Disambiguation of Standardized Personal Name Variants

Patricia Driscoll
Department of Computer Science
Johns Hopkins University
Baltimore, MD 21218
pdriscoll@gmail.com

David Yarowsky
Department of Computer Science
Johns Hopkins University
Baltimore, MD 21218
yarowsky@cs.jhu.edu

Abstract

A growing body of research addresses name disambiguation as part of coreference and entity resolution systems, but the systems do not robustly resolve the ambiguity introduced by standardized personal name variants, or nicknames. In many languages, personal name variants are governed by morphological and phonological constraints, providing a dataset rich in features which may be used to train and run matching systems. We present a supervised learning method to address the problem of standardized personal name variant matching in English. The system integrates information from multiple sources into a weighted voting model which significantly outperforms baseline methods.

Keywords

Nicknames, Name Disambiguation, Name-Matching, Personal Name Variants, Truncation, Morphology

1 Introduction

As data sources such as the Internet expand in size, the study of entity disambiguation, whose goal is to cluster large numbers of name mentions according to entity referents, has become increasingly important. As a crucial part of this process, personal name disambiguation aims to create linguistically motivated links between personal names using information about their structure and morphology which can be mined from multiple sources. Personal name disambiguation has been studied both for its place in the entity disambiguation process, where it can aid tasks like coreference resolution [1, 11, 13], and for the production of stand-alone tools and name disambiguation resources, such as proper name ontologies [9, 8], onomastica [15], and fuzzy name-matching tools [12] which accept candidate pairs as input. An important area of current research, name disambiguation systems have the capability to take into account social and occupational titles, honorifics, and variation in capitalization and punctuation [18]. Problems similar to personal name variant disambiguation, including name transliteration and cognate matching for common nouns, have also been studied in the context of machine translation [7, 10, 16].

Personal name variants, such as standardized nicknames of personal names, have been little-studied elements of the name disambiguation problem. Those systems that include standardized nicknames as equivalent to their corresponding full forms typically do so using a pre-packaged dataset such as a nickname pair list [6], or by simple string-matching methods which do not take into consideration the morphological relationship between standardized nicknames and their corresponding full forms, leaving the systems susceptible to error [12]. In this paper, we address the task of scoring arbitrary pairs of names and nicknames, creating a module to aid name disambiguation and overcome problems presented by language change, incomplete datasets, and scarce resources. By robustly extending the pool of potentially coreferent personal names, this work will enhance the recall of state-of-

the-art entity disambiguation systems.

2 Personal Name Variants

The terms 'nickname' and 'hypocoristic', are commonly used to refer to several distinct phenomena when describing personal names. One class of nicknames are pet names which are related to personal or relationship traits, but are generally linguistically unrelated to the full form name (Elvis Presley→The King). Non-standardized nicknames, while often related to long forms, are typically used to refer to one person in particular (Richard Nixon→Tricky Dick). Since links between full forms and familiar forms in these cases relate to entities rather than the names themselves, they are not able to be generalized for use in name-matching systems.

For the purposes of this paper, we will use 'nicknames' to refer to the set of standardized familiar form variants of personal names. Such familiar forms are linguistically linked to full forms, although links are governed by a combination of morphological and phonological constraints and convention that can range from highly regular (Christina→Chris) to relatively opaque (John→Jack). Name dictionaries linking these standardized familiar forms are not typically available electronically, and where available are often incomplete. Further complicating the picture is the idea that such familiar forms are somewhat productive, dynamic aspects of language for which it may be difficult to limit tasks to use of static resources.

Variation in nicknames is common, with sociological trends, idiosyncracies, and the desire to distinguish different individuals with the same name cited as some of the sources of variability [2, 17, 5]. Additionally, linguistic patterns for nickname formation are complex, governed by morphological and phonological constraints with diverse ordering conventions [17, 2]. Despite this variation, many languages do share common nickname pattern characteristics, the most commonly discussed of which are truncation, reduplication, and augmentation (See Table 1).

Because of the complexity of nicknaming patterns, handwritten rules for personal name variant matching are both time-consuming and incomplete. In this paper, we explore a variety of learning methods for building personal name variant resources and for doing matching tasks dynamically. Evaluation data for this domain is limited: name variant resources are scarce and incomplete in English and virtually nonexistent in many other languages. This data scarcity further attests to the utility of the automatic acquisition of usable resources.

Language	Full Form	Nickname
French	Paul	Paulo
	Jacques	Jacquot
	Raphaelle	Raphie
	Louis	Loulou
	Marguerite	Margot
Russian	Svetlana	Sveta
	Polina	Polya
	Nataliya	Natasha
	Yevengenya	Zhenya
Italian	Giovanna	Gianna
	Luigi	Gino
	Rosalia	Lietta
	Guiseppe	Pino

Table 1: *Nickname formation is highly structured, though each language has its own set of constraints.*

To obtain training and evaluation data, first names taken from 1990 U.S. Census data were used to query a nickname database at www.oxygen.com/babynamer, which yielded a set of 2543 name-nickname pairs using 907 of the census names. 1837 nicknames were represented in the data, which often included multiple nicknames for particular first names (Jennifer→ Jen, Jenny) as well as nicknames which were associated with multiple first names (Robert, Roberto →Bob). Although it contained many name variants, the resource was not exhaustive, thus presenting challenges for evaluation. One example of the lack of coverage occurred with spelling variants of full form names, in which nicknames were often linked to one form but not another.

An interesting property of the data was that more common first names (using probabilities given by census.gov) were likely to have more nicknames than their less common counterparts (Figure 1).

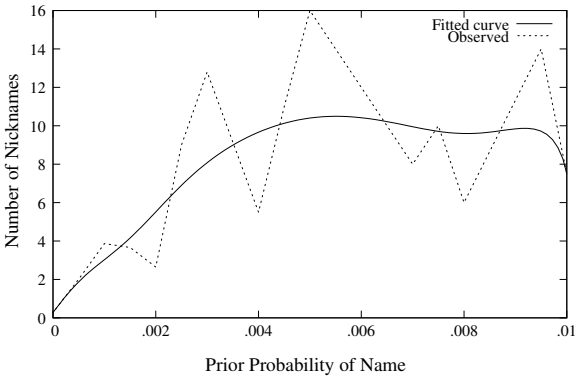

Fig. 1: *More common names have more nicknames.*

3 Detecting Name Variant Pairs

To link standardized personal name variants with corresponding full form names, several methods were chosen for strengths in accuracy, flexibility, and limited data and annotation requirements. Used both individually and in combination, the methods chosen are likely to work well in a variety of settings, including quick ramp-up for languages with little available data.

3.1 Web-Based Extraction

Although there are few languages for which entire name variant dictionaries are available for download from the Internet, the Internet can nonetheless be a reliable tool to use in the creation of such resources. Personal web pages, increasingly available in many languages, are rich in name information which can fill in gaps left by other systems, mediating recall limitations that may be addressed through the addition of alternate methods. In particular, web extraction often covers name pairs which are not related by simple morphological rules, providing knowledge about name variant matches that would be difficult to access otherwise. Findings of the web extraction method may also be used to give a boost to correct pairs which are found in the results of other methods, but are ranked below erroneous pairs.

There has been much recent work on web-based extraction, in which systems typically start with a hand-picked seed phrase [4] or seed instances [3]. For the web extraction component, we started from the seed phrase "My name is *full form* ... friends call me *variant*", issued this query to Internet search engine yahoo.com, and collected the first page of results for each seed nickname. This weakly supervised method ensures quick and easy extension to other languages. More strongly supervised methods with added alternate seed phrases and use of results beyond the one-page range would likely improve the performance of this component in English, both by eliminating erroneous hits and by expanding coverage.

Seed Nickname	Candidate Full Form
Katia	Katarina
Lynn	Madeline
Debbie	Phyllis
Lee	My

Table 2: *Web extraction component finds nicknames in a variety of contexts.*

Many of the recovered full form/variant pairs retrieved are complex but correct (Table 2). Because of the potential unreliability of this extraction method, candidate full forms that did not appear in the census data (Lee←My) were discarded. Another source of error was the fact that the "my name is" match immediately preceding the seed nickname on a particular web page was not always relevant: in some cases, long lists of online personal ads included phrases like "my name is" and "friends call me" as interchangeable (Debbie←Phyllis).

3.2 Morphological Analyzer

To exploit the feature-rich, highly constrained morphological derivation process involved in nickname formation, we used a toolkit as described in [19]. The toolkit uses the Word-Frame model which learns string transduction between inflected and root forms. It has previously been applied primarily to learning verb inflections, and its application to learning nickname inflection patterns is novel. The model itself is flexible, and can model arbitrary affixation morphological processes (prefixation, suffixation, and infixation). While it has no explicit support for processes like vowel harmony and partial word reduplication, [19] asserts that these processes can often be modeled satisfactorily by affixation rules.

Training data consisted of 1000 name-

Morphological Rule	Nickname	Full Form
IE → A	Elsie	Elsa
IE → ERTA	Albie	Alberta
EE → INA	Rosalee	Rosalina
EE → ENE	Charlee	Charlene

Table 3: *The morphological analyzer learns the common morphological nickname inflections from supervised training data.*

1. Exact nickname matches to the beginning or the end of full form.
2. Exact nicname matches anywhere in full form.
3. Lemmatized nickname matches at the beginning or the end of full form.
4. Lemmatized nickname matches anywhere in the full form.

Table 4: *The handwritten truncation rules give four levels of matches.*

nickname pairs. We additionally gave as input to the system a list of 10 suffixes, input which [19] has suggested can improve performance of the analyzer. Table 3 gives examples of some of the rules learned by the morphological analyzer.

3.3 Truncation Rules

In addition, a small set of handwritten truncation rules was used to supplement results from the weakly supervised components. The rules were developed using basic knowledge of left and right truncation with vowel-only augmentation, so as to exploit features requiring only limited knowledge and limited implementation time. Since truncation is a nickname formation mechanism found in many languages, similar rules might be written for other languages. As mentioned above, nickname formation is a complex phenomenon and attempts to provide hardcoded rules for all languages are too limited to capture all of the requisite variation.

These rules (Table 4) provide a rough cut at matching, and were used both to form a baseline and as a supplement to the system, since they provided information on the most likely guesses when highly regular matches existed.

Using the rules above, nicknames using basic truncation such as Elizabeth →Liz, Lizzie, or Beth would be recognized, while more complex forms (Elizabeth→Betsy) would not. The handwritten rules did little to constrain the truncation by pruning out unlikely examples, so Elizabeth would also be matched with candidate nicknames like Eli, Zoe, and Bea. Although highly regular, simple phonological changes were not modeled to limit size of rule set and allow for extensions to other languages.

4 Classifier Combination

Weights were trained using 145 pairs with names appearing in the top ten percent of census data. Since both the web extraction and morphological analysis components were generative and thus likely to be relatively sparse, we included a binary feature indicating whether the candidate appeared on each of these lists, regardless of score. Based on data indicating that number of nicknames varies according to full form prior (see Figure 1), these prior probabilities were also included as a candidate feature. Because Levenshtein distance, with a score cutoff of 4, gave no improved performance, it was not included in the final feature set.

The final weights chosen by the system were as follows:

Feature	Weight
Appears in the Web Extraction List	0.286
Handwritten Truncation Rule Score	0.214
Morphological Analyzer Score	0.143
Web Extraction Ranking	0.143
Full Form Prior Probability	0.143
Appears in Morphological Analyzer List	0.071

5 Experimental Results

Our initial data set of 2543 name-nickname pairs was split into three portions: test data, development data, and training for the morphological analyzer. Names from the top ten percent of the census data were used to create pairs for the test and development sets. Test

Nickname	**Confusors** (distance)	**True Full Form** (distance)
Pammie	Mammie (1) Tommie (2)	Pamela (3)
Jess	Bess (1) Jose (2)	Jessica (3)
Lenny	Benny (1) Wendy (2)	Leonard (6)

Table 5: *Levenshtein distance typically judges unrelated names to be closer than full form names.*

data using 278 pairs of these commonly occurring names and their nicknames was then chosen at random, and was expanded to include all correct names for nicknames seen in testing. Development data using 145 pairs was chosen at random from the remains of this set, and training data of 1000 pairs was selected using only nicknames not seen in test or development.

Baselines were a simple substring match ranked by proportion nickname comprised of full form, Levenshtein distance with a cutoff of 4, and the handwritten truncation rules described in Section 3. Table 5 gives examples which demonstrate why Levenshtein distance is a unsuitable approximation to the complex morphological processes involved in name variant formation. [1]

The dataset, although relatively formal and thorough, did not escape some of the issues inherent in the use of static datasets for standardized name variant matching. One particularly challenging obstacle was a seeming lack of full recall in the test set. A type of legitimate match which was often not given credit in the test set was seen with spelling variants of other full form names for which more extensive pair lists existed (e.g. Deborah → Debbie was in the test data, but Debra → Debbie was not). The inclusion of relatively obscure matches also made the problem of scoring a difficult one: if systems are expected to include matches such as Daisy ← Marguerite before being granted full credit, recall scores will be unrealistically low. To address these issues, components were built to produce ranked lists and scored by means of precision/recall curves. This was the most straightforward approach for integrating and evaluating information from a variety of heterogeneous sources, as some components produced reliable ranked lists, while others were more useful as isolated features into the system. Ongoing work is looking into combining these features into a supervised learning system such as a logistic regression model.

Table 6 shows examples of the kind of matches discovered by each of the system components. Figure 2 shows the performance of the components in isolation as well as the combined system performance. The combined system performance yields significantly superior performance to the baseline systems. It is able to combine the high precision/low recall performance of the web extraction component and the morphological analyzer with the high recall/low precision performance of the truncation rule component.

The weighted voting method, using features described above, outperformed all other methods as recall climbed above 15%. Web extraction and morphological analysis components were competitive at lower recall, indicating that these components were able to find certain pairs with good confidence. Truncation rules, substring, and Levenshtein methods were not strong on precision, although truncation rules remained steady as recall increased, indicating their strength as a component which selects many correct matches among noisy pair results.

6 Conclusions and Future Work

The results reported on matching standardized name variants with full form personal names are indicative of both capabilities in English and, because of the flexibility of the methods used, point to the likelihood of success using the methods described in languages with similar nicknaming conventions. In particular, the use of a trained morphological analyzer requires little data and supervision, and can serve as a stand-alone tool for name-

[1] Alternative noncontextual learned edit distance measures such as [14] would suffer from problems similar to those seen in traditional Levenshtein distance. Instead we look to the morphological analyzer presented in Section 3.2 as a linguistically motivated modeling tool.

Nickname	Full Form	Top Component Choices		
		Truncation Rules	Morphological Analyzer	Web Extraction
Steve	**Stephen** **Steven**	1. Stevie 2. **Steven**	1. Steve 2. Stevie	**1. Stephen** **2. Steven**
Vikie	**Victoria**	1. Vikki 2. Viki	**1. Victoria** 2. Viki	*none*
Chas	**Charles** **Chastity**	1. **Chastity**	*none*	**Charles**

Table 6: *Each of the different components captures an important aspect to the nickname process: the truncation rule component allows high recall, the morphological analysis component allows more flexible matches than simple truncation rules, and the web extraction component is a high precision matcher.*

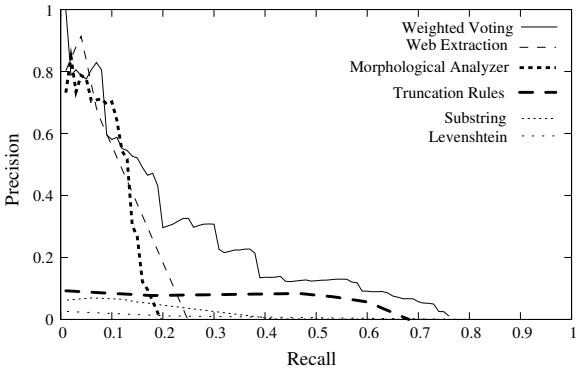

Fig. 2: *The combined system is able to take advantage of the benefits of each of the components: the high precision of the web extractor and the morphological analyzer, and the high recall of the truncation rules.*

matching with limited seed data. Web extraction methods can fill in the gaps where linguistically-motivated methods leave off, and future work which expands the seed phrases used will likely improve on the success shown here.

Because extensive name variant resources are not widely available, expensive to compile, and susceptible to change, the methods described above provide the opportunity to create such resources with limited data, supervision, and implementation time. The methods presented are dynamic and flexible and can be re-trained when data changes, and are rank-based, avoiding recall drop-offs likely in static dictionaries. Improvements in entity disambiguation via higher recall will be achieved by this more dynamic and flexible approach.

We have presented here a general paradigm for learning name variant models. Future work will explore alternative methods within this framework for learning variants, such as SVM classifiers and weighted Levenshtein distance.

In this paper, we do not explore the extension of the methods presented to other types of name variants, such as personal name spelling variants. Such work would likely have the creation of personal name equivalence-classes as a goal, which could then extend nickname results to all members of target classes.

While important, work in languages other than English has been inhibited by the difficulty of test data collection, which speaks further to the need for building reliable systems in these languages. While nickname formation in languages other than English has seen a limited amount of attention in the theoretical linguistics community, the extension of the full set of methods applied in this paper to a larger set of languages is a promising area of further research.

References

[1] A. Bagga and B. Baldwin. Entity-based cross-document coreferencing using the vector space model. In C. Boitet and P. Whitelock, editors, *Proceedings of the Thirty-Sixth Annual Meeting of the Association for Computational Linguistics and Seventeenth International Conference on Computational Linguistics*, pages 79–85, San Francisco, California, 1998. Morgan Kaufmann Publishers.

[2] L. Benua. Identity effects in morphological truncation. In J. Beckman, S. Urbanczyk, and L. Walsh, editors, *University of Massachusetts Occasional Pa-*

pers in Linguistics. University of Massachusetts, 1995.

[3] S. Brin. Extracting patterns and relations from the world wide web. In *WebDB Workshop at 6th International Conference on Extending Database Technology, EDBT'98*, pages 172–183, 1998.

[4] O. Etzioni, M. Cafarella, D. Downey, S. Kok, A.-M. Popescu, T. Shaked, S. Soderland, D. Weld, and A. Yates. Web-scale information extraction in knowitall. In *WWW*, 2004.

[5] J. Ito and A. Mester. Sympathy theory and german truncations. In V. Miglio and B. Moreen, editors, *University of Maryland Working Papers in Linguistics*. University of Maryland, 1997.

[6] Z. Kazi and Y. Ravin. Who's who? identifying concepts and entities across multiple documents. In *International Conference on System Sciences*, 2000.

[7] K. Knight and J. Graehl. Machine transliteration. *Computational Linguistics*, 24(4):599-612, 1998.

[8] C. Krstev, V. Dusko, D. Maurel, and M. Tran. Multilingual ontology of proper names. In *Language and Technology Conference*, 2005.

[9] G. S. Mann. Building a proper noun ontology for question answering. In *Proceedings of SemaNet02: Building and Using Semantic Networks*, pages 16–22, 2002.

[10] G. S. Mann and D. Yarowsky. Multipath translation lexicon induction via bridge languages. In *NAACL*, 2001.

[11] G. S. Mann and D. Yarowsky. Unsupervised personal name disambiguation. In *Proceedings of the Conference on Natural Language Learning*, pages 33–40, 2003.

[12] G. Navarro, R. Baeze-Yates, and J. Arcoverde. Matchsimile: a flexible approximate matching tool for searching proper names. *JASIST*, 54(1), 2003.

[13] Y. Ravin and Z. Kazi. Is hillary rodham clinton the president? disambiguating names across documents. In *Proceedings of the ACL '99 Workshop on Coreference and its Applications*, 1999.

[14] E. Ristad and P. Yianilos. Learning string edit distance. *IEEE Trans. PAMI*, 20(5):522-532, 1998.

[15] S. Sheremetyeva, J. Cowie, S. Nirenburg, and R. Zajac. Multilingual onomasticon as a multipurpose nlp resource. In *LREC*, 1998.

[16] T. Sherif and G. Kondrak. Substring-based transliteration. In *ACL*, 2007.

[17] N. Topintzi. Prosodic patterns and the minimal word in the domain of greek truncated nicknames. In *International Conference of Greek Linguistics*, 2003.

[18] N. Wacholder, Y. Ravin, and M. Choi. Disambiguation of proper names in text. In *Proceedings of Fifth Conference on Applied Natural Language Processing*, pages 202–208, 1997.

[19] R. Wicentowski. Multilingual noise-robust supervised morphological analysis using the wordframe model. In *ACL SIGPHON*, 2004.