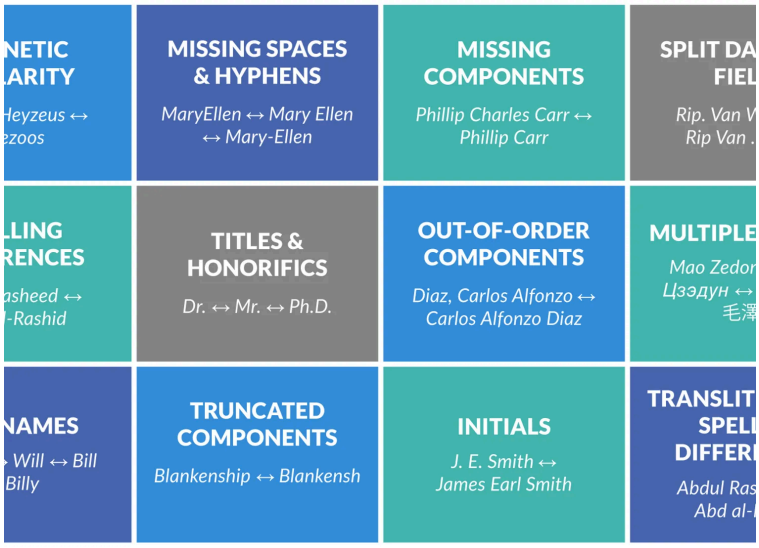
Babel Street Match

# Fuzzy Name Matching Techniques




## Methods of name matching and their strengths and weaknesses

In a structured database, names are often treated the same as metadata for some other field like an email, phone number, or an ID number. But what happens if you only have a name to lookup a record? This happens quite frequently since humans tend to prefer names to numbers and laws may prevent ID numbers from being created or shared.

When names are your only unifying data point, correctly matching similar names takes on a greater importance, however their variability and complexity make name matching a uniquely challenging task. Nicknames, translation errors, multiple spellings of the same name, and more all can result in missed matches. While there is an abundance of search tools on the market, name search is a different animal than document search, and requires a fundamentally different approach.

Different name matching methods are best suited to solve different name matching challenges. There are many ways to match names, but no one universal solution. The best name matching software uses a hybrid of multiple methods to address the maximum number of name variations:

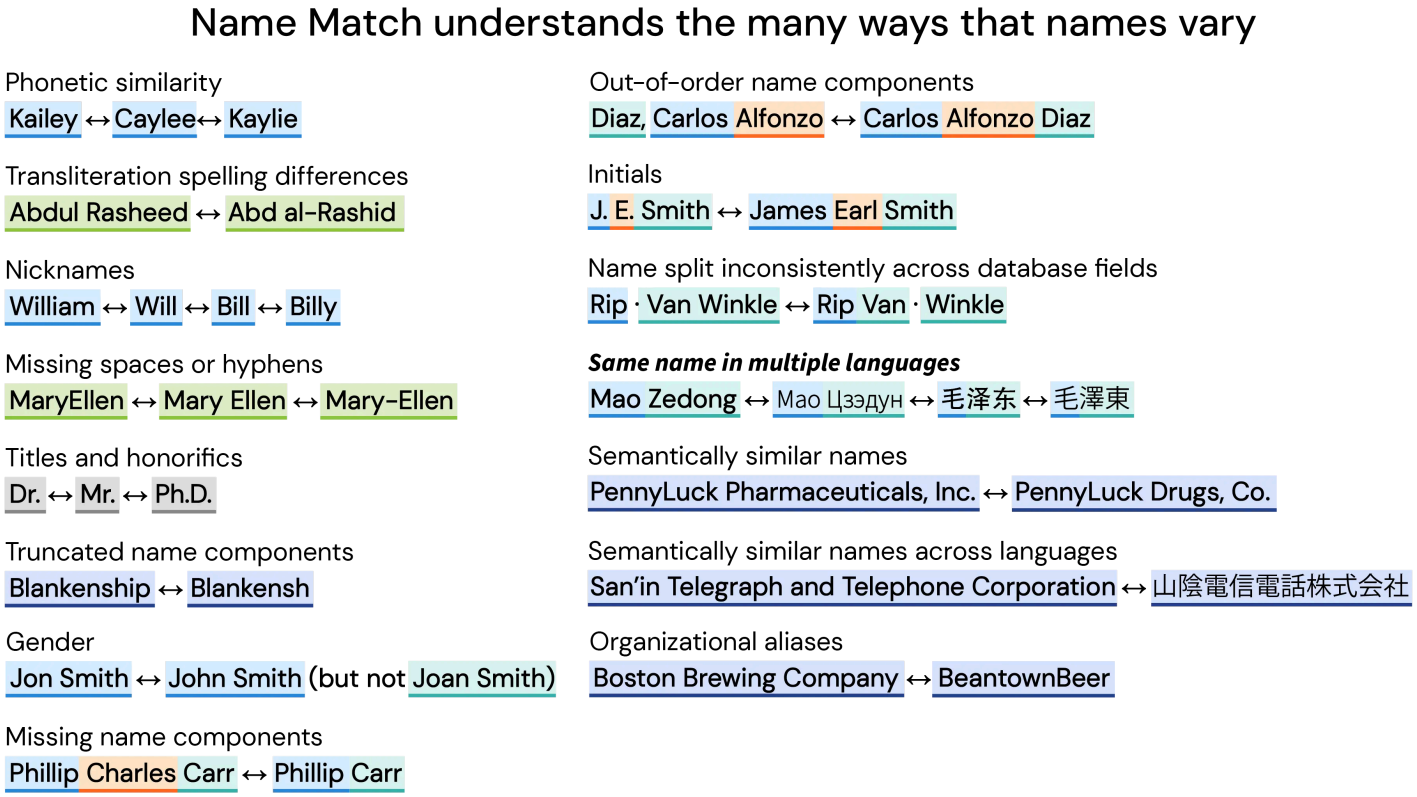
- Common key method
- List method

Skip to main content

od

method

consistent matching:




# Common key method

**Pros**: Fast execution, high recall

**Cons**: Mostly limited to Latin-based languages; transliterating non-Latin names reduces precision

These methods reduce names to a key or code based on their English pronunciation, such that similar sounding names share the same key. A well-known common key method is Soundex, patented in 1918. For example, Cyndi, Canada, Candy, Canty, Chant, Condie share the code C530.

Many methods take a similar approach to Soundex, including Metaphone and Double Metaphone. These methods use phonetic algorithms which turn similar sounding names into the same key, thus identifying similar names. Metaphone expands on Soundex with a wider set of English pronunciation rules and allowing for varying lengths of keys, whereas Soundex uses a fixed-length key.

Double Metaphone further refines the matching by returning both a primary and secondary code for each name, allowing for greater ambiguity. In addition, instead of being tied to English pronunciation of characters, it attempts to encompass pronunciations of other origins such as Slavic, Germanic, Celtic, Spanish, and Chinese.

names share a primary and secondary code of XMT indicates a degree of similarity between the names which Soundex perhaps overstates and which Metaphone misses.

| Name | Soundex Key |
|---|---|
| Smith | S530 |
| Schmidt | S530 |

| Name | Metaphone Key |
|---|---|
| Smith | SM0 |
| Schmidt | SXMTT |

While the common key method is fast to execute and has good recall, the precision suffers. Manual inspection of a few names reveals the precision issues. These names share the Soundex key H245: Haugland, Hagelin, Haslam, Heislen, Heslin, Hicklin, Highland, Hoagland.

Metaphone does a better job than Soundex, encoding the above names with different codes except for the very similar pairs Haugland/Hoagland and Heislen/Heslin.

| Name | Metaphone Key |
|---|---|
| Haugland | HKLNT |
| Hagelin | HJLN |
| Haslam | HSLM |
| Heislen | HSLN |
| Heslin | HSLN |
| Hicklin | HKLN |
| Highland | HFLNT |

| | |
|---|---|
| Hoagland | HKLNT |

For cases where name similarity is being scored against pairs of names in different scripts — for example Korean Hangul vs. English — the name must first be converted to Latin characters, which potentially introduces more errors to the comparison.

Particularly in languages such as Japanese where one character can have more than one correct pronunciations, converting first to the Latin script can introduce fatal mistakes. The common Japanese female name 洋子 can be correctly pronounced Yoko or Hiroko.

Transliteration of names (a mapping of characters or sounds in one script to another) produces many possible variations since sounds in one language have to be approximated. Variations introduced by transliteration increases the complexity of the already difficult task of matching names.

If عبد الرشيد is being evaluated against Abdal-Rachid, but the transliteration of عبد الرشيد produces Ar-Rashid, will the names come back as a match — as they should?

| Name | Soundex Key | Metaphone Key |
|---|---|---|
| Abdal-Rachid | A134 | ABTLRXT |
| Ar-Rashid | A623 | ARRXT |

One common key method, the Beider-Morse Phonetic Matching algorithm, does accept Russian in Cyrillic script and Hebrew in Hebrew script, but is otherwise Latin-bound.

# List method

**Pros**: Easy to maintain

**Cons**: Computationally intensive (read: expensive hardware needed to run against long lists of names quickly); Cannot handle names the system doesn't know about; Cannot handle names with missing/added spaces between components; Cannot handle names split between different fields; May require unacceptably long processing time for long, multi-component names (5+ components).

This method attempts to list all possible spelling variations of each name component and then looks for matching names from these lists of name variations. For example, one system produced 3,024 possible transliterations of this Arabic name "عبد الرشيد" since each separate name component alone has several variations. Here are the first five and last five variations.

5. Abdal-rachid

...

3020. ‘Abd-errshiyd

3021. ‘Abd-errchid

3022. ‘Abd-errchide

3023. ‘Abd-errcheed

3024. ‘abd-errchiyd

Trying to generate every possible name variation has a couple of obvious drawbacks. Name variations which are not in the list will not be found as matches, and perhaps an even greater issue is that of speed and size. Since multi-part names, particularly non-English names, generate an exponentially growing list of variations, searching through these lists takes time. Given a name with just three components and 20 possible variations per name, the number of possibilities is $20^3$ (=8,000), a very large search space for just one name. Now multiply it by the number of names on a watch list! There are further challenges with the list method – how do you score matches when one of your 8,000 query variants matches more than one name in the database? It is also difficult to handle other types of variation, like nicknames, initials, and titles, without expanding the search space even more.

A benefit of the list method is that it is simple to maintain. When a user complains about a missed match, it’s easily added to the name database. However, easy maintenance may not be enough to offset the decreased speed. For applications with that require high-throughput over millions of names, such as watchlist screening, anti-money laundering (AML), and know your customer (KYC), this approach is likely to be too slow or require a lot of expensive hardware.

# Edit distance method

**Pros**: Easy to implement

**Cons**: Limited to Latin-based languages; all swaps are weighted evenly, missing linguistic nuances

This approach looks at how many character changes it takes to get from one name to another. “Cindy” and “Cyndi” have an edit distance of 1 since the “i” and “y” are merely transposed, whereas “Catherine” and “Katharine” have an edit distance of 2 as the “C” turns into a “K” and the first “e” becomes an “a.”

Methods which look at the character-by-character distance between two names include Levenshtein distance, the Jaro–Winkler distance, and the Jaccard similarity coefficient. These approaches look at two factors (1) the number of similar characters and (2) the number of edit turn one name into the other — the operations being, insert, delete, and transpose.

substitution might more clearly indicate a similar name, as in "Catherine" vs. "Katherine." Further, a one-to-many character mapping is not possible, as in the case of the Arabic character "sheen" ش which is frequently mapped to "sh" in English.

And, just as with the common key method, a non-Latin script name must first be transliterated to Latin script before the comparison can be executed.

# Statistical similarity method

**Pros**: Matches across languages and scripts; offers greater precision

**Cons**: Slower performance; high barrier to entry as it requires training data and adjusting features etc.

A statistical approach takes hundreds, if not thousands, of matching name pairs and trains a model to recognize what two "similar names" look like so that the model can take two names and assign a similarity score.

A statistical model that has been trained on thousands of pairs of matching names offers high accuracy and the ability to directly match names written in different languages without first transliterating names to Latin script. This method has a higher barrier to entry, as collecting the matching names requires significant resources, but the accuracy may be well worth the effort. A downside is the slowness of execution. A system only using the statistical method to sift through millions of names to look for matches may be too slow to be feasible in high-transaction environments.

# Word embedding method for organization names

**Pros**: Makes semantic matches that a spelling-centric method would miss

**Cons**: Only relevant to organization name matching

Organization names differ from human names in that variations may include synonyms that look and sound entirely different than the target name. In these cases, two names referring to one company are semantically similar but phonetically different. For example, a human can quickly infer that corporation, company, and group are all similar words often found in an organization's name, but standard name matching techniques like the edit distance method would be unlikely to make the connection. In these cases, word embeddings can make the match.

Word embeddings are numerical vector representations of a word's semantic meaning. If two words or documents have a similar embedding, they are semantically similar. For example, the embeddings of "woman" and "girl" are close to one another in the vector space, meaning they are semantically similar. Contrastingly, the embeddings of "whale" and "philosophy" are far from one another because



ally related. Applied to organizations, the word embedding method recognizes
naceuticals and PennyLuck Drugs are most likely the same company.

similar vector space, text embeddings ensure that 国際 correctly matches with “nations.”


## A two-pass, hybrid method: The best of breed

Hybrid approaches backfill weakness in one approach with the strength of a different approach. For example, a hybrid approach may first use the common key method for high recall, and then put its results through the statistical method for greater precision.

In the first pass, the faster common key method and high recall winnow the candidate pool to a smaller set of likely matches. This step is particularly vital when a list has names in different languages, first transliterating them — typically to English — before assigning metaphones. The second pass over the culled down list then uses a high-precision statistical method to filter the highest scoring matches to the top, making fine-grained distinctions between different matches.

Compared to the common key method alone, accuracy is greatly improved by this hybrid method. Instead of being locked into a coarse comparison of derived keys (for better or worse), the second pass of the hybrid approach takes a fresh look at the original names in their original scripts before scoring their similarity.

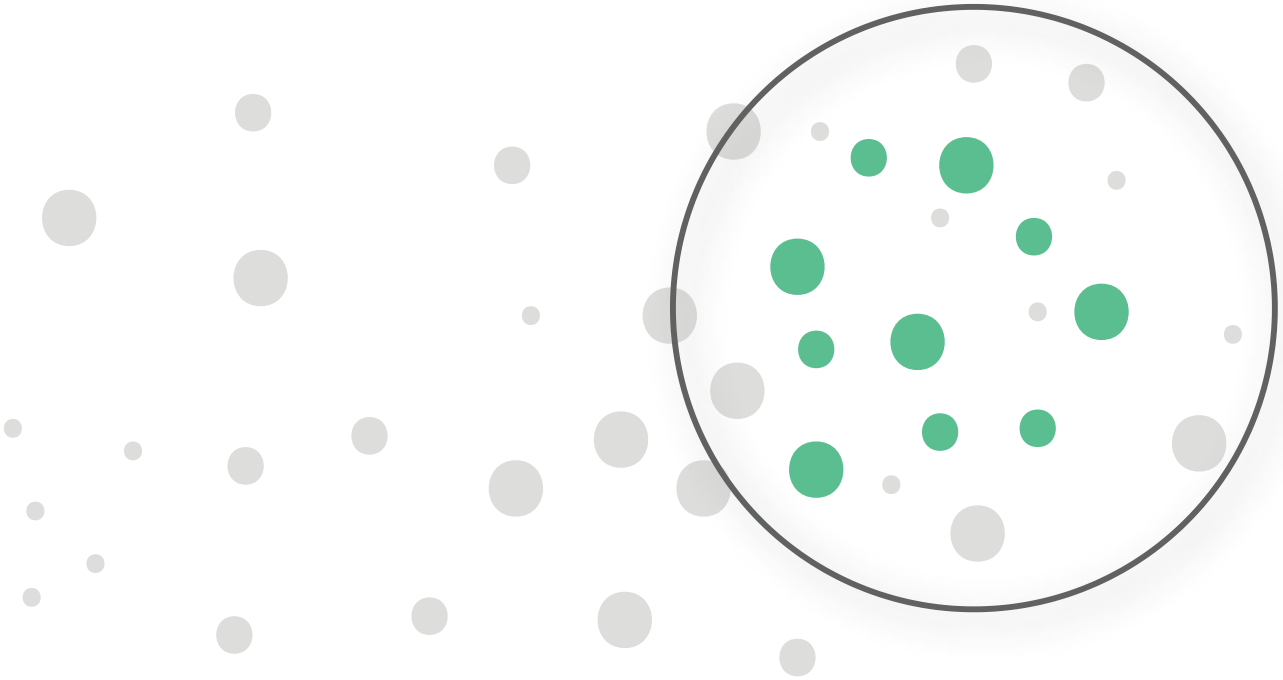
This hybrid method also avoids the weaknesses of the list approach by not relying on mass generation of name variations, but instead, uses (via the statistical model) the linguistic variations of



ge. This linguistic knowledge of name variations also gives the hybrid approach distance method, which cannot directly compare names in different scripts.

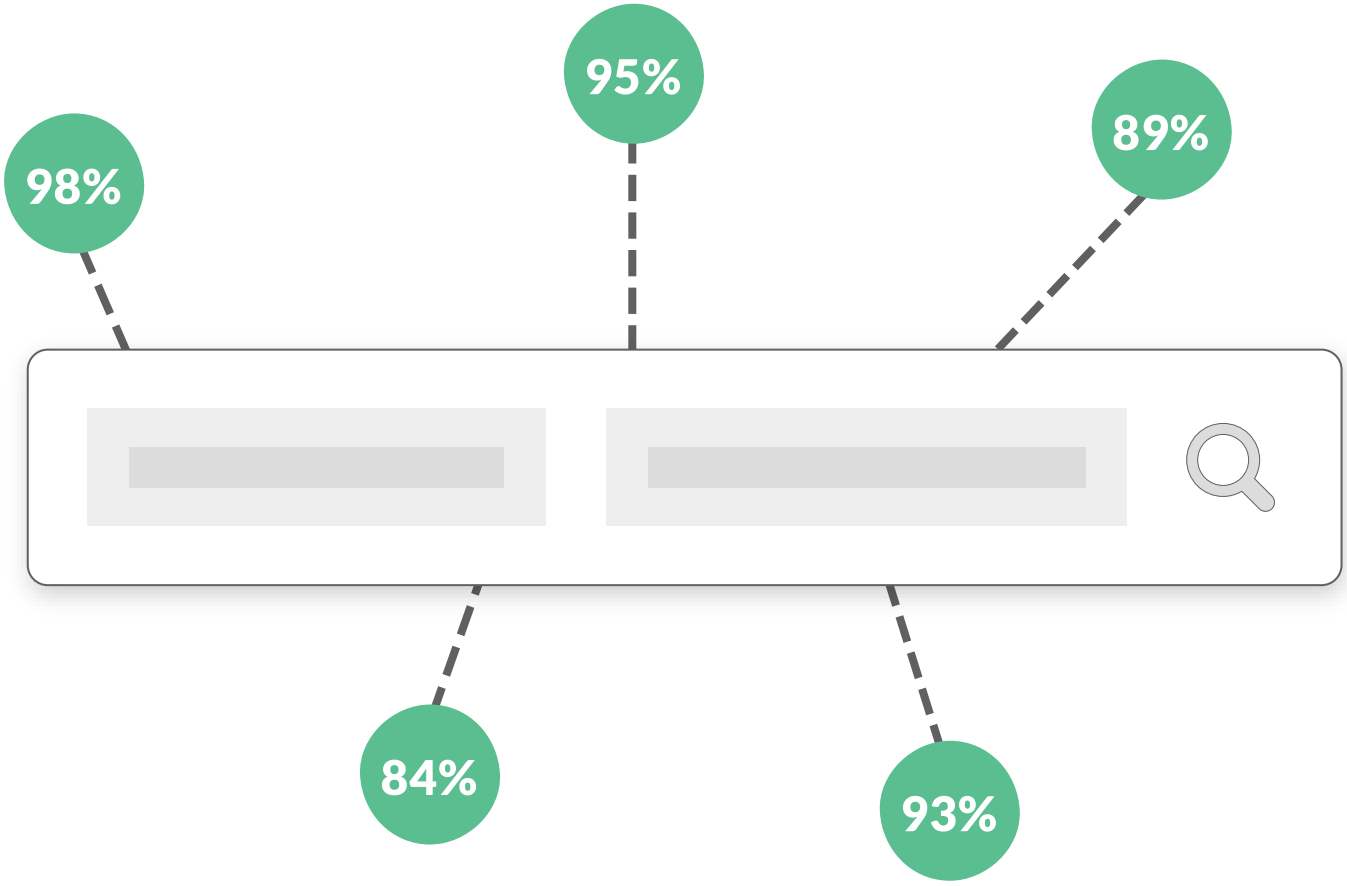
# QUERY

# FIRST PASS
## for recall

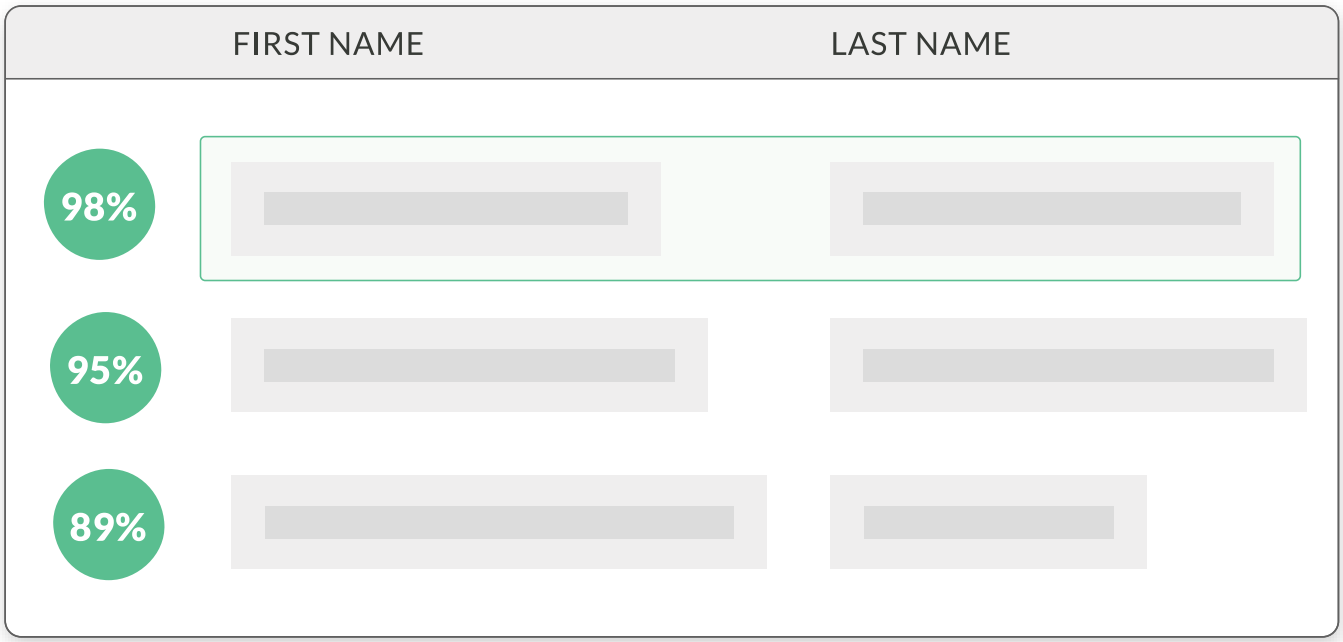


SECOND PASS

*for precision*



# MATCHES




Find out how to transform your data into actionable insights.

Schedule a Demo

**Stay Informed**

Sign up to receive the latest intel, news and updates from Babel Street.

Subscribe

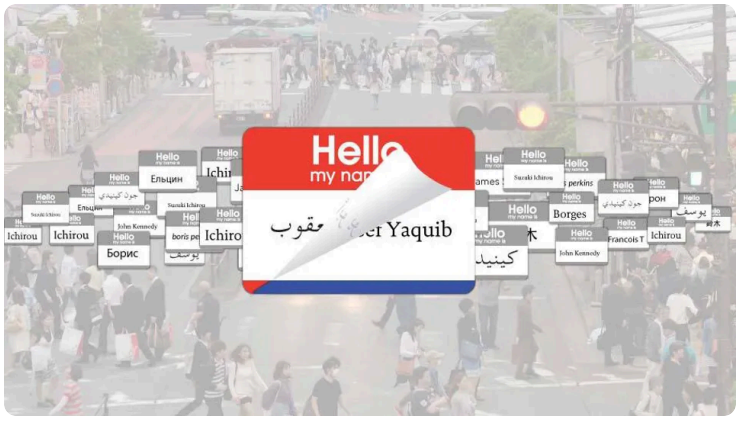
Skip to main content

LinkedIn

Email Article

Related Resources

# You may like




Blog Post

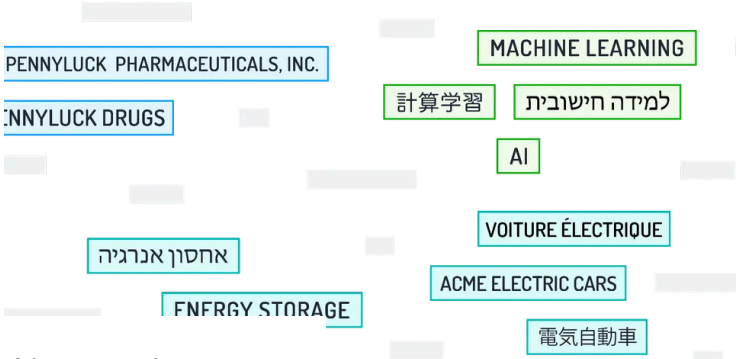
## Fuzzy Search Names in Elasticsearch

Aug 24, 2022

Blog Post

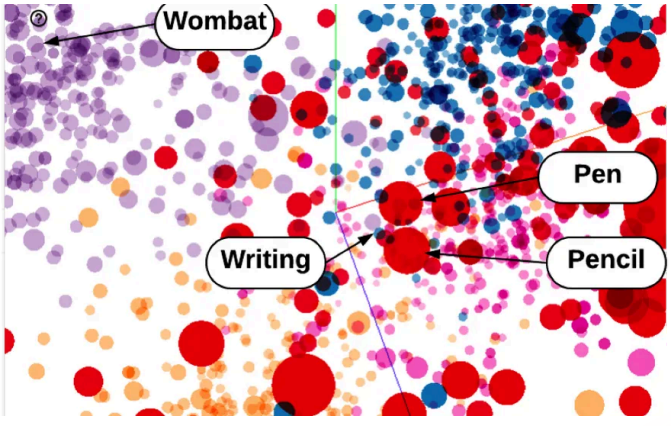
## What is Fuzzy Matching? Explore how Fuzzy Logic Boosts Name-matching Accuracy

Dec 15, 2022




Skip to main content

Blog Post




Blog Post

Aug 02, 2017

## Global Analysis, Part I

Mar 15, 2017