A Comparison of String Similarity Measures for Toponym Matching

Gabriel Recchia
Institute for Intelligent Systems
University of Memphis
365 Innovation Drive
Memphis, TN 38152
1-970-532-4287
grecchia@memphis.edu

Max Louwerse
Institute for Intelligent Systems
University of Memphis
365 Innovation Drive
Memphis, TN 38152
1-901-678-3803
mlouwerse@memphis.edu

ABSTRACT
The diversity of ways in which toponyms are specified often results in mismatches between queries and the place names contained in gazetteers. Search terms that include unofficial variants of official place names, unanticipated transliterations, and typos are frequently similar but not identical to the place names contained in the gazetteer. String similarity measures can mitigate this problem, but given their task-dependent performance, the optimal choice of measure is unclear. We constructed a task in which place names had to be matched to variants of those names listed in the GEOnet Names Server, comparing 21 different measures on datasets containing romanized toponyms from 11 different countries. Best-performing measures varied widely across datasets, but were highly consistent within-country and within-language. We discuss which measures worked best for particular languages and provide recommendations for selecting appropriate string similarity measures.

Categories and Subject Descriptors
H.2.8 [**Database Management**]: Database Applications – *spatial databases and GIS*

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing – *linguistic processing*

General Terms
Algorithms, Performance, Experimentation

Keywords
toponyms, string similarity, edit distance, gazetteers, duplicate detection, data integration, geographic information retrieval

1. INTRODUCTION
At least two topics relevant to computational models of place—point of interest conflation and place-based data integration—are closely tied to the expansion, search, and conflation of digital gazetteers. Some approaches to these tasks (particularly conflation) involve *toponym matching*, the matching of place names that share a common referent (e.g. *Ting Tsi River*, *Ting-tzu Wan*, *Tingtze River*) [23, 28]. Toponym matching has been studied in geographic record linkage [1, 23, 28], named entity recognition [25], and other tasks in geographic information retrieval (GIR). Hastings [16] developed a novel string similarity measure for use in gazetteer conflation, but the most commonly used measures in GIR remain edit distance, Jaro-Winkler, or English-specific phonetic encoding techniques such as Soundex [23, 25]. Seghal et al. [28] compared three string similarity measures on a geospatial data integration task, finding that edit distance outperformed Jaccard and Jaro-Winkler when mapping between two sets of place names in Afghanistan. Martins [23] used machine learning to classify gazetteer records as duplicates or non-duplicates and compared the importance of several feature types, including eight string similarity measures (see also [33]). Compared with the other factors they investigated, string similarity was the most informative. Of these, Jaccard, Jaro-Winkler, character overlap, and two measures of edit distance were the most useful. However, Martins did not systematically compare their performance.

This study aimed to formulate recommendations facilitating the selection of task-appropriate measures for toponym matching. Although there has been no comprehensive comparison of string similarity measures on a toponym matching task, wide-ranging evaluations have been conducted on the closely related task of personal name matching. Cohen, Ravikumar, and Fienberg [9] compared several string distance measures on a name matching task, finding the best results with a combination of Jaro-Winkler and a token-based distance function. Most relevant to the present study, Christen [5] compared a comprehensive set of 21 commonly used string similarity measures on a set of personal name matching tasks. Although some measures generally did well while others generally did poorly, algorithm performance was task-dependent, and Christen concluded that "there is no single best technique" ([5, p. 13]). Of course, one significant limitation of string similarity techniques is that many unofficial place name variants are not at all similar to each other (e.g., *New York* and *Big Apple*). In such cases, string similarity algorithms are unlikely to provide any benefit. However, they remain popular in gazetteer conflation algorithms due to the large number of place name variants that differ only slightly. Given that the best-performing techniques for personal names were dataset-dependent [17], we anticipated significant variability across datasets of toponyms representing different languages and countries.

2. MATCHING TECHNIQUES
We aimed to evaluate the performance of the comprehensive set of algorithms investigated by [5] in the context of personal name

ACM SIGSPATIAL COMP'13, November 5, 2013. Orlando, FL, USA

matching, but using a toponym matching rather than a name matching task, using datasets representing diverse countries and languages. Measures based on phonetic encoding are lacking for most non-English languages and are not designed for romanized placenames. We therefore restricted our comparisons to the 21 string similarity measures evaluated in [5]. A brief summary of each is presented here.

2.1 Edit Distance Measures

Edit distance measures quantify the difference between strings in terms of a sometimes weighted sum of the number of insertions, deletions, substitutions and/or transpositions required to yield the first string from the second. Standard Levenshtein distance [21] assigns a value of 1 to each insertion, deletion, and substitution. It can be converted to a similarity value by dividing the actual Levenshtein distance by the greatest possible Levenshtein distance for the given strings (i.e., the length of the longer string), and subtracting the resulting value from 1. Another common modification, Damerau-Levenshtein distance, additionally counts a transposition between adjacent characters as an edit operation [12]. Variations of this algorithm assign different weights to edits based on the type of operation, phonetic similarities between the sounds typically represented by the relevant characters, and other considerations [26].

2.2 Bag Distance

The *bag distance* algorithm enumerates every character in string *x* that cannot be uniquely matched with a character in string y, and vice versa; the maximum of the two values is the *bag distance.* Bag distance places an upper bound on Levenshtein distance, and has been proposed as a fast approximation thereof [2].

2.3 N-gram Measures

N-gram-based measures count the number of n-grams (substrings of length *n*) in common between the two strings being compared. Similarity is obtained either by dividing this count by the number of n-grams in the shorter string, the number in the longer string, or the average number in both strings; the resulting measures are referred to as the overlap coefficient, Jaccard index, and Dice coefficient, respectively [20, 26]. N-gram-like measures can also be calculated with skip-grams [19] or 'open bigrams' [31], bigrams that skip one or more characters. For example, *cow* and *caw* have in common the open bigram *c_w*. We calculated n-gram and positional n-gram measures for unigrams, bigrams, and trigrams, as well as skip-grams of gap size 0, 1, and 2.

2.4 Longest Common Substring Measures

The "longest common substring" (LCS) method [13], developed to match patient records in a clinical setting, begins by determining the longest common substring between the strings being compared. Its length is recorded, the substring is removed from both strings, and the process is repeated until no common substring remains that exceeds some minimum length *L*. As in [5], $L = 2$ and $L = 3$ were considered, and the result was divided by the length of the shorter string, the length of the longer string, and the average length to yield three distinct similarity measures.

2.5 Jaro Variants

The Jaro algorithm [18, 27] defines 'matching characters' as characters in strings s_1 and s_2 that are (1) the same, and (2) whose indices are no farther than. If *m* is the number of matching characters between strings *x* and *y* and *t* is the number of transpositions, the Jaro distance is defined as $\frac{1}{3}\left(\frac{m}{|x|}+\frac{m}{|y|}+\frac{m-t}{m}\right)$ when *m* is greater than 0, and 0 otherwise. A common variant generally referred to as Jaro-Winkler [27, 32] takes into account the fact that spelling errors are less likely to occur at the beginnings of names than elsewhere, and essentially assigns a higher weight to initial characters. [5] points out that while this is generally an improvement, it can be problematic if the strings to be matched contain multiple words that are differently ordered (e.g. "Sakhalin Island", "Island of Sakhalin"). To this end, he introduces the variants *sorted Winkler* and *permuted Winkler*. The former algorithm sorts both strings before calculating their similarity, while the latter calculates the similarity over all possible permutations and returns the maximum value.

2.6 Normalized Compression Distance

Normalized compression distance (NCD) [7] is a similarity metric that approximates *normalized information distance*, an (uncomputable) information-theoretic measure that quantifies the length of the shortest program *p* that computes string *x* from string *y*. NCD has been used to approximate the semantic and orthographic similarity between single words and documents of different languages [5, 8], as well as to quantify the similarity among texts suspected to be written by the same author [3, 30]. It is defined as

$$\frac{|C(x,y)| - \min(|C(x)|, |C(y)|)}{\max(|C(x)|, |C(y)|)}$$

where *C* is a compressor such as zlib or bz2.

2.7 Phone-Sensitive Measures

Phonetic encoding techniques such as Soundex are outside the scope of this investigation because measures based on phonetic encoding are lacking for most non-English languages, and the phonetic equivalencies they encode do not hold for some romanization systems. Nevertheless, we did include the three measures of string similarity evaluated by [5] that are sensitive to similarities between phonemes that frequently map onto particular letters ('t'/'d', 'g'/'k', etc.)

2.7.1 Smith-Waterman Distance

Like edit distance measures, the Smith-Waterman algorithm [24, 29] determines the sequence of operations necessary to transform one string to another, but ascribes lesser weights to transformations between similar-sounding characters and employs specialized logic for handling alignment gaps, i.e., there is a 'gap start' penalty corresponding to the beginning of a string of unmatched characters, and a separate 'gap continuation' penalty for its continuation. As with the n-gram measures, similarity may be obtained by scaling the resulting value by the length of the shorter string, the length of the longer string, or the average length.

2.7.2 Editex

Editex [34] straightforwardly combines Levenshtein distance with 'letter groups' (*aeiouy*, *bp*, *ckq*, etc.) such that letters in a similar group frequently correspond to similar phonemes. As in Levenshtein distance, the minimal number of insertions, deletions, and replacements necessary to transform one string to another is computed, but edits that replace a letter with another letter from a

different group are weighted more heavily, and deletions of letters that are frequently silent (*h* and *w*) are weighted less heavily than other deletions.

2.7.3 Syllable Alignment

Syllable alignment pattern searching [15] treats syllables rather than characters as the basic unit of analysis. Syllable locations are estimated by first applying numerous context-sensitive transformations that convert characters to character groups, yielding a series of letter groups for each string [14]. Syllable locations are estimated, and the minimum cost necessary to transform one string of character groups to the other using any of seven weighted operations (some on character groups, some on syllables) is determined.

3. STUDY

3.1 Data Preparation

Country files for China, France, Germany, Italy, Japan, Mexico, Saudi Arabia, Spain, Taiwan, the United Kingdom, and Yemen were downloaded from the National Geospatial-Intelligence Agency (NGA) GEOnet Names Server. These files consist of place names indexed to unique feature identifiers that uniquely identify the referent of the name. For example, the three alternate names of the *Tingtze River* mentioned earlier all share the same unique feature identifier (i.e., the same spatial referent). Although the country files do include some exonyms, the vast majority of the toponyms in each country file were romanizations of local place names in local languages. Place names were filtered to exclude entries rendered in non-Roman scripts, as some string similarity measures to be tested (e.g., syllable alignment) were not designed for non-Latin strings, and romanized equivalents were available for all toponyms.

Of all unique feature identifiers that were described by more than one name, 2,000 were randomly selected for each country and formed the source set S_1 for that country. The union of the sets of alternate names for the toponyms in S_1 formed the query set Q_1. No two names in S_1 shared the same referent.

3.2 Procedure

3.2.1 Evaluation of String Similarity Methods

Each name q in the query set Q_1 was considered a successful match if the name in S_1 with the highest similarity to q (according to the string similarity measure under evaluation) was also the unique element in S_1 with the same referent as q. Given that the task consisted of finding the element of the source set that shared the same referent as the element of the query set, accuracy was quantified as the proportion of elements of the query set that yielded successful matches. The datasets created from each of these 11 country files were evaluated on each of the 21 string similarity measures discussed in Section 2, using the implementations of these algorithms available in the open-source data linkage system Febrl [6]. In cases in which the parameters used in [5] were ambiguous, Febrl's default values were used. Finally, to clarify how much of the differences in performance between country files were due to country/language-specific differences (as opposed to random variation among datasets), the entire experiment was replicated with new source and query sets generated from each country file in such a way that no element of the original source set S_1 corresponded to the same referent as any element of the new source set S_2.

3.2.2 Exploratory Data Analysis

Although optimal measures may vary considerably by dataset, regularities may exist that allow the researcher to select measures that are appropriate to a particular language or country. Multidimensional scaling, factor analysis, and hierarchical clustering were used to explore these regularities. For each dataset, vectors of the accuracy of each method on that dataset were computed ("dataset vectors"). Likewise, for each method, vectors of the accuracy of that method on each dataset were computed ("method vectors"). SPSS 20 was used to calculate Euclidean distances between each pair of z-scored dataset vectors and to conduct three exploratory analyses on the resulting distance matrix: multidimensional scaling, hierarchical clustering, and factor analysis. Following the methodology of Maki & Buchanan [22], average linkage was used for hierarchical clustering, factors were extracted using the method of unweighted least squares, and a direct oblimin rotation was applied. This oblique rotation permits us to obtain a loading structure such that each variable loads primarily on a single factor, but in such a way that preserves information about correlations among factors, yielding what is likely to be a more accurate solution [10]. These analyses were initially conducted on dataset vectors to explore regularities among datasets, but were also conducted on method vectors to explore regularities among string similarity methods.

4. RESULTS AND DISCUSSION

Table 1 highlights the best-performing (most accurate) measures, while the Appendix lists precision and recall of each. As this was a matching task (1 item retrieved per element of S) rather than a retrieval task (retrieving all elements exceeding some threshold), precision and accuracy are identical. We observed substantial variation in the methods that worked best on datasets corresponding to different countries, i.e., the top-performing algorithm on the China and Japan datasets was among the worst-performing algorithms on the Spain and Mexico datasets.

For each country, a replication was conducted on a disjoint dataset derived from the same country file. For each country, the results of the replication were extremely similar to the results of the original analysis, with the set of the three best-performing measures for S_2 identical to the set of the three best-performing measures for S_1 for all countries.

While the greatest similarities were between datasets of place names corresponding to the same country, there was also a high degree of consistency in measure performance among countries with similar dominant languages. For example, the set of the five best-performing algorithms was identical for Yemen and Saudi Arabia. Likewise, the set of the five best-performing algorithms was identical for Mexico, Italy, and Spain.

For the 22 dataset vectors, results of the multidimensional scaling and factor analysis are illustrated in Figure 1 and Table 2, respectively. Each of these analyses showed that the same algorithms tended to work best on each of the two (non-overlapping) datasets drawn from each country file. This differs from prior findings that string similarity algorithms' performance tends to be dataset-specific [5, 9], and may be due to the similarity and homogeneity of toponyms from a single country (relative to the heterogeneity of the datasets investigated by Christen and Cohen et al. [5, 9]). In the hierarchical clustering analysis, the three highest-level clusters corresponded to the datasets derived from the European, Middle Eastern, and Asian countries, respectively. Likewise, the method of unweighted least

Table 1. Best-performing measures by dataset.

Dataset	China (1, 2)	France (1)	France (2)	Germany (1)	Germany (2)	Italy (1, 2)
Best	Jaro-Winkler	Skip-grams	Skip-grams	Skip-grams	Skip-grams	Skip-grams
2nd best	Perm. Winkler	Trigrams	Trigrams	Trigrams	Trigrams	Trigrams
3rd best	Jaro	Bigrams	Bigrams	Bigrams	Bigrams	Smith-Wat.
4th best	Sort. Winkler	LCS (L = 3)	Sort. Winkler	LCS (L = 3)	Smith-Wat.	Bigrams
5th best	Skip-grams	S. Winkler	Smith-Wat.	Smith-Wat.	LCS (L = 3)	LCS (L = 3)

Dataset	Japan (1)	Japan (2)	Mexico (1, 2)	Saudi A. (1)	Saudi A. (2)	Spain (1, 2)
Best	Jaro-Winkler	Jaro-Winkler	Skip-grams	Syllable	Skip-grams	Trigrams
2nd best	Perm. Winkler	Perm. Winkler	Smith-Wat.	Skip-grams	Smith-Wat.	Skip-grams
3rd best	Pos. bigrams	Pos. bigrams	Trigrams	Smith-Wat.	Syllable	Smith-Wat.
4th best	Jaro	Bigrams	Bigrams	Editex	Editex	Bigrams
5th best	Bigrams	Skip-grams	LCS (L = 3)	Bigrams	Bigrams	LCS (L = 3)

Dataset	Taiwan (1)	Taiwan (2)	U. K. (1)	U. K. (2)	Yemen (1)	Yemen (2)
Best	Editex	Editex	Bigrams	Skip-grams	Syllable	Skip-grams
2nd best	Dam.-Lev.	Levenshtein	Skip-grams	Bigrams	Editex	Editex
3rd best	Levenshtein	Dam.-Lev.	Perm. Winkler	Perm. Winkler	Skip-grams	Syllable
4th best	Jaro	Jaro	Trigrams	Trigrams	Smith-Wat.	Smith-Wat.
5th best	Jaro-Winkler	Jaro-Winkler	LCS (L = 3)	Smith-Wat.	Bigrams	Bigrams

Note. When the five best algorithms and their rankings for two datasets are identical, datasets are combined into a single column.

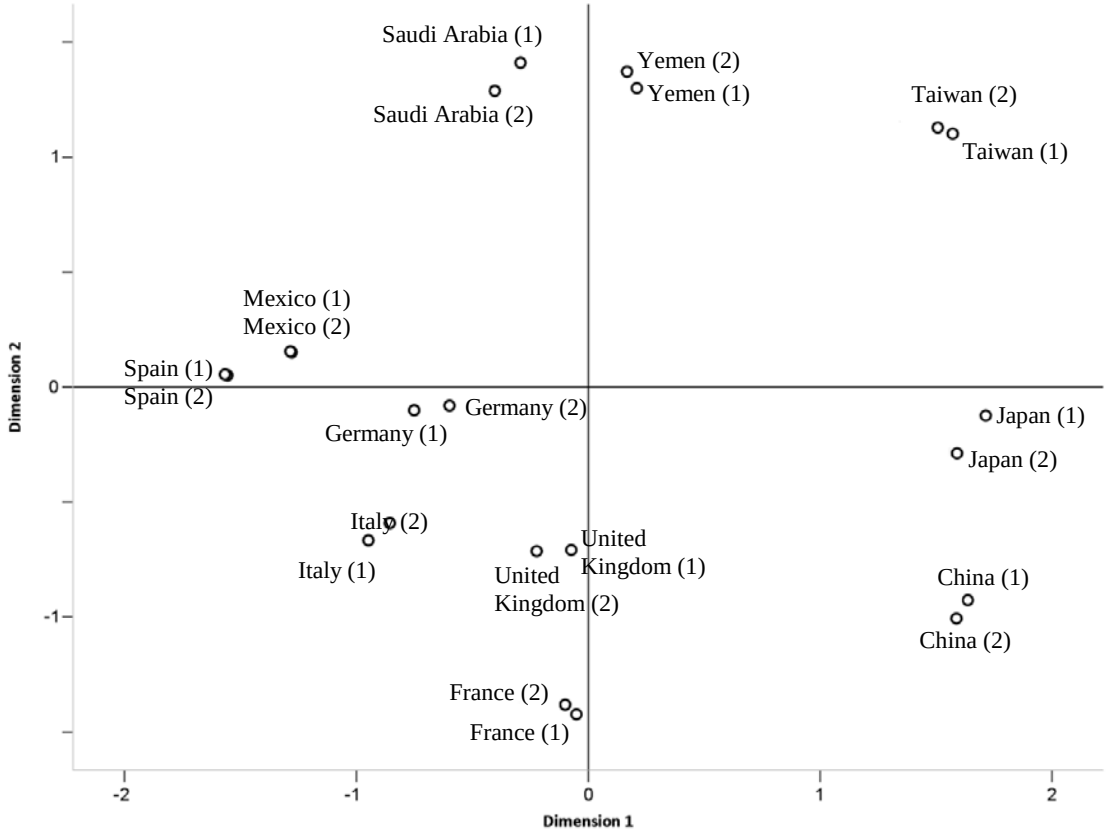

Figure 1. Multidimensional scaling plot of dataset vectors. Each point corresponds to one of the two disjoint datasets generated from each country file.

Table 2. Results of the factor analysis conducted on dataset vectors (pattern matrix); loadings having the highest weight for each variable are rendered in bold.

Dataset	Factor		
	1	2	3
China (1)	.302	**.832**	-.320
China (2)	.278	**.852**	-.298
France (1)	**.902**	.213	-.290
France (2)	**.899**	.217	-.298
Germany (1)	**.891**	.095	.153
Germany (2)	**.847**	.163	.152
Italy (1)	**1.008**	.011	-.086
Italy (2)	**.982**	.055	-.060
Japan (1)	.230	**.826**	.077
Japan (2)	.304	**.788**	.032
Mexico (1)	**.952**	-.053	.192
Mexico (2)	**.957**	-.057	.190
Saudi Arabia (1)	.389	.271	**.681**
Saudi Arabia (2)	.466	.230	**.643**
Spain (1)	**1.017**	-.247	.188
Spain (2)	**1.016**	-.243	.180
Taiwan (1)	-.203	**.992**	.243
Taiwan (2)	-.189	**.983**	.259
United Kingdom (1)	**.814**	.303	-.041
United Kingdom (2)	**.849**	.261	-.054
Yemen (1)	.360	.400	**.598**
Yemen (2)	.345	.383	**.630**

squares extracted exactly three factors, which explained 96% of the total variance and corresponded to the same three region-specific clusters (Table 2).

It seems reasonable to assume that these regularities are due to the similarities in the dominant languages among the European, Middle Eastern, and Asian countries in this dataset (Table 3). A close look at Table 1 suggests that for France, Germany, Italy, Mexico, Spain, and the United Kingdom, the three best-performing methods always include at least two of skip-grams, trigrams, and bigrams; Smith-Waterman distance and LCS with an *L* threshold of 3 are also strong contenders. In contrast, Jaro-Winkler and Permuted Winkler are the highest-performing algorithms for China and Japan, whereas three edit distance measures (Editex, Levenshtein distance, and Damerau-Levenshtein distance) perform best for Taiwan. For both datasets from both Arabic-speaking countries, the five best-performing methods were syllable alignment, Editex, Smith-Waterman distance, skip-grams, and bigrams, though not always in that order.

For the 21 method vectors, the method of unweighted least squares again extracted three factors which cumulatively explained 94% of the total variance. With a few exceptions (Smith-Waterman, sorted Winkler, and syllable alignment),

Table 3. Major languages spoken in countries investigated in the present study. Language lists are derived from [4].

Country	Languages
China	Standard Chinese or Mandarin, Cantonese, Shanghainese, Fuzhou, Hokkien-Taiwanese, Xiang, Gan, Hakka dialects, and others
France	French, regional dialects
Germany	German
Italy	Italian, German, French, Slovene
Japan	Japanese
Mexico	Spanish, indigenous languages (Mayan, Nahuatl, and others)
Saudi Arabia	Arabic
Spain	Castilian Spanish, Catalan, Galician, Basque
Taiwan	Mandarin Chinese, Taiwanese, Hakka dialects
United Kingdom	English, Scots, Scottish Gaelic, Welsh, Irish, Cornish
Yemen	Arabic

measures for which the unit of analysis included bigrams, open bigrams, or longer substrings tended to cluster together and load on Factor 1, measures for which the unit of analysis was a single character or operation (edit distance measures, bag distance, Editex, unigrams) loaded on Factor 2, and Jaro variants loaded on Factor 3. The three largest clusters in the hierarchical clustering analysis were similar, but with positional unigrams, positional bigrams, and positional trigrams clustering with the Jaro variants. These groupings seem to map onto the classes of algorithms that perform best on different country files: The best-performing algorithms on China and Japan were Jaro variants, edit distance performed best on Taiwan, and n-gram based measures worked best on countries with toponyms in Romance and Germanic languages. Although the phone-sensitive algorithms did not form their own cluster, it is interesting that these measures tended to perform well on the two Arabic-speaking countries, and with high variability on others. For example, the best-performing algorithm on one of the Yemen and Saudi Arabia datasets (syllable alignment) was among the five worst-performing algorithms on the China and France datasets, and achieved middling performance on several others. Similarly, Editex and Smith-Waterman distance performed well on Yemen and Saudi Arabia, but varied considerably with respect to their performance on other datasets.

4.1 Similarities and Differences Among Datasets

What properties of toponyms might cause some string similarity algorithms to perform better than others, and to differ so dramatically in their performance across languages? Although this study was not designed to answer this question, we can make some informed speculations by informally comparing the properties of each set of toponyms. Upon even a cursory

observation of each dataset, it is not surprising that methods that simply count the number of shared short substrings (bigrams, trigrams) tend to perform well. Variants that are abbreviations of longer forms of a place name (e.g., Ojo de Agua de Rosales / Ojo Rosales, Ojo de Agua) are ubiquitous. Pieces of the name may drop out from the beginning (Province of Ulster / Ulster), end (Abingdon-on-Thames / Abingdon), or center (Worcester and Birmingham Canal / Worcester Birmingham Canal) of a string, and can even be simultaneously added and dropped (County Borough of Wrexham / Wrexham Maelor). Abbreviations in the China, Japan, and Taiwan datasets exhibit more regularity; in these, abbreviations most frequently involve cropping the end of the string. In the set of Chinese toponyms with variants, for example, the most frequent tokens are *Zhen*, *Xiang*, *Xian*, *-hsien*, and *Qu*, all of which appear nearly always at the end of the string (when appearing as standalone tokens), and are frequently abbreviated or transformed. Correspondingly, Jaro-Winkler does well on these datasets, as it is the only one of the algorithms that intentionally penalizes differences more when they occur towards the beginning of the string.

Algorithms for toponym matching must be robust to abbreviations both large (Albertshausen / Albertshausen bei Bad Kissingen) and small (Lac d' Alles / Lac d' Allos), as well as transpositions (Aldwincle Saint Peter / Saint Peter Aldwinkle). All three kinds of transformations occur frequently across datasets. When large chunks of a toponym swap places, algorithms that count the number of character-based edits necessary to transform one string to another are overly pessimistic. Although this can be addressed with n-gram-based methods, n-grams are sensitive to minor disturbances: a single deletion (abcde→abde) will destroy three trigrams, and two bigrams. Skip-grams preserve the advantages of n-gram methods while turning a blind eye to pluralization, hyphenation, apostrophe placement, tokenization differences, vowel shifts, and other phenomena that frequently cause toponym variants to differ by only one or two characters.

We were initially surprised that the three phone-sensitive measures (Editex, Smith-Waterman, and syllable alignment) achieved such high performance on datasets consisting primarily of transliterated Arabic toponyms. However, it seems likely that the vast majority of variant toponyms in these datasets are due to phonetically similar transliterations of the same place name. For example, the following were observed in a random selection of ten toponym/variant pairs from the Yemen dataset: (Gebel Sadab / Jebel Satab), (Ra's Kathib / Ra's al Katib), (Bir Haiyirah / Bi'r Hayyirah), (Jau Mulais / Jaww Mulais), (Djol Bin Fadl / Jawl Bin Fadl), and (Wadi Balas / Wadi Bilas). In each case, phonetic differences are minimal, despite differences in spelling. This is also somewhat true of the China and Taiwan datasets, but to a lesser extent, and the phone-sensitive algorithms' assumptions about which letters represent similar speech sounds are not well-suited to the Hanyu Pinyin frequently used to transliterate Chinese place names. The Editex letter groups seem to fare somewhat better at matching names rendered in Tongyong Pinyin (frequently used in the Taiwan dataset) to alternate transliterations, although other factors are in play as well.

This discussion is not intended to provide a rigorous investigation of which differences between datasets lead to differences in algorithm performance, but may nonetheless serve as a starting place for future research. We close with some recommendations for integrating string similarity measures into GIR systems that are suggested by our results.

5. CONCLUSION

In many GIR use cases, users query a set of place names with search terms that include alternate spellings, transliterations, and variants that are similar but not identical to the place name associated with the desired record. Many string similarity algorithms can accomplish this 'fuzzy matching' task, but there has been no systematic investigation of whether different algorithms are appropriate to different kinds of toponyms. The present results demonstrate that there are at least some language-dependent regularities that hold across datasets, and form the basis of the following recommendations:

If possible, test several algorithms on a country-specific or language-specific dataset, and use the best-performing algorithm for future queries involving that dataset. In cases for which the country of the desired record is known, and enough records are available to compare the performance of various algorithms, it appears worthwhile to do so. Algorithms that perform very well on some datasets perform very poorly on others, but similar algorithms tend to do well on disjoint datasets drawn from the same country. In addition, similar algorithms appear to perform well for countries that share similar languages.

Some algorithms are strictly better than most others, irrespective of country. Averaged across all countries, the best-performing algorithms were skip-grams, bigrams, trigrams, Smith-Waterman, and LCS (L = 3), while the worst-performing algorithms were positional trigrams, unigrams, positional unigrams, bag distance, and NCD (bz2 compressor). Skip-grams were the best-performing algorithm on the plurality of datasets, and performed poorly on none. Skip-grams were always the best performing measure, with the following exceptions: They were beaten out by Jaro variants only on the China datasets, by trigrams only on the Spain datasets, and by bigrams only on one of the two UK datasets. On Japan, skip-grams were outperformed by Jaro variants, bigrams, and positional bigrams; by syllable alignment and/or Editex on half of the Saudi Arabia and Yemen datasets; and by Jaro variants, Editex, and edit distance measures on the Taiwan dataset. Therefore, skip-grams seem to be an excellent choice in the absence of other information about one's dataset, closely followed by bigrams.

There are superior alternatives to Levenshtein distance. Despite the ubiquity of Levenshtein distance as the method of choice in many toponym matching applications, we found that its average performance was only 11th out of the 21 algorithms investigated, and that it was outperformed by skip-grams and bigrams on all datasets except for Taiwan.

5.1 Future Directions

Given the excellent performance of skip-grams using the default settings in the Febrl software package, one promising next step would be to investigate whether other combinations of n-grams of various lengths and gap sizes perform even better. Interestingly, skip-grams have been investigated as the basis of a psychologically plausible model of word representation under the name "open bigrams" [31], and there are numerous competing word-form representations [11] that have not yet been investigated closely in the literature on computational linguistics or geography. This area is ripe for further study. In the meantime, we hope that the present investigation offers useful insights to researchers and practitioners interested in applying fuzzy matching techniques to place names from different countries and languages.

6. ACKNOWLEDGMENTS

This project was supported by a grant from the Intelligence Community Postdoctoral Research Fellowship Program through funding from the Office of the Director of National Intelligence.

7. REFERENCES

[1] Anastácio, I., Martins, B., and Calado, P. 2011. Supervised learning for linking named entities to knowledge base entries. In *Proceedings of Text Analysis Conference* (Gaithersburg, Maryland, November 14-15, 2011). KBP '11. National Institute of Standards and Technology, Gaithersburg, MD, n.p.

[2] Bartolini, I., Ciaccia, P., and Patella, M. 2002. String matching with metric trees using an approximate distance. In *String Processing & Information Retrieval (SPIRE), Lecture Notes in Computer Science*, 2476, 271-283, Lisbon, Portugal.

[3] Benedetto, D., Caglioti, E., and Loreto, V. 2002. Language trees and zipping. *Physical Review Letters*, 88, 4, 048702. DOI=10.1103/PhysRevLett.88.048702.

[4] Central Intelligence Agency. 2013. *The World Factbook 2013-14*. Washington, DC.

[5] Christen, P. 2006. A comparison of personal name matching: Techniques and practical issues. In *Data Mining Workshops, Sixth IEEE International Conference on Data Mining* (Hong Kong, December 18-22, 2006). IEEE, New York, 290-294.

[6] Christen, P., Churches, T., and Hegland, M. 2004. Febrl – a parallel open source data linkage system. In *Pacific Asia Knowledge Discovery and Data Mining* (Sydney, Australia, May 20-26, 2004). Springer, New York, 638-647.

[7] Cilibrasi, R. and Vitányi, P. M. B. 2005. Clustering by compression. *IEEE Transactions on Information Theory*, 51, 4, 1523-1545.

[8] Cilibrasi, R. and Vitányi, P. M. B. 2007. The Google Similarity Distance. *IEEE Transactions on Knowledge and Data Engineering*, 19, 370-383.

[9] Cohen, W. W., Ravikumar, P., and Fienberg, S. E. 2003. A comparison of string distance metrics for name-matching tasks. In *Proceedings of 2003 International Joint Conferences on Artificial Intelligence (IJCAI-03) Workshop on Information Integration on the Web* (Acapulco, Mexico, August 9-15, 2003). Morgan Kaufmann, San Francisco, 73-78.

[10] Costello, A. B., and Osborne, J. W. 2005. Best practices in exploratory factor analysis: Four recommendations for getting the most from your analysis. *Practical Assessment, Research & Evaluation*, 10, 7, 1-9.

[11] Cox, G. E., Kachergis, G., Recchia, G., and Jones, M. N. 2011. Toward a scalable holographic word-form representation. *Behavior Research Methods*, 43, 3, 602-615.

[12] Damerau, F. J. 1964. A technique for computer detection and correction of spelling errors. *Communications of the ACM*, 7, 3, 171-176.

[13] Friedman, C. and Sideli, R. 1992. Tolerating spelling errors during patient validation. *Computers and Biomedical Research*, 25, 486-509.

[14] Gadd, T. 1990. PHONIX: The algorithm. *Program: Automated Library and Information Systems*, 24, 4, 363-366.

[15] Gong, R. and Chan, T. K. 2006. Syllable alignment: A novel model for phonetic string search. *Institute of Electronics, Information and Communication Engineers (IEICE) Transactions on Information and Systems*, E89-D, 1, 332-339.

[16] Hastings, J. T. 2008. Automated conflation of digital gazetteer data. *International Journal of Geographical Information Science*, 22, 10, 1109-1127.

[17] Hastings, J. T. and Hill, L. L. 2002. Treatment of 'duplicates' in the Alexandria Digital Library gazetteer. In M.J. Egenhofer, and D.M. Mark (Eds.), *Geographic Information Science, Second International Conference (Extended Abstracts)* (September 25-28, Boulder, Colorado, 2002). Springer, New York, 64-65.

[18] Jaro, M. A. 1989. Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida. *Journal of the American Statistical Association*, 89, 414-420.

[19] Keskustalo, H., Pirkola, A., Visala, K., Leppanen, E., and Jarvelin, K. 2003. Non-adjacent digrams improve matching of cross-lingual spelling variants. In Proccedings of *String Processing & Information Retrieval (SPIRE)* (Manaus, Brazil, October 8-10, 2003). Springer, New York, 252-265.

[20] Lennon, M., Peirce, D. S., Tarry, B. D., and Willett, P. 1981. An evaluation of some conflation algorithms for information retrieval. *Journal of Information Science*, 3, 4, 177-183.

[21] Levenshtein, V. I. 1965. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10, 707-710.

[22] Maki, W. S., and Buchanan, E. 2008. Latent structure in measures of associative, semantic, and thematic knowledge. *Psychonomic Bulletin & Review*, 15, 3, 598-603.

[23] Martins, B. 2011. A supervised machine learning approach for duplicate detection over gazetteer records. In *Proceedings of the 4th International Conference on Geospatial Semantics* (Brest, France, May 12-13, 2011). Springer, Berlin Heidelberg, 34-51.

[24] Monge, A. E. and Elkan, C. P. 1996. The field-matching problem: Algorithm and applications. In *Proceedings of ACM SIGKDD* (Portland, Oregon, August 4-8, 1996). AAAI Press, Menlo Park, California, 267-270.

[25] Nadeau, D., and Sekine, S. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30, 1, 3-26.

[26] Navarro, G. 2001. A guided tour to approximate string matching. *ACM Computing Surveys*, 33, 1, 31-88.

[27] Porter, E. H. and Winkler, W. E. 1997. *Approximate String Comparison and Its Effect on an Advanced Record Linkage System*. Technical Report. US Bureau of the Census.

[28] Sehgal, V., Getoor, L., and Viechnicki, P. D. 2006, November. Entity resolution in geospatial data integration. In *Proceedings of the 14th Annual ACM International Symposium on Advances in Geographic Information Systems* (Arlington, Virginia, November 10-11, 2006). ACM, New York, NY, 83-90.

[29] Smith, T. F. and Waterman, M. S. 1981. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147, 195-197.

[30] Stamatatos, E. 2009. A survey of modern authorship attribution methods. *Journal of the American Society for information Science and Technology*, 60, 3, 538-556.

[31] Whitney, C. 2001. How the brain encodes the order of letters in a printed word: The SERIOL model and selective literature review. *Psychonomic Bulletin & Review*, 8, 221-243.

[32] Winkler, W. E. 2006. *Overview of Record Linkage and Current Research Directions.* Technical Report. US Bureau of the Census.

[33] Zheng, Y., Fen, X., Xie, X., Peng, S., & Fu, J. 2010. Detecting nearly duplicated records in location datasets. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in GIS* (San Jose, California, November 2-5, 2010). ACM, New York, NY, 137-143.

[34] Zobel, J. and Dart, P. 1996. Phonetic string matching: Lessons from information retrieval. In *Proceedings of ACM SIGIR* (Zurich, Switzerland, August 18-22, 1996). ACM, New York, NY, 166-172

APPENDIX

A. PRECISION AND RECALL

Because this was constructed as a matching task (i.e., select the best match in *Q* for each element of *S*) rather than as a retrieval task (i.e., select all matches from *Q* exceeding some threshold), the total number of retrieved items was 2,000 (1 item retrieved for each of the 2,000 names in *S*). As such, precision is equal to accuracy for this particular task. Chance precision—the precision that would be obtained by selecting a random item from *Q* for each element of *S*—is 1 divided by the size of *S* (1/2,000 = .0005). In most datasets, the vast majority of names in *S* are associated with only one alternate name in *Q*, causing mean recall to typically be only slightly lower than mean precision. Values in Table 4 represent mean precision and mean recall (respectively) on the two datasets from each particular country, and rows are sorted by mean precision.

Table 4. Mean precision/recall of each method on datasets from each country.

	China	France	Germany	Italy	Japan	Mexico	Saudi Arabia	Spain	Taiwan	United Kingdom	Yemen
Skip-grams	.26/.26	.74/.73	.75/.74	.63/.62	.52/.51	.66/.59	.52/.51	.66/.54	.51/.50	.70/.70	.58/.57
Bigrams	.25/.24	.72/.72	.75/.73	.61/.60	.52/.51	.63/.57	.50/.49	.63/.52	.47/.47	.70/.70	.56/.55
Trigrams	.25/.24	.73/.72	.75/.74	.62/.61	.51/.50	.64/.57	.47/.46	.66/.54	.39/.38	.68/.68	.54/.52
Smith-Wat.	.22/.21	.71/.71	.71/.69	.61/.60	.43/.42	.65/.58	.52/.51	.63/.52	.41/.40	.67/.66	.57/.55
LCS (L = 3)	.20/.20	.70/.70	.71/.69	.60/.59	.47/.46	.62/.56	.47/.46	.61/.50	.32/.32	.67/.66	.51/.50
Perm. Winkler	.31/.30	.70/.69	.63/.62	.56/.55	.55/.54	.52/.47	.41/.40	.40/.32	.53/.52	.69/.69	.49/.47
LCS (L = 2)	.20/.19	.63/.63	.68/.67	.54/.53	.47/.46	.57/.51	.48/.47	.54/.44	.44/.43	.65/.65	.52/.50
Jaro	.30/.30	.68/.68	.60/.59	.48/.47	.52/.50	.47/.42	.44/.43	.39/.31	.58/.57	.62/.61	.52/.50
Sort. Winkler	.29/.29	.71/.71	.61/.60	.54/.53	.49/.48	.55/.50	.40/.39	.44/.36	.39/.39	.61/.61	.46/.45
Jaro-Winkler	.32/.31	.65/.65	.59/.58	.46/.46	.56/.55	.43/.39	.40/.39	.36/.29	.56/.55	.61/.61	.48/.46
Levenshtein	.22/.21	.51/.51	.61/.60	.45/.45	.51/.50	.49/.44	.49/.48	.44/.36	.59/.58	.55/.55	.55/.54
Dam.-Leven.	.22/.21	.51/.51	.61/.60	.45/.45	.51/.49	.49/.44	.49/.48	.44/.36	.59/.58	.55/.55	.56/.54
Editex	.22/.22	.48/.48	.59/.58	.43/.43	.50/.49	.49/.44	.51/.49	.42/.35	.60/.59	.55/.54	.58/.57
Syllable	.19/.19	.48/.47	.60/.59	.47/.46	.49/.48	.55/.49	.52/.51	.48/.39	.39/.38	.58/.57	.58/.56
Pos. bigrams	.25/.24	.62/.61	.55/.54	.41/.40	.53/.52	.35/.31	.41/.40	.32/.26	.31/.30	.57/.57	.50/.49
NCD (zlib)	.12/.12	.60/.59	.56/.55	.50/.50	.31/.31	.52/.47	.42/.41	.49/.40	.17/.17	.55/.55	.47/.46
Pos. trigrams	.24/.23	.61/.61	.54/.53	.41/.41	.52/.51	.35/.31	.38/.38	.32/.26	.26/.26	.56/.56	.47/.46
Unigrams	.15/.15	.44/.43	.56/.55	.39/.38	.38/.37	.46/.41	.43/.42	.38/.31	.38/.38	.52/.52	.48/.47
Pos. unigrams	.21/.21	.52/.52	.51/.50	.36/.35	.49/.48	.32/.29	.41/.40	.28/.23	.28/.28	.53/.53	.50/.48
Bag distance	.11/.11	.36/.36	.47/.46	.32/.32	.36/.35	.37/.33	.39/.38	.28/.23	.27/.27	.43/.43	.46/.45
NCD (bz2)	.05/.05	.25/.25	.31/.30	.24/.24	.16/.16	.23/.20	.22/.22	.21/.17	.06/.06	.29/.29	.24/.23

Note. Mean precision is on the left of each slash mark, while mean recall is on the right.