The Intelius Nickname Collection: Quantitative Analyses from Billions of Public Records

Vitor R. Carvalho, Yigit Kiran and Andrew Borthwick
Intelius Data Research
500 108th Avenue NE, Bellevue, WA 98004
{vcarvalho,ykiran,aborthwick}@intelius.com

Abstract

Although first names and nicknames in the United States have been well documented, there has been almost no quantitative analysis on the usage and association of these names amongst themselves. In this paper we introduce the *Intelius Nickname Collection*, a quantitative compilation of millions of name-nickname associations based on information gathered from billions of public records. To the best of our knowledge, this is the largest collection of its kind, making it a natural resource for tasks such as coreference resolution, record linkage, named entity recognition, people and expert search, information extraction, demographic and sociological studies, etc. The collection will be made freely available.

1 Introduction

Nicknames are descriptive, invented person names that are frequently used in addition or instead of the person's official name. Very often nicknames are truncated forms of the original name that can be used for convenience — for instance, 'Betsy' instead of 'Elizabeth'.

Previous studies on nicknames have mostly focused on their origins or common descriptions. The Oxford Dictionary of First Names (Hanks et al., 2007), for instance, presents a comprehensive description of origins and common uses of most nicknames in modern English. More quantitative explorations of the subject, such as the one provided by the US Social Security Office[1] tend to focus on baby name selection and on the relative popularity of most common first names.

Alias	Conditional Probability
Betty	4.51%
Beth	3.83%
Liz	3.34%
Elisabeth	0.95%
Betsy	0.92%

Table 1: Nickname Distribution Sample for "Elizabeth"

In this paper we present a quantitative study on nickname usage in the United States. Using billions of personal public records and a state-of-the-art large-scale record linkage system, we were able to generate a comprehensive dataset with millions of name-nickname associations and their relative strength. A small sample of this collection can be seen in Table 1, where the most frequent nicknames associated with the first name "Elizabeth" and their Conditional Alias Probabilities. We explain the derivation of these probabilities in detail in Section 3.3. This collection can provide valuable features and insights for applications as diverse as entity extraction, coreference resolution, people search, language modeling, and machine translation. It will be made freely available for download from the Linguistic Data Consortium.

[1]Popular Baby Names from Social Security Online: http://www.ssa.gov/OACT/babynames/

2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 607–610,
Montréal, Canada, June 3-8, 2012.

2 Prior Work

To the best of our knowledge, there are no comprehensive, empirically derived nickname databases currently made freely available for research purposes. (Bollacker, 2008) contains an extensive database of names and nicknames[2], with listings on over 13,000 given names, containing multiple "variations" for each name. However, this database makes no attempt to distinguish between common and less common variants and skips some very common nicknames. For instance, the entry for "William" lists "Wilmot" and "Wilton" as variants of William but does not list "Bill" or "Billy". (Meranda, 1998) provides a more useful database which appears to also be manually constructed. The database is in the form of Name1|Name2|"substitution likelihood", but the author states in the comments that the substitution likelihood is "mostly guesswork" and the data contains numerous coverage gaps. For instance, common nicknames such as "Jack", "Willy", and "Sally" are all missing.

3 Generating the Nickname Distribution

The nickname collection was derived from billions of public, commercial and web records that power a major commercial People Search Engine. The process described below associates all records belonging to a particular person into clusters, and from these clusters it constructs a final person profile that is used to derive name-alias associations. The entire process is briefly described below.

3.1 Data Collection and Cleaning

The process starts by collecting billions of personal records from three different sources of U.S. personal records. The first source is derived from US government records, such as marriage, divorce and death records. The second is derived from publicly available web profiles, such as professional and social network public profiles. The third type is derived from commercial sources, such as financial and property reports (e.g., information made public after buying a house).

After collection and categorization, all records go through a cleaning process that starts with the removal of bogus, junk and spam records. Then all records are normalized to an approximately common representation. Then finally, all major noise types and inconsistencies are addressed, such as empty/bogus fields, field duplication, outlier values and encoding issues. At this point, all records are ready for the Record Linkage process.

[2]http://www.freebase.com/view/base/givennames/given_name

3.2 Record Linkage Process

The Record Linkage process should link together all records belonging to the same real-world person. That is, this process should turn billions of input records into a few hundred million clusters of records (or profiles), where each cluster is uniquely associated with a real-world unique individual.

Our system follows the standard high-level structure of a record linkage pipeline (Elmagarmid et al., 2007) by being divided into four major components: 1) data cleaning 2) blocking 3) pair-wise linkage and 4) clustering. The data cleaning step was described above. The blocking step uses a new algorithm implemented in MapReduce (Dean et al., 2004) which groups records by shared properties to determine which pairs of records should be examined by the pairwise linker as potential duplicates. The linkage step assigns a score to pairs of records using a supervised pairwise-based machine learning model whose implementation is described in detail in (Sheng et al., 2011) and achieves precision in excess of 99.5% with recall in excess of 80%, as measured on a random set with tens of thousands of human labels. If a pair scores above a user-defined threshold, the records are presumed to represent the same person. The clustering step first combines record pairs into connected components and then further partitions each connected component to remove inconsistent pair-wise links. Hence at the end of the entire record linkage process, the system has partitioned the input records into disjoint sets called profiles, where each profile corresponds to a single person. While the task is very challeging (e.g., many people share common names such as "John Smith") and this process is far from perfect, it is working sufficiently well to power multiple products at Intelius, including a major people search engine.

3.3 Algorithm

We used the MapReduce framework (Dean et al., 2004) to accomodate operations over very large datasets. The main goal of this task is to preserve the relationship amongst different names inside a profile. The algorithm's pseudocode is illustrated in Figure 1.

Many different names can be listed under a profile, including the real name (e.g., the "official" or "legal" name), nicknames, diminutives, typos, etc. In the first phase of the algorithm, a mapper visits all profiles to reveal these names and outputs a <key, value>pair for each name token. The keys are the names, and the values are a list with all other names found in the profile. This is a safe approach since we do not attempt to determine whether a given token is an original name, a diminutive or a typo. Henceforth, we refer to the key as *Name* and the values as *Aliases*.

The reducer will merge all *alias* lists of a given *name*, and count, aggregate and filter them. Since the mapper function produces intermediate pairs with all different names seen inside a profile, reducing them will create a bi-directional relation between *names* and *aliases*, where one can search for all *aliases* of a *name* as well as the reverse. The reducer also estimates conditional probabilities of the *aliases*. The **Conditional Alias Probability (CAP)** of an alias defines the probability of an *alias* being used to denote a person with a given *name*. Specifically, It can be expressed as $CAP(alias_i|name_j) = \frac{count(alias_i \wedge name_j)}{count(name_j)}$, where the $count()$ operator returns the number of profiles satisfying its criteria.

Processing a large number of profiles creates a huge *alias* lists for each *name*. Even worse, most of the *aliases* in that list are typos or very unique nicknames that would not be considered a typical alias for the name. In order to help control this noise, we used the following parameters in the algorithm. ***Alias_Count_Minimum*** sets the minimum number of profiles that should have an *alias* for the alias to be included. ***Total_Count_Minimum*** determines whether we output the whole set of *name* and *aliases*. It is determined by computing the total number of occurrences of the *name*. ***CAP_Threshold*** forces the reducer to filter out *aliases* whose probability is below a threshold.

```
MAP(profile)
1  names := ∅
2  for name ∈ profile
3      names := names ∪ name
4  for current_name ∈ names
5      aliases := ∅
6      for other_name ∈ names
7          if current_name ≠ other_name
8              aliases := aliases ∪ other_name
9      EMIT(current_name, aliases)

REDUCE(key, values)
1  aliaslist := ∅
2  for record ∈ values
3      if aliaslist.contains(record)
4          INCREMENT(aliaslist[record])
5      else
6          aliaslist[record] := 1;
7  SORT-BY-COUNT(aliaslist)
8  COMPUTE-FREQUENCIES(aliaslist)
9  FILTER(aliaslist)
10 EMIT(key, aliaslist)
```

Figure 1: MapReduce Nickname Extractor algorithm

3.4 Analysis

The number of generated name-alias associations depends largely on the specific parameter set used in by the algorithm. While different applications may benefit from different parameters, many of our internal applications had success using the following set of parameters: $Total\_Count\_Minimum = 100$, $Alias\_Count\_Minimum = 10$, and $CAP\_Threshold = 0.1\%$. Using this parameter set, the process generated 331,237 name-alias pairs.

Table 2 shows CAP values for various name-alias pairs. As expected, notice that values of $CAP(X|Y)$ can be completely different from $CAP(Y|X)$, as in the case of "Monica" and "Monic". The collection also shows that completely unrelated names can be associated to a short alias, such as "Al". Notice also that very frequent typos, such as"Jefffrey", are also part of the collection. Finally, very common name abbreviations such as "Jas" for "James" are also part of the set as long as they are statistically relevant.

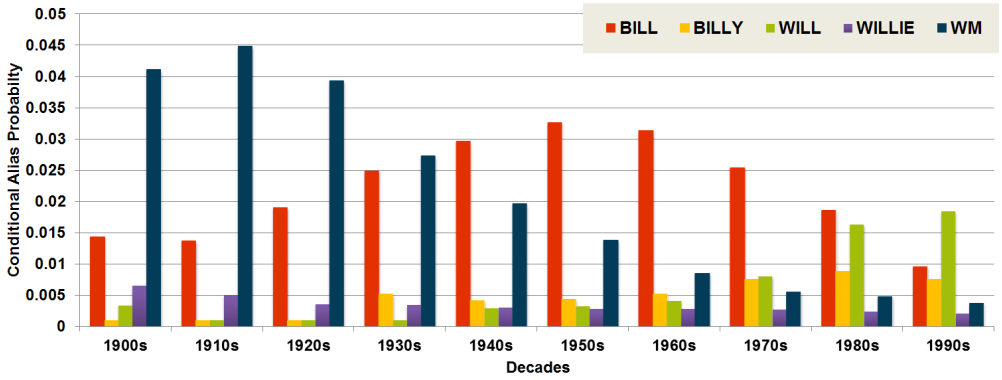

Figure 2: Conditional Probability of "William"'s Aliases over the Decades in the US.

X	Y	$CAP(Y\|X)$
Monica	Monika	1.00%
Monica	Monic	0.26%
Monic	Monica	38.76%
Al	Albert	14.83%
Al	Alfred	8.28%
Al	Alan	4.96%
Jas	James	71.94%
Jas	Jim	7.54%
James	Jas	2.09%
Jefffrey	Jeffrey	40.04%
Jefffrey	Jeff	25.69%

Table 2: Sample CAPs For Multiple Aliases.

3.5 Limitations and Future Explorations

It is important to keep in mind that the collection is only valid for adults in the USA. Also, despite the noise reduction obtained by the algorithm thresholds in Section 3.3, some cases of frequent typos, foreign spellings/transliterations, and abbreviations are still statistically indistinguishable from actual nicknames. For instance, 'WM' (a common abbreviation of William) is as frequent as many of its nicknames. While we could have used a human-edited list to filter out these cases, we decided to keep it in the collection because some applications may benefit from this information. A coreference application, for instance, could infer that "Wm Jones" and "William Jones" have a high probability of being the same person.

Looking forward, there are multiple directions to explore. Besides names, the final record clusters generally contain other information such as addresses, date of birth (DOB), professional titles, etc. As an example, Figure 2 illustrates the probability of the most frequent nicknames of 'William' for people born over different decades in the US. It is interesting to notice that, while 'Bill' was the most likely nickname for people born between the 1940s and 1980s, 'Will' has become significantly more popular since the 80s - to the point that it has become the most likely nickname in the 90s. We believe our next steps will include investigating various migration, economic, sociological and demographic patterns while also leveraging this information in record linkage and coreference resolution modules.

References

K Bollacker, C. Evans, P. Paritosh, et al. 2008. *Freebase: A collaboratively created graph database for structuring human knowledge*. ACM SIGMOD.

Ahmed Elmagarmid, Panagiotis Ipeirotis and Vassilios Verykios 2007. *Duplicate Record Detection: A Survey*. IEEE TKDE 19 (1)

Patrick Hanks, Hardcastle Kate and Flavia Hodges 2007 *Oxford Dictionary of First Names.* Oxford University Press, USA, 2nd edition, ISBN 978-0-19-861060-1.

Deron Meranda 1998 *Most Common Nicknames for First Names* http://deron.meranda.us/data.

Jean-Baptiste Michel et al. 2011 *Quantitative Analysis of Culture Using Millions of Digitized Books.* Science, Vol. 331 no. 6014 pp. 176-182

Sheng Chen, Andrew Borthwick and Vitor R. Carvalho 2011. *The Case for Cost-Sensitive and Easy-To-Interpret Models in Industrial Record Linkage.* International Workshop on Quality in Databases VLDB-2011

Jeff Dean and Sanjay Ghemawat 2004. *MapReduce: Simplified Data Processing on Large Clusters* Symposium on Operating System Design and Implementation OSDI-2004