Using Conceptual Class Attributes to Characterize Social Media Users

Shane Bergsma and Benjamin Van Durme
Department of Computer Science and Human Language Technology Center of Excellence
Johns Hopkins University
Baltimore, MD 21218, USA

Abstract

We describe a novel approach for automatically predicting the hidden demographic properties of social media users. Building on prior work in common-sense knowledge acquisition from third-person text, we first learn the distinguishing attributes of certain classes of people. For example, we learn that people in the *Female* class tend to have *maiden names* and *engagement rings*. We then show that this knowledge can be used in the analysis of first-person communication; knowledge of distinguishing attributes allows us to both classify users and to bootstrap new training examples. Our novel approach enables substantial improvements on the widely-studied task of user gender prediction, obtaining a 20% relative error reduction over the current state-of-the-art.

1 Introduction

There has been growing interest in *characterizing* social media users based on the content they generate; that is, automatically labeling users with demographic categories such as age and gender (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). Automatic user characterization has applications in targeted advertising and personalization, and could also lead to finer-grained assessment of public opinion (O'Connor et al., 2010) and health (Paul and Dredze, 2011).

Consider the following tweet and suppose we wish to predict the user's gender:

> *Dirac was one of* ***my boyhood heroes****. I'm glad I met him once. RT Paul Dirac image by artist Eric Handy:* `http:...`

State-of-the-art approaches cast this problem as a classification task and train classifiers using supervised learning (Section 2). The features of the classifier are indicators of specific words in the user-generated text. While a human would assume that someone with *boyhood heroes* is **male**, a standard classifier has no way of exploiting such knowledge unless the phrase occurs in training data. We present an algorithm that improves user characterization by collecting and exploiting such common-sense knowledge.

Our work is inspired by algorithms that processes large text corpora in order to discover the attributes of semantic classes, e.g. (Berland and Charniak, 1999; Schubert, 2002; Almuhareb and Poesio, 2004; Tokunaga et al., 2005; Girju et al., 2006; Paşca and Van Durme, 2008; Alfonseca et al., 2010). We learn the distinguishing attributes of different demographic groups (Section 3), and then automatically assign users to these groups whenever they refer to a distinguishing attribute in their writings (Section 4). Our approach obviates the need for expensive annotation efforts, and allows us to rapidly bootstrap training data for new classification tasks.

We validate our approach by advancing the state-of-the-art on the most well-studied user classification task: predicting user gender (Section 5). Our bootstrapped system, trained purely from automatically-annotated Twitter data, significantly reduces error over a state-of-the-art system trained on thousands of gold-standard training examples.

2 Supervised User Characterization

The current state-of-the-art in user characterization is to use supervised classifiers trained on annotated data. For each instance to be classified, the output is a decision about a distinct demographic property, such as *Male/Female* or *Over/Under-18*. A variety of classification algorithms have been employed, including SVMs (Rao et al., 2010), de-

Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pages 710–720,
Sofia, Bulgaria, August 4-9 2013. ©2013 Association for Computational Linguistics

cision trees (Pennacchiotti and Popescu, 2011), logistic regression (Van Durme, 2012), and the Winnow algorithm (Burger et al., 2011).

Content Features: *BoW* Prior classifiers use a set of features encoding the presence of specific words in the user-generated text. We call these features *BoW* features as they encode the standard Bag-of-Words representation which has been highly effective in text categorization and information retrieval (Sebastiani, 2002).

User-Profile Features: *Usr* Some researchers have explored features for user-profile meta-information in addition to user content. This may include the user's communication behavior and network of contacts (Rao et al., 2010), their full name (Burger et al., 2011) and whether they provide a profile picture (Pennacchiotti and Popescu, 2011). We focus on the case where we only have access to the user's screen-name (a.k.a. username). Using a combination of content and username features "represents a use case common to many different social media sites, such as chat rooms and news article comment streams" (Burger et al., 2011). We refer to features derived from a username as *Usr* features in our experiments.

3 Learning Class Attributes

We aim to improve the automated classification of users into various demographic categories by learning and applying the distinguishing attributes of those categories, e.g. that *males* have *boyhood heroes*. Our approach builds on lexical-semantic research on the topic of *class-attribute extraction*. In this research, the objective is to discover various *attributes* or *parts* of classes of entities. For example, Berland and Charniak (1999) learn that the class *car* has parts such as *headlight*, *windshield*, *dashboard*, etc. Berland and Charniak extract these attributes by mining a corpus for fillers of patterns such as '*car's X*' or '*X of a car*'. Note their patterns explicitly include the class itself (*car*). Another approach is to use patterns that are based on *instances* (i.e. hyponyms or sub-classes) of the class. For example, Paşca and Van Durme (2007) learn the attributes of the class *car* via patterns involving instances of cars, e.g. *Chevrolet Corvette's X* and *X of a Honda Civic*. For these approaches, lists of instances are typically collected from publicly-available resources such as WordNet or Wikipedia (Paşca and Van Durme, 2007; Van Durme et al., 2008), acquired automatically from corpora (Paşca and Van Durme, 2008; Alfonseca et al., 2010), or simply specified by hand (Schubert, 2002).

Creation of Instance Lists We use an instance-based approach; our instances are derived from collections of common nouns that are associated with roles and occupations of people. For the gender task that we study in our experiments, we acquire class instances by filtering the dataset of nouns and their genders created by Bergsma and Lin (2006). This dataset indicates how often a noun is referenced by a male, female, neutral or plural pronoun. We extract prevalent common nouns for males and females by selecting only those nouns that (a) occur more than 200 times in the dataset, (b) mostly occur with male or female pronouns, and (c) occur as lower-case more often than upper-case in a web-scale N-gram corpus (Lin et al., 2010). We then classify a noun as **Male** (resp. **Female**) if the noun is indicated to occur with male (resp. female) pronouns at least 85% of the time. Since the gender data is noisy, we also quickly pruned by hand any instances that were malformed or obviously incorrectly assigned by our automatic process. This results in 652 instances in total. Table 1 provides some examples.

Male: bouncer, altar boy, army officer, dictator, assailant, cameraman, drifter, chauffeur, bad guy

Female: young lady, lesbian, ballerina, waitress, granny, chairwoman, heiress, soprano, socialite

Table 1: Example instances used for extraction of class attributes for the gender classification task

Attribute Extraction We next collect and rank attributes for each class. We first look for fillers of attribute-patterns involving each of the instances. Let I represent an instance of one of our classes. We find fillers of the single high-precision pattern:

$$\underbrace{\{\text{word=}I\text{,tag=NN}\}}_{\textit{instance}}\ \underbrace{\{\text{word='s}\}}_{\textit{'s}}\ \underbrace{[\{\text{word=.}*\}^*\ \{\text{tag=}N.*\}]}_{\textit{attribute}}$$

(E.g. *dictator* 's [*former mistress*]). The expression "tag=*NN*" means that I must be tagged as a noun. The expression in square brackets is the filler, i.e. the extracted attribute, A. The notation "$\{$word=.*$\}$* tag=*N.**" means that A can be any sequence of tokens ending in a noun. We use an equivalent pattern when I is multi-token. The output of this process is a set of (I,A) pairs.

In attribute extraction, typically one must choose between the precise results of rich patterns (involving punctuation and parts-of-speech) applied to small corpora (Berland and Charniak, 1999) and the high-coverage results of superficial patterns applied to web-scale data, e.g. via the Google API (Almuhareb and Poesio, 2004). We obtain the best of both worlds by matching our precise pattern against a version of the Google N-gram Corpus that includes the part-of-speech tag distributions for every N-gram (Lin et al., 2010). We found that applying this pattern to web-scale data is effective in extracting useful attributes. We acquired around 20,000 attributes in total.

Finding Distinguishing Attributes Unlike prior work, we aim to find *distinguishing* properties of each class; that is, the kinds of properties that uniquely distinguish a particular category. Prior work has mostly focused on finding "relevant" attributes (Alfonseca et al., 2010) or "correct" parts (Berland and Charniak, 1999). A *leg* is a relevant and correct part of both a male and a female (and many other living and inanimate objects), but it does not help us distinguish males from females in social media. We therefore rank our attributes for each class by their strength of association with instances of that specific class.[1]

To calculate the association, we first disregard the count of each (I,A) pair and consider each unique pair to be a single probabilistic event. We then convert the (I,A) pairs to corresponding (C,A) pairs by replacing I with the corresponding class, C. We then calculate the pointwise mutual information (Church and Hanks, 1990) between each C and A over the set of events:

$$\text{PMI}(C,A) = \log \frac{p(C,A)}{p(C)p(A)} \qquad (1)$$

If the PMI>0, the observed probability of a class and attribute co-occurring is greater than the probability of co-occurrence that we would expect if C and A were independently distributed. For each class, we rank the attributes by their PMI scores.

Filtering Attributes We experimented with two different methods to select a final set of distinguishing attributes for each class: (1) we used a threshold to select the top-ranked attributes for each class, and (2) we manually filtered the attributes. For the gender classification task, we manually filtered the entire set of attributes to select around 1000 attributes that were judged to be discriminative (two thirds of which are female). This filtering took one annotator only a few hours to complete. Because this process was so trivial, we did not invest in developing annotation guidelines or measuring inter-annotator agreement. We make these filter attributes available online as an attachment to this article, available through the ACL Anthology.

Ultimately, we discovered that manual filtering was necessary to avoid certain pathological cases in our Twitter data. For example, our PMI scoring finds *homepage* to be strongly associated with males. In our gold-standard gender data (Section 5), however, *every* user has a homepage [by dataset construction]; we might therefore incorrectly classify every user as **Male**. We agree with Richardson et al. (1998) that "automatic procedures ... provide the only credible prospect for acquiring world knowledge on the scale needed to support common-sense reasoning" but "hand vetting" might be needed to ensure "accuracy and consistency in production level systems." Since our approach requires manual involvement in the filtering of the attribute list, one might argue that one should simply manually enumerate the most relevant attributes directly. However, the manual generation of conceptual features by a single researcher results in substantial variability both across and within participants (McRae et al., 2005). Psychologists therefore generate such lists by pooling the responses across many participants: future work may compare our "automatically generate, manually prune" approach to soliciting attributes via crowdsourcing.[2]

Table 2 gives examples of our extracted at-

[1] Reisinger and Paşca (2009) considered the related problem of finding the most appropriate *class* for each *attribute*; they take an existing ontology of concepts (WordNet) as a class hierarchy and use a Bayesian approach to decide "the correct level of abstraction for each attribute."

[2] One can also view the work of manually filtering attributes as a kind of "feature labeling." There is evidence from Zaidan et al. (2007) that a few hours of feature labeling can be more productive than annotating new training examples. In fact, since Zaidan et al. (2007) label features at the token level (e.g., in our case one would highlight "handbag" in a given tweet), while we label features at the type level (e.g., deciding whether to mark the word "handbag" as feminine in general), our process is likely even more efficient. Future work may also wish to consider this connection to so-called "annotator rationales" more deeply.

Male: wife, widow, wives, ex-girlfriend, erection, testicles, wet dream, bride, buddies, ex-wife, first-wife, penis, death sentence, manhood

Female: vagina, womb, maiden name, dresses, clitoris, wedding dress, uterus, shawl, necklace, ex-husband, ex-boyfriend, dowry, nightgown

Table 2: Example attributes for gender classes, in descending order of class-association score

tributes. Our approach captures many *multi-token* attributes; these are often distinguishing even though the head noun is ambiguous (e.g. *name* is ambiguous, *maiden name* is not). Our attributes also go beyond the traditional meronyms that were the target of earlier work. As we discuss further in Related Work (Section 7), previous researchers have worried about a proper definition of *parts* or *attributes* and relied on human judgments for evaluation (Berland and Charniak, 1999; Girju et al., 2006; Van Durme et al., 2008). For us, whether a property such as *dowry* should be considered an "attribute" of the class **Female** is immaterial; we echo Almuhareb and Poesio (2004) who (on a different task) noted that "while the notion of 'attribute' is not completely clear... our results suggest that trying to identify attributes is beneficial."

4 Applying Class Attributes

To classify users using the extracted attributes, we look for cases where users refer to such attributes in their first-person writings. We performed a preliminary analysis of a two-week sample of tweets from the TREC Tweets2011 Corpus.[3] We found that users most often reveal their attributes in the possessive construction, "my *X*" where *X* is an attribute, quality or event that they possess (in a linguistic sense). For example, we found over 1000 tweets with the phrase "my wife." In contrast, "I have a wife" occurs only 5 times.[4]

We therefore assign a user to a demographic category as follows: We first part-of-speech tag our data using CRFTagger (Phan, 2006) and then look for "my *X*" patterns where *X* is a sequence of tokens terminating in a noun, analogous to our attribute-extraction pattern (Section 3).[5] When a user uses such a "my *X*" construction, we match the filler *X* against our attribute lists for each class. If the filler is on a list, we call it a *self-distinguishing attribute* of a user. We then apply our knowledge of the self-distinguishing attribute and its corresponding class in one of the following three ways:

(1) *ARules*: Using Attribute-Based Rules to Override a Classifier When human-annotated data is available for training and testing a supervised classifier, we refer to it as *gold standard* data. Our first technique provides a simple way to use our identified self-distinguishing attributes in conjunction with a classifier trained on gold-standard data. If the user has any self-distinguishing attributes, we assign the user to the corresponding class; otherwise, we trust the output of the classifier.

(2) *Bootstrapped*: Automatic Labeling of Training Examples Even without gold standard training data, we can use our self-distinguishing attributes to automatically bootstrap annotations. We collect a large pool of unlabeled users and their tweets, and we apply the *ARules* described above to label those users that have self-distinguishing attributes. Once an example is auto-annotated, we delete the self-distinguishing attributes from the user's content. This prevents the subsequent learning algorithm from trivially learning the rules with which we auto-annotated the data. Next, the auto-annotated examples are used as training data for a supervised system.[6] Finally, when applying the *Bootstrapped* classifiers, we can still apply the *ARules* as a post-process (although in practice this made little difference in our final results).

(3) *BootStacked*: Gold Standard and *Bootstrapped* Combination Although we show that an accurate classifier can be trained using auto-annotated *Bootstrapped* data alone, we also test whether we can combine this data with any gold-standard training examples to achieve even better performance. We use the following simple but

[3] `http://trec.nist.gov/data/tweets/` This corpus was developed for the TREC Microblog track (Soboroff et al., 2012).

[4] Note that "I am a man" occurs only 20 times. Users also reveal their names in "my name is X" patterns in several hundred tweets, but this is small compared to cases of self-distinguishing *attributes*. Exploiting these alternative patterns could nevertheless be a possible future direction.

[5] While we used an "off the shelf" POS tagger in this work, we note that taggers optimized specifically for social media are now available and would likely have resulted in higher tagging accuracy (e.g. Owoputi et al. (2013)).

[6] Note that while our target gender task presents mutually-exclusive output classes, we can still train classifiers for other categories without clear opposites (e.g. for labeling users as *Parents* or *Doctors*) by using the 1-class classification paradigm (Koppel and Schler, 2004).

effective method for combining data from these two sources, inspired by prior techniques used in the domain adaptation literature (Daumé III and Marcu, 2006). We first use the trained *Bootstrapped* system to make predictions on the entire set of gold standard data (gold train, development, and test sets). We then use these predictions as *features* in a classifier trained on the gold standard data. We refer to this system as the *BootStacked* system in our evaluation.

5 Twitter Gender Prediction

To test the use of self-distinguishing attributes in user classification, we apply our methods to the task of gender classification on Twitter. This is an important and intensely-studied task within academia and industry. Furthermore, for this task it is possible to semi-automatically acquire large amounts of ground truth (Burger et al., 2011). We can therefore benchmark our approach against state-of-the-art supervised systems trained with plentiful gold-standard data, giving us an idea of how well our *Bootstrapped* system might compare to theoretically top-performing systems on other tasks, domains, and social media platforms where such gold-standard training data is not available.

Gold Data Our data is derived from the corpus created by Burger et al. (2011). Burger et al. observed that many Twitter users link their Twitter profile to homepages on popular blogging websites. Since "many of these [sites] have well-structured profile pages [where users] must select gender and other attributes from dropdown menus," they were able to link these attributes to the Twitter users. Using this process, they created a large multi-lingual corpus of Twitter users and genders.

We filter non-English tweets from this corpus using the LID system of Bergsma et al. (2012) and also tweets containing URLs (since many of these are spam) and re-tweets. We then filter users with <40 tweets and randomly divide the remaining users into 2282 training, 1140 development, and 1141 test examples.

Classifier Set-up We train logistic-regression classifiers on this gold standard data via the LIBLINEAR package (Fan et al., 2008). We optimize the classifier's regularization parameter on development data and report final results on the held-out test examples. We also report the results of our new attribute-based strategies (Section 4) on the test data. We report *accuracy*: the percentage of examples labeled correctly.

Our classifiers use both *BoW* and *Usr* features (Section 2). To increase the generality of our *BoW* features, we preprocess the text by lower-casing and converting all digits to special '#' symbols. We then create real-valued features that encode the *log*-count of each word in the input. While Burger et al. (2011) found "no appreciable difference in performance" when using either binary presence/absence features or encoding the frequency of the word, we found real-valued features worked better in development experiments. For the *Usr* features, we add special beginning and ending characters to the username, and then create features for all character n-grams of length two-to-four in the modified username string. We include n-gram features with the original capitalization pattern and separate features with the n-grams lower-cased.

Unlabeled Data For *Bootstrapped* training, we also use a pool of unlabeled Twitter data. This pool comprises the union of 2.2 billion tweets from 05/2009 to 10/2010 (O'Connor et al., 2010), 1.9 billion tweets collected from 07/2011 to 11/2012, and 80 million tweets collected from the followers of 10-thousand location and language-specific Twitter feeds. We filter this corpus as above, except we do not put any restrictions on the number of tweets needed per user. We also filter any users that overlap with our gold standard data.

Bootstrapping Analysis We apply our *Bootstrapped* auto-annotation strategy to this unlabeled data, yielding 789,285 auto-annotated examples of users and their tweets. The decisions of our bootstrapping process reflect the true gender distribution; the auto-annotated data is 60.5% **Female**, remarkably close to the 60.9% proportion in our gold standard test set. Figure 1 shows that a wide range of self-distinguishing attributes are used in the auto-annotation process. This is important because if only a few attributes are used (e.g. *wife/husband* or *penis/vagina*), we might systematically miss a segment of users (e.g. young people that don't have husbands or wives, or people that don't frequently talk about their genitalia). Thus a wide range of common-sense knowledge is useful for bootstrapping, which is one reason why automatic approaches are needed to acquire it.

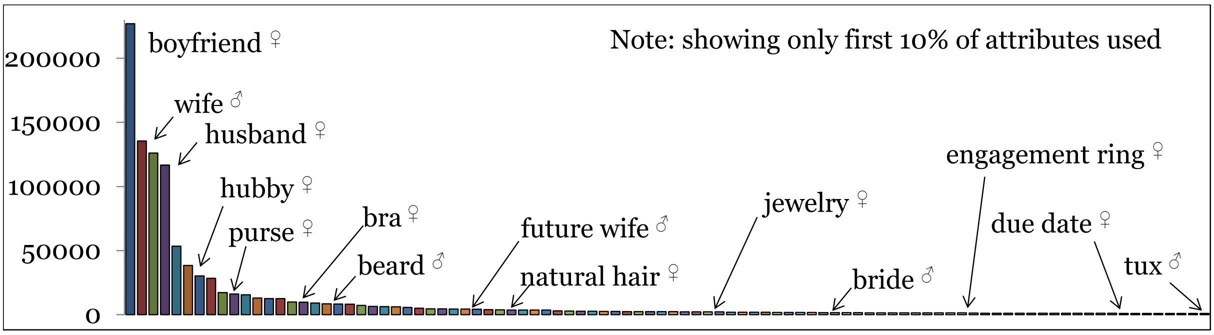

Figure 1: Frequency with which attributes are used to auto-annotate examples in the bootstrapping approach. The plot identifies some attributes and their corresponding class (labeled via gender symbol).

Majority-class baseline	60.9
Supervised on 100 examples	72.0
Supervised on 2282 examples	84.0
Supervised on 100 examples + *ARules*	74.7
Supervised on 2282 examples + *ARules*	84.7
Bootstrapped	86.0
BootStacked	**87.2**

Table 3: Classification accuracy (%) on gold standard test data for user gender prediction on Twitter

6 Results

Our main classification results are presented in Table 3. The majority-class baseline for this task is to always choose **Female**; this achieves an accuracy of 60.9%. A standard classifier trained on 100 gold-standard training examples improves over this baseline, to 72.0%, while one with 2282 training examples achieves 84.0%. This latter result represents the current state-of-the-art: a classifier trained on thousands of gold standard examples, making use of both *Usr* and *BoW* features. Our performance compares favourably to Burger et al. (2011), who achieved 81.4% using the same features, but on a very different subset of the data (also including tweets in other languages).[7]

Applying the *ARules* as a post-process significantly improves performance in both cases (McNemar's, $p<0.05$). It is also possible to use the *ARules* as a stand-alone system rather than as a post-process, however the coverage is low: we find a distinguishing attribute in 18.3% of the 695 **Female** instances in the test data, and make the correct decision in 96.9% of these cases. We find a distinguishing attribute in 11.4% of the 446 **Male** instances, with 86.3% correct decisions.

The *Bootstrapped* system substantially improves over the state-of-the-art, achieving 86% accuracy and doing so *without using any gold standard training data*. This is important because having thousands of gold standard annotations for every possible user characterization task, in every domain and social media platform, is not realistic. Combining the bootstrapped classifier with the gold standard annotations in the *BootStacked* model results in further gains in performance.[8] These results provide strong validation for both the inherent utility of class-attributes knowledge in user characterization and the effectiveness of our specific strategies for exploiting such knowledge.

Figure 2 shows the learning curve of the *Bootstrapped* classifier. Performance rises consistently across all the auto-annotated training data; this is encouraging because there is theoretically no reason not to vastly increase the amount of auto-annotated data by collecting an even larger collection of tweets. Finally, note that most of the gains of the *Bootstrapped* system appear to derive from the tweet content itself, i.e. the *BoW* features. However, the *Usr* features are also helpful at most training sizes.

We provide some of the top-ranked features of the *Bootstrapped* system in Table 4. We see that a variety of other common-sense knowledge is learned by the system (e.g., the association between males and urinals, boxers, fatherhood, etc.), as well as stylistic clues (e.g. **Female** users using *betcha* and *xox* in their writing). The username

[7] Note that it is possible to achieve even higher performance on gender classification in social media if you have further information about a user, such as their full first and last name (Burger et al., 2011; Bergsma et al., 2013).

[8] We observed no further gains in accuracy when applying the *ARules* as a post-process on top of these systems.

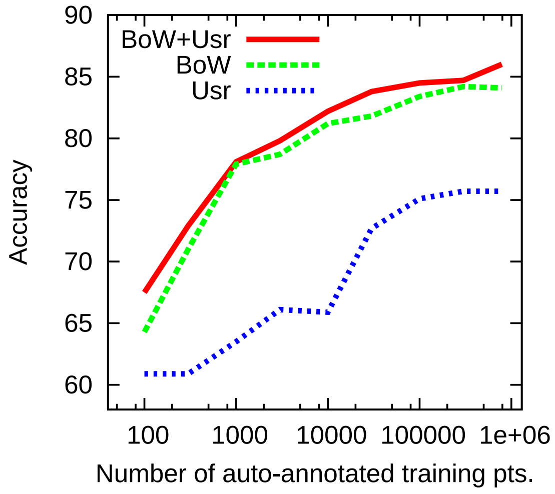

Figure 2: Learning curve for *Bootstrapped* logistic-regression classifier, with automatically-labeled data, for different feature classes.

Male *BoW* features: wife, wifey, sucked, shave, boner, boxers, missus, installed, manly, in-laws, brah, urinal, kickoff, golf, comics, ubuntu, homo, nhl, jedi, fatherhood, nigga, movember, algebra

Male *Usr* features: boy, mike, ben, guy, mr, dad, jr, kid, tony, dog, lord, sir, omar, dude, man, big

Female *BoW* features: hubby, hubs, jewelry, sewing, mascara, fabulous, bf, softball, betcha, motherhood, perky, cozy, zumba, xox, cuddled, belieber, bridesmaid, anorexic, jammies, pad

Female *Usr* features: mrs, mom, jen, lady, wife, mary, joy, mama, pink, kim, diva, elle, woma, ms

Table 4: Examples of highly-weighted *BoW* (content) and *Usr* (username) features (in descending order of weight) in the *Bootstrapped* system for predicting user gender in Twitter.

features capture reasonable associations between gender classes and particular names (such as *mike, tony, omar*, etc.) and also between gender classes and common nouns (such as *guy, dad, sir*, etc.).

7 Related Work

User Characterization The field of sociolinguistics has long been concerned with how various morphological, phonological and stylistic aspects of language can vary with a person's age, gender, social class, etc. (Fischer, 1968; Labov, 1972). This early work therefore had an emphasis on analyzing the *form* of language, as opposed to its *content*. This emphasis continued into early machine learning approaches, which predicted author properties based on the usage of function words, parts-of-speech, punctuation (Koppel et al., 2002) and spelling/grammatical errors (Koppel et al., 2005).

Recently, researchers have focused less on the sociolinguistic implications and more on the tasks themselves, naturally leading to classifiers with feature representations capturing content in addition to style (Schler et al., 2006; Garera and Yarowsky, 2009; Mukherjee and Liu, 2010). Our work represents a logical next step for content-based classification, a step partly suggested by Schler et al. (2006) who noted that "those who are interested in automatically profiling bloggers for commercial purposes would be well served by considering additional features - which we deliberately ignore in this study - such as author self-identification."

Many recent papers have analyzed the language of social media users, along dimensions such as ethnicity (Eisenstein et al., 2011; Rao et al., 2011; Pennacchiotti and Popescu, 2011; Fink et al., 2012) time zone (Kiciman, 2010), political orientation (Rao et al., 2010; Pennacchiotti and Popescu, 2011) and gender (Rao et al., 2010; Burger et al., 2011; Van Durme, 2012).

Class-Attribute Extraction The idea of using simple patterns to extract useful semantic relations goes back to Hearst (1992) who focused on hyponyms. Hearst reports that she "tried applying this technique to meronymy (i.e., the part/whole relation), but without great success." Berland and Charniak (1999) did have success using Hearst-style patterns for part-whole detection, which they attribute to their "very large corpus and the use of more refined statistical measures for ranking the output." Girju et al. (2006) devised a supervised classification scheme for part/whole relation discovery that integrates the evidence from multiple patterns. These efforts focused exclusively on the *meronymy* relation as used in WordNet (Miller et al., 1990). Indeed, Berland and Charniak (1999) attempted to filter out attributes that were regarded as *qualities* (like *driveability*) rather than parts (like *steering wheels*) by removing words ending with the suffixes *-ness*, *-ing*, and *-ity*. In our work, such qualities are not filtered and are ultimately valuable in classification; for example, the attributes *peak fertility* and *loveliness* are highly associated with females.

As subsequent research became more focused on applications, looser definitions of class attributes were adopted. Almuhareb and Poesio (2004) automatically mined class attributes that include parts, qualities, and those with an "agentive" or "telic" role with the class. Their extended set of attributes was shown to enable an improved representation of nouns for the purpose of clustering these nouns into semantic concepts. Tokunaga et al. (2005) define attributes as properties that can serve as focus words in questions about a target class; e.g. *director* is an attribute of a *movie* since one might ask, "Who is the *director* of this *movie*?" Another line of research has been motivated by the observation that much of Internet search consists of people looking for *values* of various class attributes (Bellare et al., 2007; Paşca and Van Durme, 2007; Paşca and Van Durme, 2008; Alfonseca et al., 2010). By knowing the attributes of different classes, search engines can better recognize that queries such as "altitude guadalajara" or "population guadalajara" are seeking *values* for a particular city's "altitude" and "population" *attributes* (Paşca and Van Durme, 2007). Finally, note that Van Durme et al. (2008) compared instance-based and class-based patterns for broad-definition attribute extraction, and found both to be effective.

Of course, text-mining with custom-designed patterns is not the only way to extract class-attribute information. Experts can manually specify the attributes of entities, as in the WordNet project (Miller et al., 1990). Others have automatically extracted attribute relations from dictionary definitions (Richardson et al., 1998), structured online sources such as Wikipedia infoboxes, (Wu and Weld, 2007) and large-scale collections of high-quality tabular web data (Cafarella et al., 2008). Attribute extraction has also been viewed as a sub-component or special case of the information obtained by general-purpose knowledge extractors (Schubert, 2002; Pantel and Pennacchiotti, 2006).

NLP Applications of Common-Sense Knowledge The kind of information derived from class-attribute extraction is sometimes referred to as a type of *common-sense knowledge*. The need for computer programs to represent common-sense knowledge has been recognized since the work of McCarthy (1959). Lenat et al. (1990) defines common sense as "human consensus reality knowledge: the facts and concepts that you and I know and which we each assume the other knows."

While we are the first to exploit common-sense knowledge in user characterization, common sense has been applied to a range of other problems in natural language processing. In many ways WordNet can be regarded as a collection of common-sense relationships. WordNet has been applied in a myriad of NLP applications, including in seminal works on semantic-role labeling (Gildea and Jurafsky, 2002), coreference resolution (Soon et al., 2001) and spelling correction (Budanitsky and Hirst, 2006). Also, many approaches to the task of sentiment analysis "begin with a large lexicon of words marked with their prior polarity" (Wilson et al., 2009). Like our class-attribute associations, the common-sense knowledge that the word *cool* is positive while *unethical* is negative can be learned from associations in web-scale data (Turney, 2002). We might also view information about *synonyms* or *conceptually-similar words* as a kind of common-sense knowledge. In this perspective, our work is related to recent work that has extracted distributionally-similar words from web-scale data and applied this knowledge in tasks such as named-entity recognition (Lin and Wu, 2009) and dependency parsing (Täckström et al., 2012).

8 Conclusion

We have proposed, developed and successfully evaluated a novel approach to user characterization based on exploiting knowledge of user class attributes. The knowledge is obtained using a new algorithm that discovers *distinguishing* attributes of particular classes. Our approach to discovering distinguishing attributes represents a significant new direction for research in class-attribute extraction, and provides a valuable bridge between the fields of user characterization and lexical knowledge extraction.

We presented three effective techniques for leveraging this knowledge within the framework of supervised user characterization: rule-based post-processing, a learning-by-bootstrapping approach, and a stacking approach that integrates the predictions of the bootstrapped system into a system trained on annotated gold-standard training data. All techniques lead to significant improvements over state-of-the-art supervised systems on the task of Twitter gender classification.

While our technique has advanced the state-of-the-art on this important task, our approach may prove even more useful on other tasks where training on thousands of gold-standard examples is not even an option. Currently we are exploring the prediction of finer-grained user roles, such as *student*, *waitress*, *parent*, and so forth, based on extensions to the process laid out here.

References

Enrique Alfonseca, Marius Paşca, and Enrique Robledo-Arnuncio. 2010. Acquisition of instance attributes via labeled and related instances. In *Proc. SIGIR*, pages 58–65.

Abdulrahman Almuhareb and Massimo Poesio. 2004. Attribute-based and value-based clustering: An evaluation. In *Proc. EMNLP*, pages 158–165.

Kedar Bellare, Partha P. Talukdar, Giridhar Kumaran, Fernando Pereira, Mark Liberman, Andrew McCallum, and Mark Dredze. 2007. Lightly-Supervised Attribute Extraction. In *NIPS Workshop on Machine Learning for Web Search*.

Shane Bergsma and Dekang Lin. 2006. Bootstrapping path-based pronoun resolution. In *Proc. Coling-ACL*, pages 33–40.

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.

Shane Bergsma, Mark Dredze, Benjamin Van Durme, Theresa Wilson, and David Yarowsky. 2013. Broadly improving user classification via communication-based name and location clustering on twitter. In *Proc. NAACL*.

Matthew Berland and Eugene Charniak. 1999. Finding parts in very large corpora. In *Proc. ACL*, pages 57–64.

Alexander Budanitsky and Graeme Hirst. 2006. Evaluating WordNet-based measures of lexical semantic relatedness. *Computational Linguistics*, 32(1):13–47.

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.

Michael J. Cafarella, Alon Y. Halevy, Daisy Zhe Wang, Eugene Wu, and Yang Zhang. 2008. WebTables: exploring the power of tables on the web. *Proc. PVLDB*, 1(1):538–549.

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

Hal Daumé III and Daniel Marcu. 2006. Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research*, 26.

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.

Clayton Fink, Jonathon Kopecky, Nathan Bos, and Max Thomas. 2012. Mapping the Twitterverse in the developing world: An analysis of social media use in Nigeria. In *Proc. International Conference on Social Computing, Behavioral Modeling, and Prediction*, pages 164–171.

John L. Fischer. 1968. Social influences on the choice of a linguistic variant. *Word*, 14:47–56.

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

Daniel Gildea and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. *Computational Linguistics*, 28:245–288.

Roxana Girju, Adriana Badulescu, and Dan Moldovan. 2006. Automatic discovery of part-whole relations. *Computational Linguistics*, 32(1):83–135.

Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proc. Coling*, pages 539–545.

Emre Kiciman. 2010. Language differences and metadata features on Twitter. In *Proc. SIGIR 2010 Web N-gram Workshop*, pages 47–51.

Moshe Koppel and Jonathan Schler. 2004. Authorship verification as a one-class classification problem. In *Proc. ICML*, pages 489–495.

Moshe Koppel, Shlomo Argamon, and Anat Rachel Shimoni. 2002. Automatically categorizing written texts by author gender. *Literary and Linguistic Computing*, 17(4):401–412.

Moshe Koppel, Jonathan Schler, and Kfir Zigdon. 2005. Determining an author's native language by mining a text for errors. In *Proc. KDD*, pages 624–628.

William Labov. 1972. *Sociolinguistic Patterns*. University of Pennsylvania Press.

Douglas B. Lenat, R. V. Guha, Karen Pittman, Dexter Pratt, and Mary Shepherd. 1990. CYC: toward programs with common sense. *Commun. ACM*, 33(8):30–49.

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

Dekang Lin, Kenneth Church, Heng Ji, Satoshi Sekine, David Yarowsky, Shane Bergsma, Kailash Patil, Emily Pitler, Rachel Lathbury, Vikram Rao, Kapil Dalwani, and Sushant Narsale. 2010. New tools for web-scale N-grams. In *Proc. LREC*, pages 2221–2227.

John McCarthy. 1959. Programs with common sense. In *Proc. Teddington Conference on the Mechanization of Thought Processes*, pages 75–91. London: Her Majesty's Stationery Office.

Ken McRae, George S. Cree, Mark S. Seidenberg, and Chris McNorgan. 2005. Semantic feature production norms for a large set of living and nonliving things. *Behavior Research Methods*, 37(4):547–559.

George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J. Miller. 1990. Introduction to WordNet: an on-line lexical database. *International Journal of Lexicography*, 3(4).

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.

Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A. Smith. 2013. Improved part-of-speech tagging for online conversational text with word clusters. In *Proc. of NAACL*.

Patrick Pantel and Marco Pennacchiotti. 2006. Espresso: leveraging generic patterns for automatically harvesting semantic relations. In *Proc. Coling-ACL*, pages 113–120.

Marius Paşca and Benjamin Van Durme. 2007. What you seek is what you get: extraction of class attributes from query logs. In *Proc. IJCAI*, pages 2832–2837.

Marius Paşca and Benjamin Van Durme. 2008. Weakly-supervised acquisition of open-domain classes and class attributes from web documents and query logs. In *Proc. ACL-08: HLT*, pages 19–27.

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.

Xuan-Hieu Phan. 2006. CRFTagger: CRF English POS Tagger. `crftagger.sourceforge.net`.

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.

Delip Rao, Michael Paul, Clay Fink, David Yarowsky, Timothy Oates, and Glen Coppersmith. 2011. Hierarchical bayesian models for latent attribute detection in social media. In *Proc. ICWSM*, pages 598–601.

Joseph Reisinger and Marius Paşca. 2009. Latent variable models of concept-attribute attachment. In *Proc. ACL-IJCNLP*, pages 620–628.

Stephen D. Richardson, William B. Dolan, and Lucy Vanderwende. 1998. MindNet: Acquiring and structuring semantic information from text. In *Proc. ACL-Coling*, pages 1098–1102.

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.

Lenhart Schubert. 2002. Can we derive general world knowledge from texts? In *Proc. HLT*, pages 84–87.

Fabrizio Sebastiani. 2002. Machine learning in automated text categorization. *ACM Comput. Surv.*, 34:1–47.

Ian Soboroff, Dean McCullough, Jimmy Lin, Craig Macdonald, Iadh Ounis, and Richard McCreadie. 2012. Evaluating real-time search over tweets. In *Proc. ICWSM*.

Wee Meng Soon, Hwee Tou Ng, and Daniel Chung Yong Lim. 2001. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4).

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.

Kosuke Tokunaga, Jun'ichi Kazama, and Kentaro Torisawa. 2005. Automatic discovery of attribute words from web documents. In *Proc. IJCNLP*, pages 106–118.

Peter D. Turney. 2002. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In *Proc. ACL*, pages 417–424.

Benjamin Van Durme, Ting Qian, and Lenhart Schubert. 2008. Class-driven attribute extraction. In *Proc. Coling*, pages 921–928.

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

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2009. Recognizing contextual polarity: An exploration of features for phrase-level sentiment analysis. *Computational Linguistics.*, 35(3):399–433.

Fei Wu and Daniel S. Weld. 2007. Autonomously semantifying Wikipedia. In *Proc. CIKM*, pages 41–50.

Omar Zaidan, Jason Eisner, and Christine Piatko. 2007. Using “annotator rationales” to improve machine learning for text categorization. In *Proc. NAACL-HLT*.