Aligning Entity Names with Online Aliases on Twitter

Kevin McKelvey **Peter Goutzounis**
Stephen da Cruz **Nathanael Chambers**
Department of Computer Science
United States Naval Academy
nchamber@usna.edu

Abstract

This paper presents new models that automatically align online aliases with their real entity names. Many research applications rely on identifying entity names in text, but people often refer to entities with unexpected nicknames and aliases. For example, *The King* and *King James* are aliases for *Lebron James*, a professional basketball player. Recent work on entity linking attempts to resolve mentions to knowledge base entries, like a wikipedia page, but linking is unfortunately limited to well-known entities with pre-built pages. This paper asks a more basic question: can aliases be aligned without background knowledge of the entity? Further, can the semantics surrounding alias mentions be used to inform alignments? We describe statistical models that make decisions based on the lexicographic properties of the aliases with their semantic context in a large corpus of tweets. We experiment on a database of Twitter users and their usernames, and present the first human evaluation for this task. Alignment accuracy approaches human performance at 81%, and we show that while lexicographic features are most important, the semantic context of an alias further improves classification accuracy.

1 Introduction

A wide range of research in natural language processing focuses on entities. These range from basic language tasks like coreference resolution to broader aggregation applications like sentiment analysis and information extraction. Building an accurate picture of an entity (e.g., aggregate sentiment toward the entity, entity tracking across websites, database population) requires an understanding of all the varying ways people refer to that entity. Tracking ”facebook” is not enough to know how people feel about it, as mentions of ”fbook”, ”FB”, and ”the book” also need to be understood. Although many applications exist for tracking known mentions of entities, less research exists for detecting nicknames and aliases.

This paper presents new models to align an entity's name (e.g., “Bank of America”) with its online aliases (“BAmerica”) and nicknames (“BofA”). Unlike the traditional *entity linking* task that relies on known knowledge base (KB) entries, our task is unique by removing the assumption that a KB is available for each entity. Instead, we simply begin with an entity name and an alias, and ask if the two are likely to refer to the same real-world entity. By asking this more basic question first, several research threads will benefit.

For instance, aligning entity names is important to *sentiment analysis*, but typically ignored for simplification reasons. Companies track social media for mentions of their company in hopes of identifying the public sentiment toward them. Political races rely on similar models, tracking mentions of politicians (“Trump” might be negatively referred to as “Frump”). Research on contextual sentiment analysis has exploded as a result, but the vast majority assumes a single known entity name. In fact, this paper's work originally came about because the authors wanted to track events surrounding ‘Bank of America’, but we kept coming across unexpected new aliases that referred to the company.

Another application is *user profiling* across websites. User accounts that span multiple websites often use different usernames. Most research in this area has focused on aligning account attributes and graph structure. This paper con-

Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media , pages 25–35,
Valencia, Spain, April 3-7, 2017. ©2017 Association for Computational Linguistics

tributes by first addressing the more basic challenge of username alignment.

Finally, this paper also furthers research in *event detection*. If a subset of users on Twitter are talking about a *Katy Perry* concert next week, the task is to extract the date and artist. However, are they referring to the same concert when other users mention *Fruit Sister*? Still others might discuss *Katey Parry* (a misspelling) and *Katherine Hudson* (previous name)? Despite the popularity of this artist, some of these names don't exist in preconstructed KBs. The challenge is exacerbated when the target artist is relatively unknown. This paper experiments with new learning models to align examples like these using only the corpus in which they appear.

The first set of models rely on purely lexicographic characteristics. We propose a series of character and word-based features, trained with discriminative classifiers. Many aliases share obvious characteristics, such as acronym usage and word shortening. These models learn the patterns used when people create nicknames.

The second set of models take a distributional semantics approach. Names like *fruit sister* and *katy perry* have no obvious lexical overlap, so the task of aligning the aliases is impossible without understanding their usage in language. We first present experiments with distributional word vectors to represent the context of aliases, and then measure vector similarity to inform the alignment decision. We then round off the contextual approach with word embeddings from recent neural network research in NLP.

To our knowledge, these are the first machine learned models that align entity names without prior knowledge of the entities. Further, we describe the first human study to measure task difficulty and compare model performance. The lexical and semantic models approach human performance on the task.

2 Previous Work

This paper aligns entity names ("Shem Ayegba") to their aliases ("shemo4real"). Relevant previous work falls into three categories: detecting attributes of online users, entity linking, and user linking.

A large body of work has looked into **attribute detection** of social media users, particularly those on Twitter. Given a particular user, what is that person's age (Nguyen et al., 2013), gender, education background, political orientation, ethnicity (Bergsma et al., 2013), etc. This paper is related in learning a different type of attribute: aliases and nicknames.

Early experiments on attribute detection rely on a user's text (e.g., tweets) to predict a range of different attributes (Rao et al., 2010). Many build learning models that are applicable across a variety of different user attributes (Chen et al., 2015a; Volkova et al., 2015; Beretta et al., 2015). Among the attributes, political preferences is a frequent area of research, again relying on features from user tweets, and making use of graph-based algorithms over their friends' attributes (Golbeck and Hansen, 2011; Conover et al., 2011; Wong et al., 2013; Cohen and Ruths, 2013; Volkova et al., 2014). Pla et al. (2014) even uses sentiment analysis.

This paper is related to attribute extraction in the desire to learn about an online user. However, the task at hand is to resolve *mentions* of a user's online alias (i.e., twitter handle) and *mentions* of a named entity (i.e., a business or a person's real name). Unlike the body of work on attribute extraction, we assume we do not have an entity's body of published text, but instead just observed their name in text.

More related to our goal of name understanding is research on gender identification (Rao et al., 2010; Burger et al., 2011; Van Durme, 2012; Filippova, 2012; Ciot et al., 2013). In many of these, the name of a user is informative. Most work thus uses the name as an indicator, but then also uses the user's text posts to assist. The name itself offers insight into this answer, and some models rely first on dictionaries of names before backing off to a machine learner trained on user tweets (Liu and Ruths, 2013; Volkova and Yarowsky, 2014).

Chen et al. (Chen et al., 2015b) pursued a novel line of investigation which uses only a user's name, and infers *visual attributes* by using click-through data on name searches and web images. Although very different in the type of predicted knowledge, this paper is similar in that we only have a name and must infer properties of that person, namely *who* it is in real life.

Others have studied whether gender and language can be identified from only the username. They looked at characters and morphological units with an unsupervised learning approach (Jaech and Ostendorf, 2015). This paper is similar in challenge in that we also start with only the user's name, but the methods are very different.

This paper is also a new form of **entity linking**. Entity linking is an active research field that aims to resolve an entity mention (e.g., "michael jordan") to an entry in a knowledge base (e.g., michael jordan's wikipedia page). Most work in this field relies on the text context around the mention to measure similarity with the text description of the entity in the KB, such as a wikipedia entry's text (Adafre and de Rijke, 2005; Bunescu and Pasca, 2006; Mihalcea and Csomai, 2007; Milne and Witten, 2008; Dredze et al., 2010; Ratinov et al., 2011; Han et al., 2011; Demartini et al., 2012; Moro et al., 2014; Moro and Navigli, 2015). All of these papers assume they have knowledge base entries with text to assist in resolving entity mentions. This paper is different in that we don't have a knowledge base, but just an online alias. We start from the assumption that we don't have text from that alias, and must rely solely on observed mentions and properties of the name/alias itself.

Finally, **user linking** across website communities is an active research area. Research typically focuses on finding similarities in the social network structure (Backstrom et al., 2007; Narayanan and Shmatikov, 2009; Tan et al., 2014; Liu et al., 2016), similar user attributes across the sites (Li and Lin, 2014; Goga, 2014; Zafarani et al., 2015; Goga et al., 2015; Naini et al., 2016), or both (Liu et al., 2014; Chung et al., 2014). Not many focus on usernames, with the exception of Liu et al. (2013), but they study how to identify different users who use *the same username*. Very recently, Wang et al. (2016) describe a heuristic text comparison model between different usernames. While similar in goal to this paper, we apply a far wider range of features, incorporate semantic knowledge, and use modern machine learning techniques.

3 Datasets

The main experiment dataset is a list of name/alias pairs. Table 1 shows a few examples of these pairs. The list is comprised of approximately 110k pairs of names and their actual twitter handles extracted from a single day of tweets in November 2015. We selected 110k tweets, and paired the name listed on the profile of the user who wrote the tweet with the same user's twitter handle. This name/handle pair is a single datum in the dataset.

Profile Name	**Twitter Handle**
Shem Ayegba	@shemo4real
mimi sanson viola	@palomahepzibah
Alisha	@alishajii
The Hammer of Facts	@FactsHammer
John Kielbowicz	@kibblebits

Table 1: Examples of name/handle pairs in the dataset.

We then generated another 110k *false* pairs by randomly selecting twitter handles and matching them with incorrect profile names. Combined with the correct 110k pairs, the resulting dataset is 220k name/alias pairs, half of which are correct and half incorrect. This list is then broken up into 160k pairs for training, 40k for a held-out test set, and 20k for a development set. Finally, we remove all pairs in the test set that contain a username or a handle that also appears in the training set. This avoids all overlap between train and test. A very minor reduction in test set size resulted from this.

Since our experiments rely on corpus-based features, we use one year of tweets from the freely available Twitter streaming API from Aug 2014 to Aug 2015. We refer to this corpus later as our "one-year tweet corpus".

4 Models

We model the discovery of online aliases for a real name as a pairwise classification task. Given an entity's name and a single alias, what is the probability that the two refer to the same entity? This pairwise classifier can then be employed in a variety of practical systems, such as producing a ranked list of likely names for an alias, or the inverse problem of identifying the most likely alias for a target entity.

4.1 Learning Models

We experimented with both support vector machines (SVM) and maximum entropy classifiers. The input is an entity name and alias pair $x = (e, a)$ that is mapped to a set of features $f(x)$, described next. We used the Stanford CoreNLP toolkit [1] for implementations of the models using the software's default parameters.

[1] http://stanfordnlp.github.io/CoreNLP/

4.2 Lexicographic Features

The primary features for a name/alias pair are based on the characters and string commonalities between the two. We experimented with thirteen such lexicographic features.

Edit Distance. This feature uses the Levenshtein edit distance between the name and alias. This computes the number of character additions, deletions, and substitutions that are required to turn the alias into the name. The name and alias are both lowercased first, and the @ symbol removed from the alias. White space is included in the comparison. The feature returns $1.0 - editdist(n, a)$. The higher the value, the more similar the strings.

Exact Match. If the lowercased name and alias are exact matches after white space is removed, this binary feature is turned on.

First and Last Name. Three features were developed based on an entity having a first and last name. If the alias starts with the first name of the entity, the feature returns the length of that name. The last name feature is the same, but looking instead for the last name. A third feature is a binary feature that indicates if the entity name appears in whole (with spaces removed) anywhere in the alias. For example, *John Williams* occurs in the twitter handle, *@JohnWilliams2*.

Percent Substring. Returns the number of overlapping characters between the alias and name, divided by the length of the name. This is a representation of the percentage of an alias that contains a name. However, this feature is not case-sensitive nor sequential, meaning that the position of the characters does not matter, only their presence is accounted for.

Starts and Ends With. These are two distinct features. Starts-with compares the lowercased alias-name pair and returns the count of overlapping characters in the longest shared prefix. The ends-with feature is the same, but instead counts the longest shared suffix.

Capital Substring. This feature splits the alias into substrings based on capital letters, and returns the number of such substrings that are contained within the name (not necessarily capitalized).

Acronyms. Two features: if the alias is an acronym of the lowercased name's tokens, the binary acronym feature is turned on. Second, a capital-acronym feature compares the number of capital letters in the name that occur in the alias. This feature is the number of overlapping matches.

Exact Capitalization. Capitalization is a binary feature that compares the capital letters of a name to the capital letters of an alias and returns true on an identical match. This overlaps in purpose with the acronym features, but it captures a broader set.

Reverse Substring. This is a binary feature that returns true if the alias is the name in reverse, or vice-versa. This was inspired by examples like *Janey* and *@yenaj*.

Unused Lexicographic Features. Two lexical features were ultimately abandoned: one-word-substring and semi-acronym. One-word-substring returns the length of any one word in the entity that was contained in the handle. Semi-acronym attempted to construct acronyms using full prepositions (i.e. "BofA"). Neither had a positive effect on development set results.

4.3 Semantic Features

The above lexical features approximate what is available to a naïve user/system who is presented with a name/alias pair. Overlap and similarity of the characters is the only available means to make a decision, and if the name and alias share little similarity, there is no remaining recourse to fall back on.

This section describes our attempts at broadening the learning model into shallow semantics by making use of a large corpus of tweets. Semantic similarity has a rich history in computational semantics of representing words with *context vectors*. This is often called distributional semantics where a word (or a name in our case) *is known by the company it keeps* (Firth, 1957). We follow the traditional approach by representing a name (or alias) by a vector of word counts from the words seen in tweets surrounding the name. The following shows a tweet with entity *Dominic Dyer*, and the corresponding context vector.

> **Tweet**
> The launch of the new book by **Dominic Dyer** at Birdfair today.
>
> **Context Vector**
> (the 2, launch 1, of 1, new 1, book 1, by 1, at 1, Birdfair 1, today 1)

We use the one-year tweet corpus to compute context vectors for each name and alias. All observed mentions of a name (or alias) are summed into its single aggregate semantic context vector.

Word Vectors. Each entity and alias has a context word vector, as described above. All context tokens were lowercased, and leading/trailing punctuation removed. We then created three features using the vectors: binary, cosine, and averaged-cosine variants. The **cosine** feature is the traditional context vector feature: compute the cosine between the name's vector and the alias' vector. The **binary** feature is a binarized version of cosine, true if the cosine is greater than 0.1 and false otherwise. The **averaged-cosine** feature is motivated by the observation that large vectors tend to result in higher cosine scores (more likely to contain overlapping tokens). This feature returns the difference between an entity's average cosine score and its cosine score with the alias in question:

$$f(n, a) = \frac{cos(n, a) * N}{\sum_x cos(n, x)} - 1 \qquad (1)$$

where n is the name vector, a is the alias vector, and N is the number of aliases. If the $cos(n, a)$ is bigger than average, this equation returns a value greater than 0.

Word Embeddings. Recent work on neural networks have shown *word embeddings* to outperform traditional context vectors on a variety of NLP tasks. We thus trained a skip-gram neural net on our twitter data, and created word embeddings for the entity names and aliases. The open-source Word2Vec from deeplearning4j was used as the model implementation[2]. Word2Vec implements a skip-gram neural model where the input is the target word (or entity name), and predicted output are the words to the left and right of the target. A word's embedding is the vector of weights for the hidden layer. The implementation is based on Mikolov et al. (2013).

Once word embeddings are learned for all observed names and aliases, we duplicate the three word vector features described above. Since word embeddings are also vectors, the features are implemented the same.

5 Experiments

We focus on the basic task of predicting whether an alias belongs to a name, given only a corpus of tweets and no other entity knowledge. Experiments use the name/alias pairs as described above in Datasets. Given a name/alias pair, the task is to predict "yes" or "no" to whether the two mentions likely refer to the same entity. As a binary prediction task, the random baseline is 50% accuracy. Each name in the dataset appears in both one correct pair with its true twitter handle, and one incorrect pair with a randomly selected twitter handle.

We use accuracy as the evaluation metric with its normal definition:

$$Accuracy = \frac{\#correct}{N} \qquad (2)$$

where N is the size of the evaluation set.

We report accuracy on the entire evaluation set (**Accuracy: all**) as well as a subset of the evaluation that includes only entity pairs such that the entity name and the twitter handle were each seen at least 100 times in the one-year twitter corpus (**Accuracy: 100**). This second set serves the purpose of distinguishing the importance of frequency when using semantic vector features. Entity mentions that rarely occur have sparse vectors, and a prediction relies solely on the lexicographic features.

The features in the models were developed on the training and development sets only. We report on several feature ablation tests on the development set. Feature ablation was not conducted on the test set. The test set was only used to generate the final results table.

Both SVM and MaxEnt models used the default settings in CoreNLP, but we only report MaxEnt results as neither significantly outperformed the other.

Four baseline models are included to illustrate the non-trivial nature of this task. At first blush, it may appear that this paper's focus is a trivial string match. Part of the motivation for this paper's focus is to illustrate how even the most basic of username mapping tasks is non-trivial. The first baseline, **Simple-Match**, simply lowercases and removes white space from both the name and alias. If the two changed strings now match exactly, then the baseline predicts match. The second baseline, **Alpha-Match**, is a variation of Simple-Match, but

[2] http://deeplearning4j.org/word2vec

also removes all characters not in the a-z alphabet (e.g., 'david' and '88david-2' match). The third baseline, **Alpha-RelaxMatch** relaxes Alpha-Match by only requiring the first 5 characters in both name and alias to match. Finally, the fourth baseline is a machine learned model using only the edit distance feature (**Edit-Dist**) on lowercased and white-space condensed strings.

Finally, we ran a human evaluation to measure ideal performance on this task. We randomly selected 2,010 pairs from the bigger test set, and several undergraduates were asked to judge whether each name/alias pair was likely or not to be the same entity. We ran our best models on this same smaller test set and report accuracy.

6 Results

Table 2 shows results on the development set for the basic model with additional character-based features. The **Simple-Match** baseline performs surprisingly low at 57.66%, **Alpha-Match** slightly better, and **Alpha-RelaxMatch** the best baseline at 69.57%. Entity names and their twitter handles are not often clear matches. The machine learned baseline that uses only edit distance somewhat surprising barely performs better than random chance.

The most important feature is **first** and **last name** matching, increasing accuracy from 57.66% to 72.44%. These features match if the entity's first (last) name appears at the start (end) of the alias. Other features with further 4% gains each are the percent substring feature, and the numeric feature "starts with" (or "ends with") that returns prefix or suffix matches. This partly overlaps with the first name feature, but is more general and significantly improves performance.

The above experiment only had access to an entity's name and possible alias. The best classifier with just lexicographic features is 81.63% accurate. The next experiment expands to assume the availability of a corpus with entity mentions. Starting with distributional word vectors, Table 3 shows the performance on the subset of data where the entity mention was seen at least 100 times. Word vectors are useless for new and rare mentions, so we focus on the portion of data where vectors are applicable. The word vector features combine for a 1.4% relative gain.

Though a small gain, for insight into how these features might help, see Figure 1 for the top token

Development Set Accuracy

	Acc: All	Acc: 100
Base (Simple-Match)	57.66	45.24
Base (Alpha-Match)	61.72	47.62
Base (Alpha-RMatch)	69.57	54.76
Base (ML Edit-Dist)	57.66	45.24
+first-last	72.44	59.62
+percent substring	77.02	62.00
+starts-ends	81.06	67.46
+acronyms	81.16	67.6
+all lexical feats	81.63	70.93

Table 2: Development set results and feature comparison. Numbers are % accuracy: 81.6 and 70.9

Dev Set Accuracy with Word Vectors

	Accuracy: 100
All Lexical	70.93%
w/ binary word vector	71.92%
w/ cosine word vector	72.22%
w/ average word vector	70.93%
+ all vector features	**71.93**%

Table 3: Word vector accuracy on the development set. Each row is the feature by itself without the other vector features. The final row is the inclusion of all three features in one learned model.

Test Set Accuracy

	Acc: All	Acc: 100
Base (Simple-Match)	54.63	42.02
Base (Edit-Dist)	56.80	50.08
Base (Alpha-Match)	58.98	44.94
Base (Alpha-RMatch)	67.33	52.25
All Lexical	80.73	71.60
+binary word vec	80.82	72.59
+cosine word vec	80.81	72.47
+average word vec	80.73	71.60
+all vec features	**80.83**	**72.59**

Table 4: Word vector accuracy results on the Test set. All features are included.

Test Set Accuracy with Embeddings

	Acc: All	Acc: 100
All Lexical	80.73	71.60
Lexical+vector	80.83	72.59
Lexical+embed	80.71	71.2T0

Table 5: Accuracy on the Test set when adding word embedding features.

Precision and Recall Comparison

	Match		Non	
	P	**R**	**P**	**R**
Simple-Match	100	13.1	51.3	100
All Lexical	93.5	67.8	72.9	94.8
All Lexical + vec	92.8	68.5	73.2	94.2

Table 6: Precision and Recall on the Test set for correctly identifying alignment pairs.

Human Evaluation Test Set

	Accuracy
Baseline (Edit-Dist)	49.03
Baseline (Simple-Match)	56.14
All Lexical + vector	80.96
Human	**81.01**

Table 7: Human evaluation comparison on a separate test set of approximately 2000 pairs.

counts in entity/alias vectors. Word context can capture their typical contexts as long as they occur in the corpus.

After developing features on the dev set, we ran the same feature ablation one time on the test set. Results are shown in Table 4. The improvement from the individual word vector features is similar to the development set, confirming that we did not overfit model development. The final relative improvement on the Accuracy:100 set is again 1.4% over the lexical-only model. This improvement is statistically significant ($p < 0.000001$, McNemar's test, 2-tailed).

We also tested word embeddings as a substitute feature for distributional word vectors. We trained our own embeddings for each entity string and twitter handle using word2vec. Table 5 shows the results as virtually identical to the distributional vectors. The two essentially capture the same information in this particular task as including both types of features offered no further gain.

Table 6 gives precision and recall for correctly guessing *yes* and *no* as individual labels.

Finally, Table 7 shows human performance compared to our best model. The two are virtually the same at 81% accuracy. The all lexical model is able to capture the same knowledge a typical human brings to identifying likely aliases of new entities.

7 Error Analysis

Several questions hide behind the accuracy numbers. We briefly address a few of them here.

7.1 Why is accuracy for high frequency entities lower?

The results for **Acc:100** in the result tables are significantly lower than the **Acc:All** results. These are the entities that occurred at least 100 times in our one year corpus, so they represent entities that are discussed more frequently than others. The best baseline at 67% on Test drops to only 52% for this subset of frequent entities.

The main reason for high frequency entities to be more difficult appears to be due to the fact that high frequency entities have less similarity in their twitter handles. We computed the edit distance between each entity's name and handle, dividing by the length of the entity string. The average normalized edit distance across all development set pairs is **0.97**. Computing this normalized edit distance for only entities occurring 100 or more times, and the average is twice as high at **1.84**. The direct answer for why high frequency entities are more difficult is that their profile names have far less in common with their twitter handles. But why?

Manual error analysis revealed that high frequency entities often have short profile names.

dominic dyer	**URL**, **born**, **badger**, anneka, svenska, lionaid, #lionsbetrayed, #bantrophyhunting, interviewed, tells, **trust**, bbc, ceo, speech, @badgertrust, ...
@domdyer70	**URL**, **badger**, london, join, march, cull, protest, wildlife, **trust**, badgers, against, army, saturday, @lionaid, **born**, ...
@CraftsmenLtd	**URL**, @poppyscupcakes, #ff, #creativebizhour, @etsy, cupcakes, mock, clever, @lizzie_chantree, @sweettoothmarti, @chichicardsuk, vine, #handmadehour, ...

Figure 1: Word vectors for an entity and two possible aliases. *@domdyer70* is correct for *dominic dyer*.

This surprised the authors as we assumed uniform behavior in profile names. It turns out that many frequently mentioned online entities have *shorter profile names*, most likely due to their popularity. Our manual analysis shows that many of these use only given names or nicknames, avoiding surnames. When someone is less known, they perhaps prefer a full name to distinguish who they are. Once someone is known, shorter names become a benefit of the popularity. However, this shorter name behavior does *not* transfer to the twitter handle. Since twitter handles must be unique across all users, short names are unavailable and tend to be longer for everyone. This appears to explain most of why so many more edits are required in the edit distance computation of high frequency entities.

High frequency entities are more difficult because they contain less lexicographic overalp due to conciseness of their profile names. This also explains why the trained classifier performs lower based on only character-based features. This leads us to analyze the non-character context vectors.

7.2 Do context vectors actually help?

The 1.4% relative improvement on Test when adding context vectors is not particularly impressive, though the improvement is statistically significant (using McNemar's test) on the test set. One possible explanation for the smaller gain is that word vectors do help, but they help on the same entity/alias pairs that lexical features already correctly classify. To test this reasoning, we trained the model with only word vector features and wihtout the full lexical model. Do context vectors improve over the baselines?

Table 8 shows their accuracy on the Test set is **57.16%**. Note that this vector-only model completely ignores character-level similarity between the entity's name and alias. If the name is "david" and the alias is "@david2", this trained model does not take that into account. The features only compare the contexts observed around those two mentions in the corpus. Its performance is a 14% relative increase over a random baseline, and notably, almost 10% relative over our strongest baseline (Alpha-RelaxedMatch, comparing the name and alias strings).

	Acc: 100
Baseline (Random Guess)	50.00
Baseline (Alpha-RMatch)	52.25
Trained only w/ context vecs	**57.16%**

Table 8: Measuring performance of the word context vector features by themselves as the only classifier input. Accuracy is reported on the pairs seen at least 100 times in the corpus.

Clearly the context vectors do provide a useful signal, albeit less of a contribution when the full lexical information is also included.

7.3 What types of errors remain to be solved?

The main observed error occurs when the alias has no overt lexical relation to its true entity name **and** they rarely occur in the corpus. Some examples are given here (these are correct names with their twitter handles):

Nicola @Luckyminx79
Avery @moodyscience
Tobin Heath @lanaxjauregui
Amanda @bieberfto2l
Manuel @angel1110497

Without lexical clues, and no word context vector due to sparsity, our models fail. Humans obviously fail too. Our results around 80% accuracy suggest a ceiling of 20% of the data contains these errors. A far more complex and resource-heavy model that can spider alias feeds, conversations, and profiles to build a user profile is required. Section 2 discusses several relevant works. In regards to this paper's core question (can we resolve aliases without pre-knowledge of entities?), these errors are not addressed.

8 Discussion

This is the first proposal, to our knowledge, to align entity mentions with their online aliases *without prior knowledge of the entities*. While similar in spirit to *entity linking*, there is no knowledge base with a grounded referent. The challenge instead is to resolve the plethora of ways people refer to the same person or organization. It is a stripped down, base task, aimed at experimenting with how accurate such a knowledge-light model can be. We aimed to experiment with the bare minimum knowledge.

We found that prediction actually approaches *human-level* performance when using a rich set of lexicographic features. This is perhaps unsurprising because humans don't have background knowledge of random online users, so they also rely solely on lexical observations. It is encouraging that our models approximate some of this reasoning, although even humans only achieve 81% accuracy on this task.

Semantic word vectors achieved a slight increase in accuracy over the lexical model, but were shown useful when used as features by themselves. This suggests other tasks may benefit from building context representations for their entities. One important caveat is that the increased performance is only for entities seen frequently, otherwise semantic context cannot be extracted.

By simplifying the resolution task to pairwise comparison, we believe this work benefits a number of research areas. This paper is perhaps not a practical task by itself, but a very useful first tool. We will release the code as an easy-to-use API (as well as the data). First, it can be used as a plugin to improve *user linking* across websites, comparing user names and profile names ahead of time. Second, entity linking might benefit as another input on top of the usual suite of features. Many papers ignore mention comparison and only focus on context, but our results suggest that a fresh look at mention names is needed. Third, and perhaps most significant, contextual sentiment analysis can be expanded beyond keyword search. Instead of a strict entity match, a broader net can be cast to include the nicknames and aliases of the desired entity. The authors are already leveraging it for this purpose.

The training and test data used in this paper's experiments can be accessed online at www.usna.edu/Users/cs/nchamber. We hope that its release will assist related research needs.

Acknowledgments

This work was supported by a grant from the Office of Naval Research, and a grant from the DoD HPC Modernization Program Office.

References

Sisay Fissaha Adafre and Maarten de Rijke. 2005. Discovering missing links in wikipedia. In *Proceedings of the 3rd international workshop on Link discovery*, pages 90–97.

Lars Backstrom, Cynthia Dwork, and Jon Kleinberg. 2007. Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography. In *Proceedings of the 16th international conference on World Wide Web*, pages 181–190.

Valentina Beretta, Daniele Maccagnola, Timothy Cribbin, and Enza Messina. 2015. An interactive method for inferring demographic attributes in twitter. In *Proceedings of the 26th ACM Conference on Hypertext & Social Media*, pages 113–122.

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 *Proceedings of the Conference on Human Languate Technologies (HLT-NAACL)*, pages 1010–1019.

Razvan C. Bunescu and Marius Pasca. 2006. Using encyclopedic knowledge for named entity disambiguation. In *Proceedings of the European Chapter of the Association on Computational Linguistics (EACL)*, volume 6, pages 9–16.

John D. Burger, John Henderson, George Kim, and Guido Zarrella. 2011. Discriminating gender on twitter. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1301–1309. Association for Computational Linguistics.

Xin Chen, Yu Wang, Eugene Agichtein, and Fusheng Wang. 2015a. A comparative study of demographic attribute inference in twitter. *Proceedings of the International Conference on Web and Social Media*, pages 590–593.

Yan-Ying Chen, Yin-Hsi Kuo, Chun-Che Wu, and Winston H. Hsu. 2015b. Visually interpreting names as demographic attributes by exploiting click-through data. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 44–50. AAAI Press.

Cheng Ta Chung, Chia Jui Lin, Chih Hung Lin, and Pu Jen Cheng. 2014. Person identification between different online social networks. In *Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)-Volume 01*, pages 94–101. IEEE Computer Society.

Morgane Ciot, Morgan Sonderegger, and Derek Ruths. 2013. Gender inference of twitter users in non-english contexts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1136–1145.

Raviv Cohen and Derek Ruths. 2013. Classifying political orientation on twitter: It's not easy! In *Proceedings of the International Conference on Web and Social Media*, pages 91–99.

Michael D. Conover, Bruno Gonçalves, Jacob Ratkiewicz, Alessandro Flammini, and Filippo Menczer. 2011. Predicting the political alignment of twitter users. In *Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on*, pages 192–199. IEEE.

Gianluca Demartini, Djellel Eddine Difallah, and Philippe Cudré-Mauroux. 2012. Zencrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In *Proceedings of the 21st international conference on World Wide Web*, pages 469–478.

Mark Dredze, Paul McNamee, Delip Rao, Adam Gerber, and Tim Finin. 2010. Entity disambiguation for knowledge base population. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 277–285. Association for Computational Linguistics.

Katja Filippova. 2012. User demographics and language in an implicit social network. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, page 1478–1488.

J. R. Firth. 1957. A synopsis of linguistic theory, 1930-1955. *Studies in Linguistic Analysis*.

Oana Goga, Patrick Loiseau, Robin Sommer, Renata Teixeira, and Krishna P. Gummadi. 2015. On the reliability of profile matching across large online social networks. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1799–1808.

Oana Goga. 2014. *Matching user accounts across online social networks: methods and applications*. Ph.D. thesis, LIP6-Laboratoire d'Informatique de Paris 6.

Jennifer Golbeck and Derek Hansen. 2011. Computing political preference among twitter followers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1105–1108.

Xianpei Han, Le Sun, and Jun Zhao. 2011. Collective entity linking in web text: a graph-based method. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 765–774.

Aaron Jaech and Mari Ostendorf. 2015. What your username says about you. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 2032–2037.

Chung-Yi Li and Shou-De Lin. 2014. Matching users and items across domains to improve the recommendation quality. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 801–810.

Wendy Liu and Derek Ruths. 2013. What's in a name? using first names as features for gender inference in twitter. In *AAAI spring symposium: Analyzing microtext*, volume 13, pages 10–16.

Jing Liu, Fan Zhang, Xinying Song, Young-In Song, Chin-Yew Lin, and Hsiao-Wuen Hon. 2013. What's in a name?: an unsupervised approach to link users across communities. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 495–504.

Siyuan Liu, Shuhui Wang, Feida Zhu, Jinbo Zhang, and Ramayya Krishnan. 2014. Hydra: Large-scale social identity linkage via heterogeneous behavior modeling. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pages 51–62.

Li Liu, William K. Cheung, Xin Li, and Lejian Liao. 2016. Aligning users across social networks using network embedding. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-16)*, pages 1774–1780.

Rada Mihalcea and Andras Csomai. 2007. Wikify!: linking documents to encyclopedic knowledge. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 233–242.

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

David Milne and Ian H. Witten. 2008. Learning to link with wikipedia. In *ACM Conference on Information and Knowledge Management*, pages 509–518.

Andrea Moro and Roberto Navigli. 2015. Semeval-2015 task 13: multilingual all-words sense disambiguation and entity linking. *Proceedings of SemEval-2015*, pages 288–297.

Andrea Moro, Alessandro Raganato, and Roberto Navigli. 2014. Entity linking meets word sense disambiguation: a unified approach. *Transactions of the Association for Computational Linguistics*, 2:231–244.

Farid M. Naini, Jayakrishnan Unnikrishnan, Patrick Thiran, and Martin Vetterli. 2016. Where you are is who you are: User identification by matching statistics. *IEEE Transactions on Information Forensics and Security*, 11(2):358–372.

Arvind Narayanan and Vitaly Shmatikov. 2009. De-anonymizing social networks. In *2009 30th IEEE symposium on security and privacy*, pages 173–187. IEEE.

Dong-Phuong Nguyen, Rilana Gravel, R.B. Trieschnigg, and Theo Meder. 2013. "how old do you think i am?" a study of language and age in twitter. In *Proceedings of the Seventh International AAAI Conference on Weblogs and Social Media*. AAAI Press.

Ferran Pla and Lluís F. Hurtado. 2014. Political tendency identification in twitter using sentiment analysis techniques. In *Proceedings of the International Conference on Computational Linguistics*, pages 183–192.

Delip Rao, David Yarowsky, Abhishek Shreevats, and Manaswi Gupta. 2010. Classifying latent user attributes in twitter. In *Proceedings of the 2nd international workshop on Search and mining user-generated contents*, pages 37–44.

Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. 2011. Local and global algorithms for disambiguation to wikipedia. In *Association for Computational Linguistics: Human Language Technologies*, pages 1375–1384.

Shulong Tan, Ziyu Guan, Deng Cai, Xuzhen Qin, Jiajun Bu, and Chun Chen. 2014. Mapping users across networks by manifold alignment on hypergraph. In *AAAI*, volume 14, pages 159–165. AAAI Press.

Benjamin Van Durme. 2012. Streaming analysis of discourse participants. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 48–58. Association for Computational Linguistics.

Svitlana Volkova and David Yarowsky. 2014. Improving gender prediction of social media users via weighted annotator rationales. In *NIPS 2014 Workshop on Personalization*.

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

Svitlana Volkova, Yoram Bachrach, Michael Armstrong, and Vijay Sharma. 2015. Inferring latent user properties from texts published in social media. In *AAAI*, pages 4296–4297. AAAI Press.

Yubin Wang, Tingwen Liu, Qingfeng Tan, Jinqiao Shi, and Li Guo. 2016. Identifying users across different sites using usernames. *Procedia Computer Science*, 80:376–385.

Felix Ming Fai Wong, Chee Wei Tan, Soumya Sen, and Mung Chiang. 2013. Quantifying political leaning from tweets and retweets. *Proceedings of the International Conference on Web and Social Media*, 13:640–649.

Reza Zafarani, Lei Tang, and Huan Liu. 2015. User identification across social media. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 10(2).