USGS
science for a changing world

Eastern Geographic Science Center

# Fast, Inclusive Searches for Geographic Names Using Digraphs

Chapter 1 of
**Book 7, Automated Data Processing and Computations**
Section A, Algorithms

Techniques and Methods 7–A1

**U.S. Department of the Interior**
**U.S. Geological Survey**

# Fast, Inclusive Searches for Geographic Names Using Digraphs

By David I. Donato

Chapter 1 of
**Book 7, Automated Data Processing and Computations**
Section A, Algorithms

Techniques and Methods 7–A1

**U.S. Department of the Interior**
**U.S. Geological Survey**

**U.S. Department of the Interior**
DIRK KEMPTHORNE, Secretary

**U.S. Geological Survey**
Mark D. Myers, Director

U.S. Geological Survey, Reston, Virginia: 2008

For product and ordering information:
World Wide Web: http://www.usgs.gov/pubprod
Telephone: 1–888–ASK–USGS

For more information on the USGS—the Federal source for science about the Earth, its natural and living resources, natural hazards, and the environment:
World Wide Web: http://www.usgs.gov
Telephone: 1–888–ASK–USGS



Suggested citation:
Donato, D.I., 2008, Fast, inclusive searches for geographic names using digraphs: U.S. Geological Survey, Techniques and Methods, book 7, chap. A1, 6 p.

# Contents



## Figures



## Tables

# Fast, Inclusive Searches for Geographic Names Using Digraphs

By David I. Donato

## Abstract

An algorithm specifies how to quickly identify names that approximately match any specified name when searching a list or database of geographic names. Based on comparisons of the digraphs (ordered letter pairs) contained in geographic names, this algorithmic technique identifies approximately matching names by applying an artificial but useful measure of name similarity. A digraph index enables computer name searches that are carried out using this technique to be fast enough for deployment in a Web application. This technique, which is a member of the class of n-gram algorithms, is related to, but distinct from, the soundex, PHONIX, and metaphone phonetic algorithms. Despite this technique's tendency to return some counterintuitive approximate matches, it is an effective aid for fast, inclusive searches for geographic names when the exact name sought, or its correct spelling, is unknown.

## Introduction

Sometimes users need to search a computer database of geographic names[1] for near matches to specified names. Although finding near matches for specified names in a database is more complex than finding exact matches, the technique based on digraphs (ordered letter pairs) presented in this article makes it possible to develop a straightforward computer application for finding near matches. The digraph-based search technique can be implemented by a deterministic algorithm that processes names purely as abstract character strings without regard for their meaning or cultural associations.

Throughout this article the term **digraph** means any ordered pair of adjacent letters or characters extracted from a written geographic name. By contrast, in phonetics the term **digraph** refers only to diphthongs and other special pairs of letters (such as "ae" or "th" or "ph") that represent particular sounds. The term **digraph** is used here in its broadest sense, not in the restricted sense of phonetics.

This article describes the technique of digraph-based searches for geographic names, and provides an illustrative example of an algorithm that implements the technique. The description of the technique is suitable for software developers who wish to implement fast, inclusive searches of lists of geographic names using this technique.

Searches for near (or approximate) matches to specified names are called **inclusive** in this article. Though these name searches might be informally described as **fuzzy** searches, this usage would not be technically correct; the digraph-based search technique described here is not based on fuzzy logic or fuzzy set theory (nor on any methods generally considered to be within the purview of the field of artificial intelligence).

This digraph-based technique was developed in 1998 when the U.S. Geological Survey (USGS) began offering lists of topographic maps through an interactive World Wide Web site. The problem faced then was to design a map-name search facility that could rapidly retrieve the following:

1. all names containing an exact match for the user's **search-for name**[2],
2. all names a typical user would regard as similar to the search-for name,
3. the correct name when the user's search-for name is misspelled,
4. near matches not necessarily beginning with the same letter as the user's search-for name, and
5. no more names than can easily be perused in 15 to 20 seconds.

This requirement for fast and inclusive—but not promiscuous—searches of large lists of map names for near matches led to the development of digraph-based searches. The implementation was fast enough to be deployed on the World Wide Web for simultaneous use by multiple users with a typical response time of less than 5 seconds. Although the original implementation applied only to map names, the technique can be applied to any collection of geographic names represented by alphabetic writing.

[1] Geographic names include names of places, names of governmental subdivisions, and names of other kinds of geographic features (U.S. Board on Geographic Names, 2007).

[2] The **search-for name** is the geographic name provided by a user to initiate a search for names that are near (that is, approximate) matches.

## Digraphs and Similarity

A collection of geographic names certainly cannot be divided into all groups of similar names based solely on comparison of the letters and digraphs occurring in the names. This is because human intelligence determines the similarity of a list of two or more different geographic names by applying knowledge not implied by mere orthography. A subjective human judgment as to the similarity or dissimilarity of names involves an interplay of general knowledge, personal experience, intention, mood, and context. So human judgment finds similarities along many axes and identifies a much richer collection of similarity groups than can be identified using any deterministic procedure that processes names as abstract character strings.

Despite their admitted limitations, however, name comparisons based on letters and digraphs can be useful—imperfect but useful. Comparisons do not have to emulate the richness of human judgment in order to help users find the names they are seeking; they only need to mimic human choices along a single but significant axis of comparison. Comparisons made among letters and digraphs capture similarities based on the representation of sounds in accordance with the conventions of English orthography.

Although the alphabet used in written English does not convey pronunciation unambiguously, the spelling of words is firmly based on a finite set of associations of sounds with letters and letter combinations. Since English orthography is not phonetic, the relation of letters and sounds is not one-to-one. The number of sounds distinguishable by native speakers of English (the number of phonemes in the English language) far exceeds the 26 letters of the English alphabet; consequently, English orthography entails the use of numerous letter combinations to represent these many sounds. The order of letters in digraphs and other letter combinations is significant (for example, **ng** in "fishi**ng**" does not have the same effect on pronunciation as **gn** in "beni**gn**") and order is significant in spelling generally; therefore, a large number of digraphs shared between two words is a stronger indicator of similarity than just a large number of shared letters. Words that share many digraphs are likely to sound alike, and words that sound alike are more likely to be related semantically than words chosen at random. This explains why the introduction of digraph comparisons into an algorithm for determining name similarity informs and imbues the algorithm with a sensitivity to the order of the letters in words and the ability to capture a significant proportion of phonetic correspondences. Thus, digraphs are useful for identifying a limited but significant kind of similarity among names when searching for near matches.

## Using a Digraph Index for Fast, Inclusive Searches

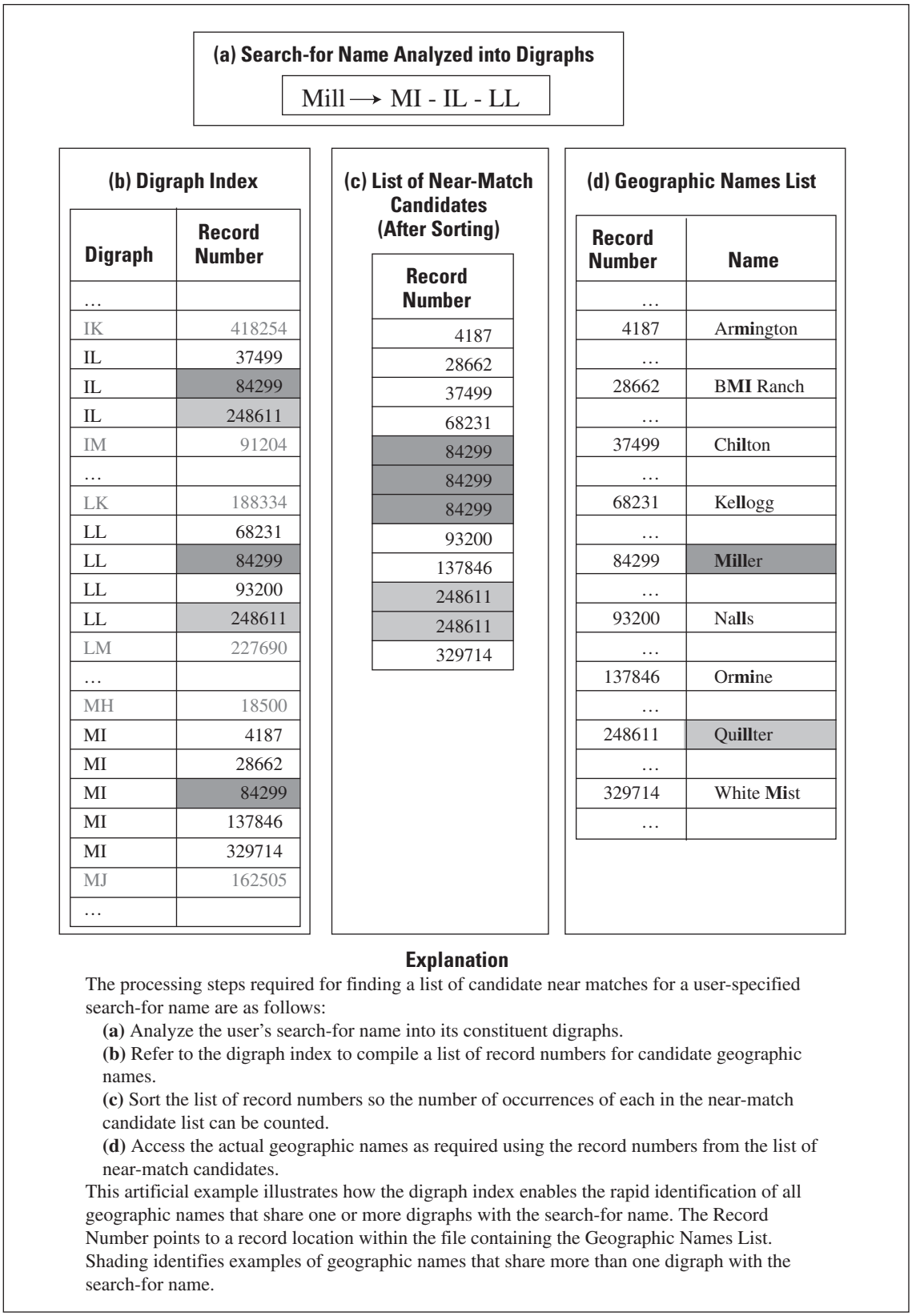
If a computer name-search application were to compare the user's search-for name one by one with each of thousands (or even millions) of names in a geographic-names database, the application would be slow and probably unusable as a Web-based application. Applications processed on a Web-server host must generally be fast and small in order to provide acceptably quick responses, even when there are several simultaneous users. Heavily used Web applications may receive several client requests per second during periods of peak usage.

The use of a digraph index enables a name-search application to restrict comparisons to a small subset of the database of names, thus speeding up processing. A digraph index lists in digraph sequence all digraphs found in the database of geographic names (see fig. 1). The entry for a particular digraph consists of a set of pointers to all names in the geographic-names database that include at least one occurrence of that digraph. Once the search-for name has itself been broken down into a list of its unique constituent digraphs, the digraph index can then be used to produce a candidate list containing only those names from the database that include at least one of the digraphs found in the search-for name. In most cases, several names in the candidate list will be referenced more than once (by different digraphs). After sorting the list of candidate names (thus, bringing together all occurrences of each particular candidate name), the number of occurrences of each candidate name can be counted. The number of occurrences of a candidate name will be the same as the number of unique digraphs a candidate name has in common with the search-for name. This number can be regarded as a first-approximation measure of a name's similarity to the search-for name, with larger numbers corresponding to stronger similarity.

In the following example, names have been analyzed into digraphs to provide a concrete illustration of how names may be compared using counts of their common digraphs:

- Millstone – mi il ll ls st to on ne
- Milltown – mi il ll lt to ow wn
- Millville – mi il ll lv vi il ll le
- Steam Mill – st te ea am mm mi il ll
- Airville – ai ir rv vi il ll le

"Millstone" has four digraphs in common with "Milltown" and "Steam Mill" but only three digraphs in common with "Millville". "Millville" and "Steam Mill" have three digraphs in common. "Airville" has four digraphs in common with "Millville" but only two in common with the other three names. When "Millstone" is the search-for name and the search algorithm screens out names with fewer than four unique digraphs in common, then "Milltown" and "Steam Mill" are in the candidate list, but "Millville" and "Airville" are not. When, however, the algorithm requires only three digraphs in common, then "Millville" joins the candidate list, though "Airville" still does not.




**Explanation**

The processing steps required for finding a list of candidate near matches for a user-specified search-for name are as follows:

**(a)** Analyze the user's search-for name into its constituent digraphs.

**(b)** Refer to the digraph index to compile a list of record numbers for candidate geographic names.

**(c)** Sort the list of record numbers so the number of occurrences of each in the near-match candidate list can be counted.

**(d)** Access the actual geographic names as required using the record numbers from the list of near-match candidates.

This artificial example illustrates how the digraph index enables the rapid identification of all geographic names that share one or more digraphs with the search-for name. The Record Number points to a record location within the file containing the Geographic Names List. Shading identifies examples of geographic names that share more than one digraph with the search-for name.

**Figure 1.** Example of compilation of a list of near-match candidates.

When a digraph index is used, the analysis of names in the database into digraphs only needs to be performed when the database is updated; thus, when a particular search request is processed, the only name that must be analyzed into digraphs is the search-for name. Since any name from the database that shares one or more digraphs with the search-for name will be included in the initial list of candidate names, and since the names from the database that have no digraphs in common with the search-for name are by assumption dissimilar to the search-for name, the digraph-based search algorithm does perform a virtual comparison of the search-for name with every name in the database, though without having to perform an actual (and slow) record-by-record comparison.

To make name searches with the digraph index as fast as possible, the following programming techniques should be used:

1. Set up the digraph index and the geographic-names list for direct access (access by record number) to avoid repetitive, sequential reading of these files. (Storing the list of geographic names in a flat file is preferable to setting up direct access to geographic names stored in a relational database.)

2. Provide an index to the digraph index. (For example, the first entry for the digraph **gh** in the digraph index might be record number 4,258. An index to the digraph index would allow a jump directly to record number 4,258 when the search-for term contains the digraph **gh** so that it would not be necessary to read any records in the digraph index other than those for **gh**.)

3. In the digraph index, refer to names in the list of geographic names by record number rather than by name, both to save space and to speed up processing.

4. Use fixed-length records in the digraph index and the file of geographic names to be searched so that file positions can be quickly and simply computed from the record length and the record number.

## An Example Algorithm for Fast, Inclusive Searches

In the online map-list application, the algorithm for selecting a set of near matches consisted of these steps:

1. Analyze the search-for name to identify and count all unique digraphs and to identify and count all unique letters.

2. Set the minimum number of common digraphs as the largest integer less than or equal to 65% of the number of unique digraphs in the search-for name, but not more than 6. (For example, if the search-for name contains 7 unique digraphs, then any near-match candidate will have at least 4 digraphs in common since 7 x 0.65 = 4.55. If the search-for name contains 11 unique digraphs, even though 11 x 0.65 = 7.15, a near-match candidate requires only 6 digraphs in common with the search-for name.)

3. Identify a set of candidate near matches from the database (containing the minimum number of digraphs in common with the search-for name) and then select or reject each candidate name in turn as follows:

    a. Convert both the search-for name and the candidate name to upper-case letters.

    b. Select the candidate name if it is a case-insensitive exact match for the search-for name.

    c. Select the candidate name if it contains the search-for name.

    d. Reject the candidate name if either the candidate name or the search-for name is at least twice as long as the other name.

    e. Select the candidate name if 70% of its letters (whether repeated or not) also occur in the search-for name; otherwise, reject the candidate name.

This algorithm operates with a number of numerical parameters that can easily be modified in computer code to render the algorithm either more or less inclusive. The table presents four examples of search-for names and the near and exact matches identified for them (from a list of approximately 64,000 map names) using this algorithm. These examples show that the algorithm finds some near matches that are consistent with human expectations, but also some that are jarringly inconsistent.

## Related Methods

The technique described in this article addresses a special case of a more general problem: how to search efficiently and effectively for names or character strings in a computer database when the name or string sought is uncertain. Some of the methods related to the digraph-search technique are as follows:

1. **soundex**—Soundex enables searches for personal surnames based on the surname's sound (or pronunciation) rather than on its spelling (U.S. National Archives and Records Administration, 2007). Soundex is especially useful in genealogical research, such as searching census records for ancestors who may have used variant spellings of the family name.

2. **PHONIX**—PHONIX is based on but extends and enhances soundex. PHONIX uses a more complex algorithm than soundex to provide a more sensitive

measure of phonetic similarity for personal names (Gadd, 1988, 1990).

3. **Metaphone and Double Metaphone**—Double Metaphone is the second version of Metaphone, a phonetic encoding algorithm in the same class of algorithms as soundex and PHONIX. Metaphone and Double Metaphone make use of detailed understanding of many of the conventions and idiosyncrasies of the spelling and pronunciation of surnames by native speakers of English (Philips, 2000).

4. **n-gram algorithms**—An n-gram (also known as a q-gram) is an ordered substring of length n. The concept of the n-gram is a generalization of the digraph concept: a digraph is a 2-gram, a trigraph is a 3-gram, and so on. Algorithms based on n-grams have been tested in such applications as indexing lexicons and processing approximate character strings (Zobel and Dart, 1995; Gravano and others, 2001).

The idea of similarity among character strings appears frequently in the literature of approximate, vague, or fuzzy searches. The ideas of distance between strings, or the relevance of strings, also appear repeatedly in the literature. Despite the literature's common thread of ideas about measuring similarity, distance, or relevance among names or other character strings, no single class of algorithms stands out as unequivocally superior in these applications. In the general case, however, when searching for surnames intermingled with other kinds of names and character strings, n-gram algorithms may be preferable to the phonetic algorithms, especially when n-gram search results are refined using supplementary phonetic measures (Pfeifer and others, 1995). The question of how to carry out approximate searches remains open for new developments, with each problem domain (such as finding near matches among geographic names) offering opportunities for development of domain-specific solutions.

**Table 1.** Four examples of search-for names and their near matches.

[These four examples show the near matches (and, in some cases, exact matches) found for the four search-for names from a list of approximately 64,000 map names]

| Beulah | Irving | Margarita | Xavier |
|---|---|---|---|
| | | | |
| Beaulieu | Arvin | Barataria | Avinger |
| Beulah | Avinger | Farisita | Beaverville |
| Beulah Belle Lake | Garvin | La Garita | Cavalier |
| Beulah Cemetery | Girvin | Margaret | Erieville |
| Beulah NE | Girvin NE | Margarita Peak | Mavie |
| Beulah NW | Girvin NW | Marietta | Prairieview |
| Beulahville | Irvine | Marmarth | Riverview |
| Beulaville | Irving | Raritan | Riviera |
| Eufaula | Irving College | Santa Margarita | Saint Xavier |
| Eula | Irvington | Santa Maria | Saint Xavier NE |
| Puu Ulaula | Kirvin | Sarita | San Xavier Mission |
| Taholah | Novinger | | San Xavier Mission SW |
| Tallulah | Ringling | | Sierraville |
| | Ringling NW | | Tavernier |
| | Viking | | Weaverville |
| | Vining | | |
| | Virgilina | | |
| | Virgin | | |
| | Virginia | | |

## Summary

Digraph-based searches of geographic names are effective at finding lists of names similar to user-specified search-for names, including names that do not begin with the same letter. Computer name-search applications can be made fast enough for interactive deployment on the Web through the use of a digraph index, and the algorithms used to implement digraph-based name searches can be readily tuned to be more or less inclusive (that is, to return more or fewer near matches) depending on user requirements. Digraph-based searches do, however, have the disadvantage that their lists of near matches frequently include names that seem to users jarringly unlike their search-for names. Despite this disadvantage, if users understand and accept that digraph-based searches return lists that may be incomplete and that may include some unlikely entries, digraph-based searches can provide fast, economical, and practical help in finding geographic names from a large list of names when users cannot provide the exact name to be found.

## References Cited

Gadd, T.N., 1988, 'Fisching fore werds': phonetic retrieval of written text in information systems: Program—automated library and information systems, v. 22, no. 3 (July 1988), p. 222–237.

Gadd, T.N., PHONIX—the algorithm: Program—automated library and information systems, v. 24, no. 4 (October 1990), p. 363–366.

Gravano, Luis, Ipeirotis, P.G., Jagadish, H.V., Koudas, Nick, Muthukrishnan, S., Pietarinen, Lauri, and Srivastava, Divesh, 2001, Using q-grams in a DBMS for approximate string processing: IEEE Data Engineering Bulletin, v. 24, no. 4 (December 2001), p. 28–34.

Pfeifer, Ulrich, Poersch, Thomas, and Fuhr, Norbert, 1995, Searching proper names in databases, in Proceedings of the Conference on Hypertext - Information Retrieval - Multimedia (HIM '95), *Synergieeffekte elektronischer Informationssysteme*: Konstanz, Germany, 1995, Universitätsverlag Konstanz, p. 259–275.

Philips, Lawrence, 2000, The Double Metaphone Search Algorithm: C/C++ Users Journal, v. 18, no. 6 (June 2000), p. 38–43.

U.S. Board on Geographic Names, 2007, U.S. Board on Geographic Names: U.S. Geological Survey Web site at *http://geonames.usgs.gov/index.html.* (Accessed August 16, 2007.)

U.S. National Archives and Records Administration, 2007, The soundex indexing system: The National Archives and Records Administration Web site at *http://www.archives.gov/genealogy/census/soundex.html*. (Accessed August 16, 2007.)

Zobel, Justin, and Dart, Philip, 1995, Finding approximate matches in large lexicons: Software—Practice and Experience, v. 25, no. 3 (March 1995), p. 331–345.