Turk. We again discard the tweet content and extract each user's first name, last name, and user location as our input data, while taking the annotated language as the class label.

Gender We predict whether a Twitter user is male or female using data from Burger et al. (2011). This data was created by linking Twitter users to structured profile pages on other websites where users must select their gender. Unlike prior systems using this data (Burger et al., 2011; Van Durme, 2012), we make the predictions using only user names.

5.3 Other Applications

Origin Knowing the origin of a name can improve its automatic pronunciation (Llitjos and Black, 2001) and transliteration (Bhargava and Kondrak, 2010). We evaluate our cluster data on name-origin prediction using a corpus of names marked as either Indian or non-Indian by Bhargava and Kondrak (2010). Since names in this corpus are not marked for entity type, we include separate cluster features from both our first and last name clusters.

Ethnicity We also evaluate on name-origin data from Konstantopoulos (2007). This data derives from lists of football players on European national teams; it marks each name (with diacritics removed) as arising from one of 13 European languages. Following prior work, we test in two settings: (1) using last names only, and (2) using first and last names.

Race We also evaluate our ability to identify ethnic groups at a sub-national level. To obtain data for this task, we mined the publicly-available arrest records on `mugshots.com` for the U.S. state of New Jersey (a small but diverse and densely-populated area). Over 99% of users were listed as either *black* or *white*, and we structure the task as a binary classification problem between these two classes. We predict the race of each person based purely on their name; this contrasts with prior work in social media which looked at identifying African Americans on the basis of their Twitter *content* (Eisenstein et al., 2011; Pennacchiotti and Popescu, 2011).

6 Classification Results

Table 6 gives the results on each task. The system incorporating our novel *Clus* features consistently improves over the *Ngm+Tok* system; all differences between *All* and *Ngm+Tok* are significant (McNemar's, $p<0.01$). The relative reduction in error from adding *Clus* features ranges between 7% and 51%. The *All* system including *Clus* features also exceeds human performance on all studied tasks.

On **Country**, the U.S. is the majority class, occurring in 42.5% of cases.[3] It is impressive that *All* so significantly exceeds *Tok+Ngm* (86.7% vs. 84.8%); with 782K training examples, we did not expect such room for improvement. Both names and locations play an important role: *All* achieves 66% using names alone and 70% with only location. On the subset of data where all three attributes are non-empty, the full system achieves 93% accuracy.

Both feature classes are likewise important for **Lang. ID**; *All* achieves 67% with only first+last names, 72% with just locations, but 83% with both.

Our smallest improvement is on **Gender**. This task is easier (with higher human/system accuracy) and has plenty of training data (more data *per class* than any other task); there is thus less room to improve. Looking at the feature weights, the strongest-weighted *female* cluster apparently captures a subcommunity of Justin Bieber fans (showing loyalty with "first names" *jbieber*, *belieb*, *biebz*, *beliebing*, *jbiebs*, etc.). Just because a first name like *madison* has a high similarity to this cluster does not imply girls named *Madison* are Justin Bieber fans; it simply means that Madisons have similar names to the *friends* of Justin Bieber fans (who tend to be girls). Also, note that while the majority of the 34K users in our training data are assigned this cluster somewhere in their soft clustering, only 6 would be assigned this

[3] We tried other baselines: e.g., we predict countries if they are substrings of the location (otherwise predicting U.S.); and we predict countries if they often occur as a string following the given location in our profile data (e.g., we predict *Spain* for *Madrid* since *Madrid, Spain* is common). Variations on these approaches consistently performed between 48% and 56%.

Task	Input	Num. Train	Num. Class	*Base*	*Human*	*Tok*	*Ngm*	*Clus*	*Tok+ Ngm*	*All*	Δ
Country	first+last+loc	781920	53	42.5	71.7	83.0	84.5	80.2	84.8	**86.7**	12.5
Lang. ID	first+last+loc	2492	9	27.0	74.2	74.6	80.6	71.1	80.4	**82.7**	11.7
Gender	first+last	33805	2	52.4	88.3	85.3	88.6	79.5	89.5	**90.2**	6.7
Origin	entity name	500	2	52.4	80.4	-	75.6	81.2	75.6	**88.0**	50.8
Ethnicity	last	6026	13	20.8	47.9	-	54.6	48.5	54.6	**62.4**	17.2
Ethnicity	first+last	7457	13	21.2	53.3	67.6	77.5	73.6	78.4	**81.3**	13.4
Race	first+last	7977	2	54.7	71.4	80.4	81.6	84.6	82.4	**84.6**	12.5

Table 6: Task details and accuracy (%) for attribute-based classification tasks. Δ = relative error reduction (%) of *All* (*Tok+Ngm+Clus*) over *Ngm+Tok*. *All* always exceeds both *Tok+Ngm* and the human performance.

cluster in a *hard* clustering. This clearly illustrates the value of the soft clustering representation.

Note the *All* system performed between 83% and 90% on each Twitter task. This level of performance strongly refutes the prevailing notion that Twitter profile information is useless in general (Pennacchiotti and Popescu, 2011) and especially for geolocation (Cheng et al., 2010; Hecht et al., 2011).

We now move to applications beyond social media. Bhargava and Kondrak (2010) have the current state-of-the-art on **Origin** and **Ethnicity** based on an SVM using character-n-gram features; we reimplemented this as *Ngm*. We obtain a huge improvement over their work using *Clus*, especially on **Origin** where we reduce error by >50%.[4] This improvement can partly be attributed to the small amount of training data; with fewer parameters to learn, *Clus* learns more from limited data than *Ngm*. We likewise see large improvements over the state-of-the-art on **Ethnicity**, on both last name and full name settings.

Finally, *Clus* features also significantly improve accuracy on the new **Race** task. Our cluster data can therefore help to classify names into sub-national groups, and could potentially be used to infer other interesting communities such as castes in India and religious divisions in many countries.

In general, the relative value of our cluster models varies with the amount of training data; we see huge gains on the smaller **Origin** data but smaller gains on the large **Gender** set. Figure 1 shows how performance of *Clus* and *Ngm* varies with training data on **Race**. Again, *Clus* is especially helpful with less

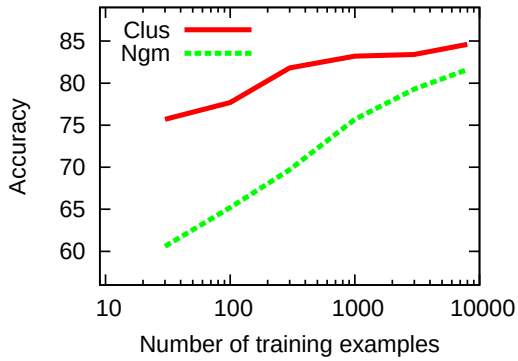

Figure 1: Learning curve on Race: *Clus* perform as well with 30 training examples as *Ngm* features do with 1000.

data; thousands of training examples are needed for *Ngm* to rival the performance of *Clus* using only a handful. Since labeled data is generally expensive to obtain or in short supply, our method for exploiting unlabeled Twitter data can both save money and improve top-end performance.

7 Geolocation by Association

There is a tradition in computational linguistics of grouping words both by the similarity of their context vectors (Hindle, 1990; Pereira et al., 1993; Lin, 1998) and directly by their statistical association in text (Church and Hanks, 1990; Brown et al., 1992). While the previous sections explored clusters built by vector similarity, we now explore a direct application of our attribute association data (§2).

We wish to use this data to improve an existing Twitter geolocation system based on user profile locations. The system operates as follows: 1) normal-

[4]Note *Tok* is not used here because the input is a single token and training and test splits have distinct instances.

ize user-provided locations using a set of regular expressions (e.g. remove extra spacing, punctuation); 2) look up the normalized location in an *alias list*; 3) if found, map the alias to a unique string (target location), corresponding to a structured location object that includes geo-coordinates.

The alias list we are currently using is based on extensive work in hand-writing aliases for the most popular Twitter locations. For example, the current aliases for *Nashville, Tennessee* include *nashville*, *nashville tn*, *music city*, etc. Our objective is to improve on this human-designed list by automatically generating aliases using our association data.

Aliases by Association For each target, we propose new aliases from the target's top-PMI associates (§2). To become an alias, the PMI between the alias and target must be above a threshold, the alias must occur more than a fixed number of times in our profile data, the alias must be within the top-N_1 associates of the target, and the target must be within the top-N_2 associates of the alias. We merge our automatic aliases with the manually-written aliases. The new aliases for *Nashville, Tennessee* include *east nashville*, *nashville tenn*, *music city usa*, *nashvegas*, *cashville tn*, etc.

Experiments To evaluate the geolocation system, we use tweets from users with GPS enabled (§5.2). For each tweet, we resolve the location using the system and compare to the gold coordinates. The system can skip a location if it does not match the alias list; more than half of the locations are skipped, which is consistent with prior work (Hecht et al., 2011). We evaluate the alias lists using two measures: (1) its coverage: the percentage of locations it resolves, and (2) its precision: of the ones resolved, the percentage that are correct. We define a *correct* resolution to be one where the resolved coordinates are within 50 miles of the gold coordinates.

We use 56K gold tweets to tune the parameters of our automatic alias-generator, trading off coverage and precision. We tune such that the system using these aliases obtains the highest possible coverage, while being at least as precise as the baseline system. We then evaluate both the baseline set of aliases and our new set on 56K held-out examples.

Results On held-out test data, the geolocation system using baseline aliases has a coverage of 38.7% and a precision of 59.5%. Meanwhile, the system using the new aliases has a coverage of 44.6% and a precision of 59.4%. With virtually the same precision, the new aliases are thus able to resolve 15% more users. This provides an immediate benefit to our existing Twitter research efforts.

Note that our alias lists can be viewed as *clusters* of locations. In ongoing work, we are exploring techniques based on discriminative learning to infer alias lists using not only *Clus* information but also *Ngm* and *Tok* features as in the previous sections.

8 Related Work

In both real-world and online social networks, "people socialize with people who are like them in terms of gender, sexual orientation, age, race, education, and religion" (Jernigan and Mistree, 2009). Social media research has exploited this for two main purposes: (1) to predict friendships based on user properties, and (2) to predict user properties based on friendships. Friendship prediction systems (e.g. Facebook's *friend suggestion tool*) use features such as whether both people are computer science majors (Taskar et al., 2003) or whether both are at the same location (Crandall et al., 2010; Sadilek et al., 2012). The inverse problem has been explored in the prediction of a user's location given the location of their peers (Backstrom et al., 2010; Cho et al., 2011; Sadilek et al., 2012). Jernigan and Mistree (2009) predict a user's sexuality based on the sexuality of their Facebook friends, while Garera and Yarowsky (2009) predict a user's gender partly based on the gender of their conversational partner. Jha and Elhadad (2010) predict the cancer stage of users of an online cancer discussion board; they derive complementary information for prediction from both the text a user generates and the cancer stage of the people that a user interacts with.

The idea of clustering data in order to provide features for supervised systems has been successfully explored in a range of NLP tasks, including named-entity-recognition (Miller et al., 2004; Lin and Wu, 2009; Ratinov and Roth, 2009), syntactic chunking (Turian et al., 2010), and dependency parsing (Koo et al., 2008; Täckström et al., 2012). In each case,

the clusters are derived from the distribution of the words or phrases in text, not from their communication pattern. It would be interesting to see whether prior distributional clusters can be combined with our communication-based clusters to achieve even better performance. Indeed, there is evidence that features derived from text can improve the prediction of name ethnicity (Pervouchine et al., 2010).

There has been an explosion of work in recent years in predicting user properties in social networks. Aside from the work mentioned above that analyzes a user's social network, a large amount of work has focused on inferring user properties based on the content they generate (e.g. Burger and Henderson (2006), Schler et al. (2006), Rao et al. (2010), Mukherjee and Liu (2010), Pennacchiotti and Popescu (2011), Burger et al. (2011), Van Durme (2012)).

9 Conclusion and Future Work

We presented a highly effective and readily replicable algorithm for generating language resources from Twitter communication patterns. We clustered user attributes based on both the communication of users with those attributes as well as substring similarity. Systems using our clusters significantly outperform state-of-the-art algorithms on each of the tasks investigated, and exceed human performance on each task as well. The power and versatility of our clusters is exemplified by the fact we reduce error by a larger margin on each of the non-Twitter tasks than on any Twitter task itself.

Twitter provides a remarkably large sample and effectively a partial census of much of the world's population, with associated metadata, descriptive content and sentiment information. Our ability to accurately assign numerous often unspecified properties such as race, gender, language and ethnicity to such a large user sample substantially increases the sociological insights and correlations one can derive from such data.

References

Lars Backstrom, Eric Sun, and Cameron Marlow. 2010. Find me if you can: improving geographical prediction with social and spatial proximity. In *Proc. WWW*, pages 61–70.

Shane Bergsma, Paul McNamee, Mossaab Bagdouri, Clayton Fink, and Theresa Wilson. 2012. Language identification for creating language-specific Twitter collections. In *Proceedings of the Second Workshop on Language in Social Media*, pages 65–74.

Aditya Bhargava and Grzegorz Kondrak. 2010. Language identification of names with SVMs. In *Proc. HLT-NAACL*, pages 693–696.

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

John D. Burger and John C. Henderson. 2006. An exploration of observable features related to blogger age. In *Proc. AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*, pages 15–20.

John D. Burger, John Henderson, George Kim, and Guido Zarrella. 2011. Discriminating gender on Twitter. In *Proc. EMNLP*, pages 1301–1309.

Simon Carter, Wouter Weerkamp, and Manos Tsagkias. 2013. Microblog Language Identification: Overcoming the Limitations of Short, Unedited and Idiomatic Text. *Language Resources and Evaluation Journal*. (forthcoming).

Zhiyuan Cheng, James Caverlee, and Kyumin Lee. 2010. You are where you tweet: a content-based approach to geo-locating Twitter users. In *Proc. CIKM*, pages 759–768.

Eunjoon Cho, Seth A. Myers, and Jure Leskovec. 2011. Friendship and mobility: user movement in location-based social networks. In *Proc. KDD*, pages 1082–1090.

Kenneth W. Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Mach. Learn.*, 20(3):273–297.

David J. Crandall, Lars Backstrom, Dan Cosley, Siddharth Suri, Daniel Huttenlocher, and Jon Kleinberg. 2010. Inferring social ties from geographic coincidences. *Proceedings of the National Academy of Sciences*, 107(52):22436–22441.

Jeffrey Dean and Sanjay Ghemawat. 2008. MapReduce: simplified data processing on large clusters. *Commun. ACM*, 51(1):107–113.

Inderjit S. Dhillon and Dharmendra S. Modha. 2001. Concept decompositions for large sparse text data using clustering. *Mach. Learn.*, 42(1-2):143–175.

Jacob Eisenstein, Brendan O'Connor, Noah A. Smith, and Eric P. Xing. 2010. A latent variable model for geographic lexical variation. In *Proc. EMNLP*, pages 1277–1287.

Jacob Eisenstein, Noah A. Smith, and Eric P. Xing. 2011. Discovering sociolinguistic associations with structured sparsity. In *Proc. ACL*, pages 1365–1374.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A library for large linear classification. *J. Mach. Learn. Res.*, 9:1871–1874.

Nikesh Garera and David Yarowsky. 2009. Modeling latent biographic attributes in conversational genres. In *Proc. ACL-IJCNLP*, pages 710–718.

Brent Hecht, Lichan Hong, Bongwon Suh, and Ed H. Chi. 2011. Tweets from Justin Bieber's heart: the dynamics of the location field in user profiles. In *Proc. CHI*, pages 237–246.

Donald Hindle. 1990. Noun classification from predicate-argument structures. In *Proc. ACL*, pages 268–275.

Carter Jernigan and Behram F. T. Mistree. 2009. Gaydar: Facebook friendships expose sexual orientation. *First Monday*, 14(10). [Online].

Mukund Jha and Noemie Elhadad. 2010. Cancer stage prediction based on patient online discourse. In *Proc. 2010 Workshop on Biomedical Natural Language Processing*, pages 64–71.

Stasinos Konstantopoulos. 2007. What's in a name? In *Proc. Computational Phonology Workshop, RANLP*.

Terry Koo, Xavier Carreras, and Michael Collins. 2008. Simple semi-supervised dependency parsing. In *Proc. ACL-08: HLT*, pages 595–603.

Dekang Lin and Xiaoyun Wu. 2009. Phrase clustering for discriminative learning. In *Proc. ACL-IJCNLP*, pages 1030—1038.

Dekang Lin. 1998. Automatic retrieval and clustering of similar words. In *Proc. Coling-ACL*, pages 768–774.

Ariadna Font Llitjos and Alan W. Black. 2001. Knowledge of language origin improves pronunciation accuracy of proper names. In *Proceedings of EuroSpeech-01*, pages 1919–1922.

Scott Miller, Jethran Guinness, and Alex Zamanian. 2004. Name tagging with word clusters and discriminative training. In *Proc. HLT-NAACL*, pages 337–342.

Arjun Mukherjee and Bing Liu. 2010. Improving gender classification of blog authors. In *Proc. EMNLP*, pages 207–217.

Brendan O'Connor, Ramnath Balasubramanyan, Bryan R. Routledge, and Noah A. Smith. 2010. From tweets to polls: Linking text sentiment to public opinion time series. In *Proc. ICWSM*, pages 122–129.

Michael Paul and Mark Dredze. 2011. You are what you tweet: Analyzing Twitter for public health. In *Proc. ICWSM*, pages 265–272.

Marco Pennacchiotti and Ana-Maria Popescu. 2011. A machine learning approach to Twitter user classification. In *Proc. ICWSM*, pages 281–288.

Fernando Pereira, Naftali Tishby, and Lillian Lee. 1993. Distributional clustering of English words. In *Proc. ACL*, pages 183–190.

Vladimir Pervouchine, Min Zhang, Ming Liu, and Haizhou Li. 2010. Improving name origin recognition with context features and unlabelled data. In *Coling 2010: Posters*, pages 972–978.

Delip Rao, David Yarowsky, Abhishek Shreevats, and Manaswi Gupta. 2010. Classifying latent user attributes in Twitter. In *Proc. International Workshop on Search and Mining User-Generated Contents*, pages 37–44.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proc. CoNLL*, pages 147–155.

Ryan Rifkin and Aldebaro Klautau. 2004. In defense of one-vs-all classification. *J. Mach. Learn. Res.*, 5:101–141.

Stephen Roller, Michael Speriosu, Sarat Rallapalli, Benjamin Wing, and Jason Baldridge. 2012. Supervised text-based geolocation using language models on an adaptive grid. In *Proc. EMNLP-CoNLL*, pages 1500–1510.

Adam Sadilek, Henry Kautz, and Jeffrey P. Bigham. 2012. Finding your friends and following them to where you are. In *Proc. WSDM*, pages 723–732.

Jonathan Schler, Moshe Koppel, Shlomo Argamon, and James W. Pennebaker. 2006. Effects of age and gender on blogging. In *Proc. AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*, pages 199–205.

Oscar Täckström, Ryan McDonald, and Jakob Uszkoreit. 2012. Cross-lingual word clusters for direct transfer of linguistic structure. In *Proc. NAACL-HLT*, pages 477–487.

Ben Taskar, Ming-Fai Wong, Pieter Abbeel, and Daphne Koller. 2003. Link prediction in relational data. In *Proc. NIPS*, volume 15.

Erik Tromp and Mykola Pechenizkiy. 2011. Graph-based n-gram language identication on short texts. In *Proc. 20th Machine Learning conference of Belgium and The Netherlands*, pages 27–34.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: A simple and general method for semi-supervised learning. In *Proc. ACL*, pages 384–394.

Benjamin Van Durme. 2012. Streaming analysis of discourse participants. In *Proc. EMNLP-CoNLL*, pages 48–58.

Benjamin Wing and Jason Baldridge. 2011. Simple supervised document geolocation with geodesic grids. In *Proc. ACL*, pages 955–964.