Understanding Babel Street Match and Intelligent Name Matching

BABELSTREET

Understanding Babel Street Match and Intelligent Name Matching

Introduction

Organizations today require increased global awareness and rapid data analysis to mitigate identity risks and threats. However, most organizations don’t have adequate resources to effectively address these threats — creating a Risk-Confidence Gap, or chasm between the increasing risk organizations face and confidence in their analysis and decision-making systems to mitigate those risks.

Organizations across government and industry are faced with the complexity of matching the names of individuals and businesses across diverse, multilingual sources in high-volume and high-velocity data environments. The consequences of name matching failure can be severe, especially in critical use cases, such as border crossings, counterterrorism, crimefighting, financial compliance, payment verification, and customer/patient identification and records linking.

By leveraging advanced AI and a deeply nuanced understanding of names, Babel Street Match enhances the clarity, accuracy, and transparency of identity matching. Our explainable AI significantly reduces false positives and makes the complex process of identity resolution transparent so you can see the ‘why’ and ‘how’ behind your AI-driven decisions.

More accurate name matching means fewer false positives (when a name is erroneously matched) and false negatives (when a match is missed) which increases operational efficiency and reduces risk exposure.

This technical brief explores the challenges of achieving accurate name matching and how Babel Street Match intelligently fuzzy matches names, addresses and dates across languages and cultures through a patented two-pass process.

Name matching is not like keyword search

Searching for a name in a list of names or across documents is not the same as a keyword search. Since new names are being created constantly, there is no single correct spelling for a name and no set of rules that encompasses how names are formed. They are variable across languages, cultures, ethnicities, and more.

Furthermore, unlike standard text search, where a typo can be overcome by looking at the frequency or rarity of words, each letter in a name has an outsized importance. "Cyndy" and "Cindy" are equivalent since it can't be assumed that "Cyndy" is a typo — it might be a real name.

Fuzzy name matching is a highly specialized field because names vary in myriad ways and good fuzzy name matching must simultaneously consider all those issues and arrive at a final match calculation. Here are some of the variations that Babel Street Match considers.

Phonetic similarity
Kailey ↔ Caylee↔ Kaylie

Transliteration spelling differences
Abdul Rasheed ↔ Abd al-Rashid

Nicknames
William ↔ Will ↔ Bill ↔ Billy

Missing spaces or hyphens
MaryEllen ↔ Mary Ellen ↔ Mary-Ellen

Titles and honorifics
Dr. ↔ Mr. ↔ Ph.D.

Truncated name components
Blankenship ↔ Blankensh

Gender
Jon Smith ↔ John Smith (but not Joan Smith)

Missing name components
Phillip Charles Carr ↔ Phillip Carr

Out-of-order name components
Diaz, Carlos Alfonzo ↔ Carlos Alfonzo Diaz

Initials
J. E. Smith ↔ James Earl Smith

Name split inconsistently across database fields
Rip · Van Winkle ↔ Rip Van · Winkle

Same name in multiple languages
Mao Zedong ↔ Мао Цзэдун ↔ 毛泽东 ↔ 毛澤東

Semantically similar names
PennyLuck Pharmaceuticals, Inc. ↔ PennyLuck Drugs, Co.

Semantically similar names across languages
San'in Telegraph and Telephone Corporation ↔ 山陰電信電話株式会社

Organizational aliases
Boston Brewing Company ↔ BeantownBeer

Learn more about why keyword search doesn't address the challenges of name matching.

Read **The Complete Guide to Name Matching**

How Match works

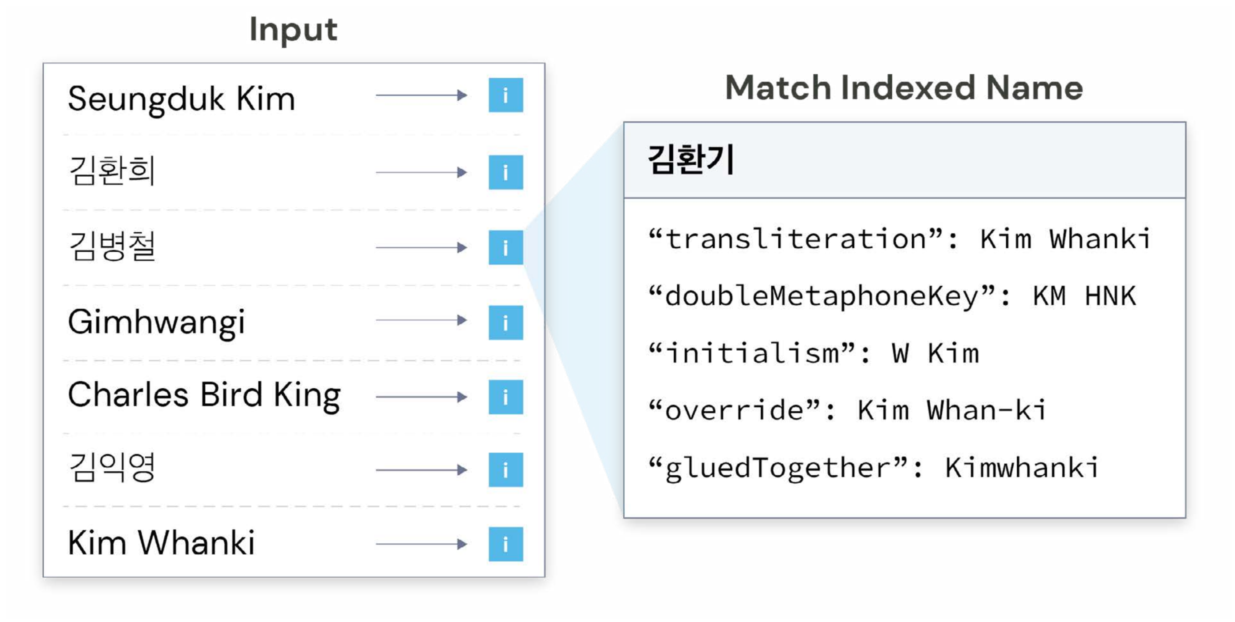
Every name matching operation begins with a database against which names will be checked. It can be as simple as a list with first and last names or have multiple fields containing other attributes. But it is structured and fielded.

Match combines multiple name matching methods into a multi-step process for the ideal balance of accuracy and speed.

- The indexing phase prepares the database for faster searches
- The querying phase compares an input (query name) to database entries in a first pass to identify potential match candidates
- The matching and scoring phase is a high-precision second pass that assigns an intuitive match score to reflect the confidence of the match. Based on the score, matches can be ranked from most to least likely.

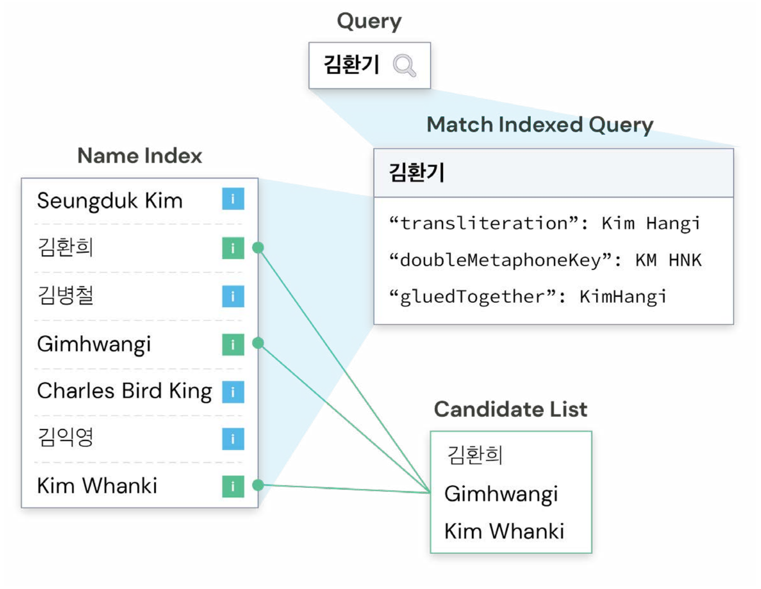
Indexing: Enriching the data

In the indexing phase, one or more databases of names are indexed so that each individual entry is enhanced with numerous keys. These keys are based on the many ways that names can vary, and they enrich the database(s) that will be searched. Match is essentially adding more information to each name in the database, such as how it appears in other languages, or what it looks like using initials, or how it may be pronounced. This operation expands the possible ways that an input (queried name) could match names in the database.

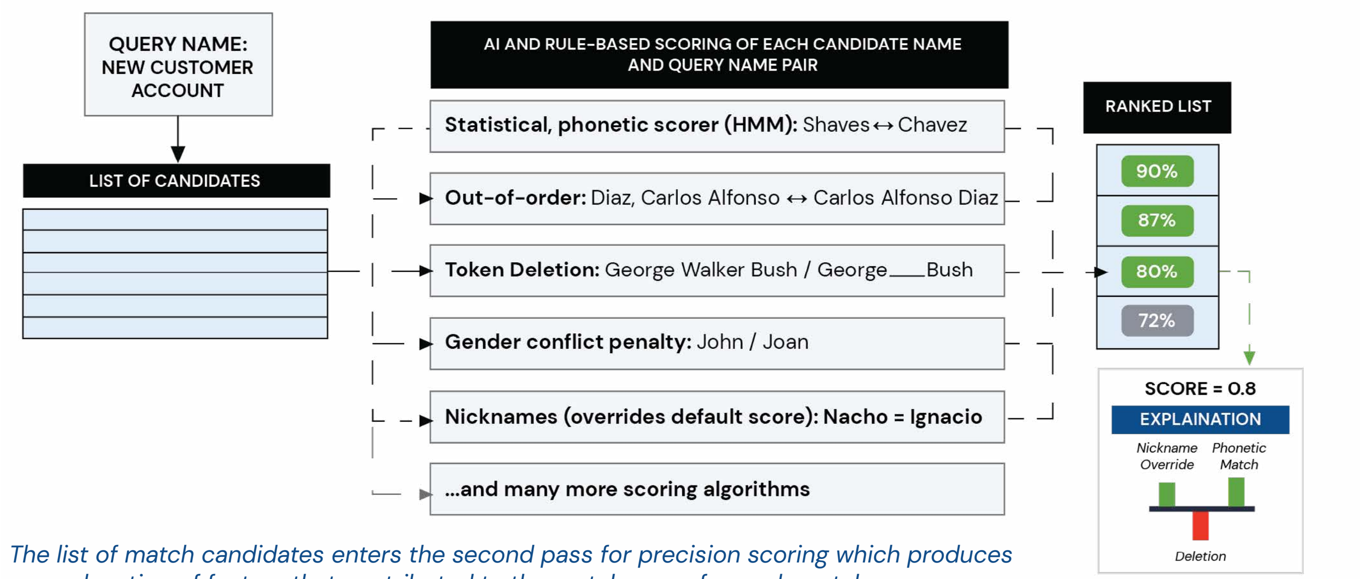


Querying: First pass to find match candidates

When a user inputs an *ad hoc* or batch request to match a name (or names), Match creates keys for each name in the query, similar to the indexing phase. This lightning fast first-pass is designed to be high recall (so it is unlikely to miss a potential match). The first pass retrieves match candidates by comparing keys in the query to keys in the database. Candidates are ranked from high to low match probability.
A user-chosen number of highest ranked candidates is sent to the second pass. Search recall increases if more candidates go to the second pass, but it also increases query execution time. This recall versus speed balance can be adjusted according to user preferences and risk profiles and set differently per user or even per query.



Matching: Second pass for precise match scoring

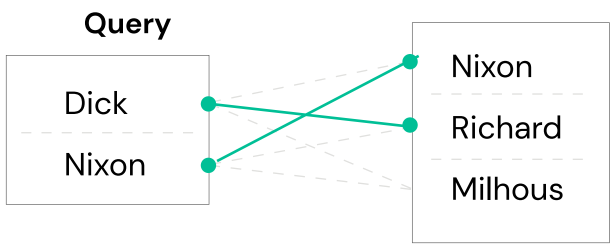

QUERY NAME: NEW CUSTOMER ACCOUNT
LIST OF CANDIDATES
AI AND RULE-BASED SCORING OF EACH CANDIDATE NAME AND QUERY NAME PAIR
Statistical, phonetic scorer (HMM): Shaves ↔ Chavez
Out-of-order: Diaz, Carlos Alfonso ↔ Carlos Alfonso Diaz
Token Deletion: George Walker Bush / George___Bush
Gender conflict penalty: John / Joan
Nicknames (overrides default score): Nacho = Ignacio
...and many more scoring algorithms
RANKED LIST
90%
87%
80%
72%
SCORE = 0.8
EXPLAINATION
Nickname Override
Phonetic Match
Deletion

The list of match candidates enters the second pass for precision scoring which produces an explanation of factors that contributed to the match score for each match.

The AI-based second pass is slower than the first pass but brings high precision matching (accuracy).

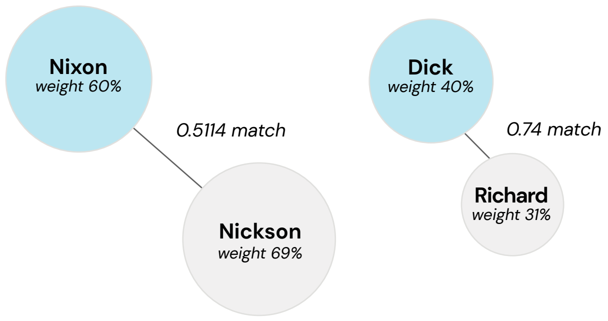
Token alignment

Match breaks down the candidate names into tokens (name components) and compares them against all the query's tokens to determine the best match.



Token scoring

Each token is weighted by its rarity in the given language, as well as by name and language-specific biases. The query's and candidate's tokens are compared and the ones that are most closely aligned are selected, no matter their order in the sequence.



Weighting rarity of name

The weighting model recognizes that certain names, like "John," are more common so that finding a match is not as significant as finding a match for a more unusual name, like "Dweezil."

Although the token pairs have the same match score, when considered in the context of the full name, the scores will be weighted differently when the rarity of the token changes. In the example below, Match applies a greater weight to the Dweezil match.

The match score for John Jones vs. John Johnson is 0.7902			
john	john	MATCH	1.0000
jones	johnson	HMM_MATCH	0.4396

The match score for Dweezil Jones vs. Dweezil Johnson is 0.9156			
dweezil	dweezil	MATCH	1.0000
'jones	johnson	HMM_MATCH	0.4396

The tokens at the ends of a name are considered slightly more significant than the tokens in the middle. So, a conflict in the final position will lower the score more than a conflict in the middle.

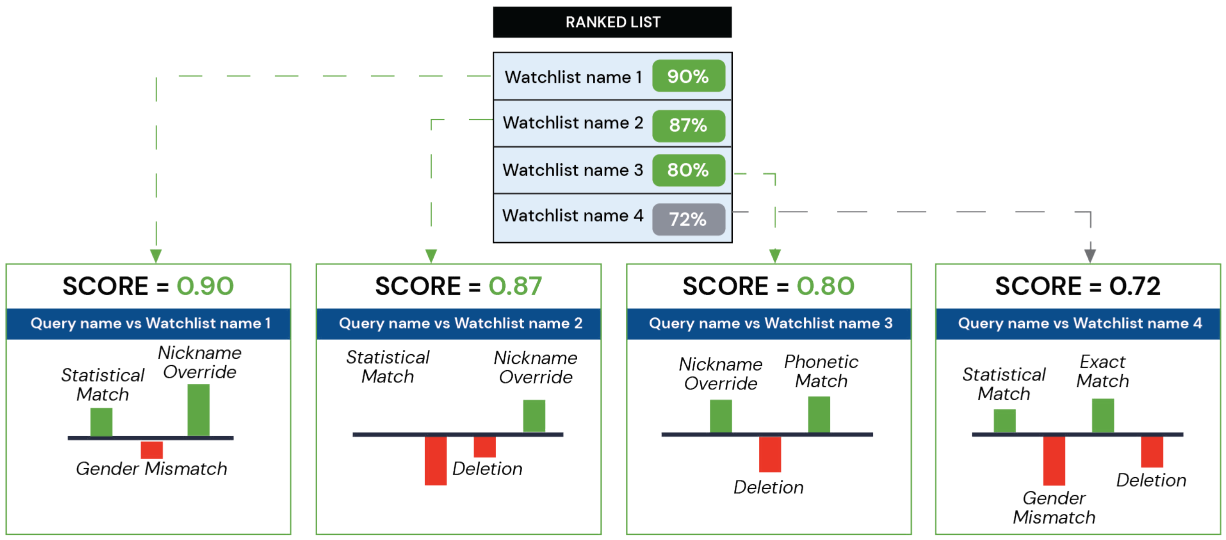
The match score for Richard William Smith vs. Richard Johnson Smith is 0.6155			
richard	richard	MATCH	1.0000
smith	smith	MATCH	1.0000
william	johnson	CONFLICT	0.1800

The match score for Richard Smith **William** vs. Richard Smith **Johnson** is 0.6110			
richard	richard	MATCH	1.0000
smith	smith	MATCH	1.0000
william	johnson	CONFLICT	0.1800

As each candidate/query pair is analyzed, AI and other algorithms adjust the match score for various types of matches. These algorithms have been carefully curated and calibrated by our engineers to optimize how each factor affects the final match score calculation. Configurations can also be set by the user to accommodate the characteristics of their data and desired goals. For example, a neural model that is especially accurate for matching names written in English and Japanese Katakana can be enabled when searching against a list of Japanese Katakana names.

Scoring

A final score is produced based on the weights assigned to various match phenomena as reflected in the system configuration. Scores are presented on a scale of 0 (no match) to 1 (exact match) and are also represented as intuitive percentages.



Algorithms addressing name variations (match phenomena) in the second pass

During the matching process, the algorithms in Match compare the input (query name) with the names in the database along multiple dimensions based on the ways names vary, also known as match phenomena. The following examples show how scores are calculated in the second pass, given different types of match phenomena.

Exact match

If two names are completely identical before normalization, they'll get a perfect score.

The match score for René Lindström vs. René Lindström is 1.0000 or 100%.			
rene	rene	MATCH	1.0000
lindstrom	lindstrom	MATCH	1.0000

Normalization

Names are normalized to a standard format, for example making all-capitalized tokens lowercase, removing diacritical marks, and disregarding "stop words" like Mr., Dr., or Esquire.

If the two names normalize to the same tokens, they'll get a near-perfect score.

The match score for René Lindström Jones vs. Rene LINDSTROM JONES is 0.9900			
rene	rene	MATCH*	1.0000
lindstrom	lindstrom	MATCH	1.0000
jones	jones	MATCH	1.0000

Statistically similar fuzzy matching and phonetics

Machine learning is used to recognize when two tokens are similar enough that they should match. One dimension is phonetic similarity. In the case of "Richard" and "Richerd," HMM_MATCH refers to the use of a statistical model that indicates "Richard" and "Richerd" are most likely the same.

The match score for Richard Smith vs. Richerd Smith is 0.9360			
richard	richerd	HMM_MATCH*	0.8017
smith	smith	MATCH	1.0000

Nicknames

Match recognizes common nicknames and cognates.

The match score for Richard Nixon versus Dick Nixon is 0.9516			
nixon	nixon	MATCH	1.0000
richard	dick	OVERRIDE	0.7400

Spaces and hyphens

Tokens that appear to have been glued together (are missing spaces) will be separated by inserting spaces between them and then matched as separate tokens. Hyphenated names are treated as if they were separated by a space.

The match score for MuhammadMulan Park versus Mulan Park is 0.8465				
mulan	mulan	MATCH	Original 1.0000	Adjusted 0.8667
park	park	MATCH	1.0000	
muhammad		DELETION	0.2690	

Similarly, adjacent tokens are also evaluated to see whether joining them together results in a compound token that might be a good match for any tokens in the other name.

The match score for Fred Will Sun vs. Fred Wilson is 0.8838			
fred	fred	MATCH	1.0000
will sun	wilson	HMM_MATCH	0.7090

Titles and honorifics

Normalization includes stop word elimination, such as titles and honorifics.

The match score for Mr. John Smith vs. John Smith is 0.9900			
john	john	MATCH	1.0000
smith	smith	MATCH	1.0000

Truncated name components

Tokens that have been slightly truncated are considered to match.

The match score for Morris Murgatroyd vs. Morris Murgatroy is 0.9522			
murgatroyd	murgatroy	TRUNCATED_EXACT_MATCH	0.8727
morris	morris	MATCH	1.0000

Gender

In this example, the fuzzy matcher finds 'joe' to be a better match for 'joan' than it does for 'john.' But the apparent gender of the two names does not match, so the final score is adjusted down.

The match score for Joe Smith vs. John Smith is 0.7258			
smith	smith	MATCH	1.0000
joe	john	HMM_MATCH	0.3675

The match score for Joe Smith vs. Joan Smith is 0.4188			
smith	smith	MATCH	1.0000
joe	joan	HMM_MATCH	0.4279

Missing name components

Sometimes, a piece of a name is missing, so a penalty is applied to the score.

The match score for Richard William Smith vs. Richard Smith is 0.8429			
richard	richard	MATCH	1.0000
smith	smith	MATCH	1.0000
william		DELETION	0.2690

Out-of-order name components

The order in which tokens appear is important, and there are several ways they can fall out of order.

Tokens that match, but that appear to be out-of-order, have their match scores adjusted to reflect that fact based on the severity of the mis-ordering.

The match score for George Herbert Walker Bush vs. George Walker Herbert Bush is 0.9523				
bush	bush	MATCH	1.0000	
herbert	herbert	MATCH	Original 1.0000	Adjusted 0.8000
walker	walker	MATCH	Original 1.0000	Adjusted 0.8000
george	george	MATCH	1.0000	

The match score for George Herbert Walker Bush vs. Walker George Bush Herbert is 0.8415				
bush	bush	MATCH	Original 1.0000	Adjusted 0.8000
herbert	herbert	MATCH	Original 1.0000	Adjusted 0.6000
walker	walker	MATCH	Original 1.0000	Adjusted 0.6000
george	george	MATCH	Original 1.0000	Adjusted 0.8000

A deletion is considered out of order if the context that it originally appeared in has also disappeared. In the first example below, 'herbert' has been deleted, but the tokens it appeared between are still adjacent. In the second example, that is no longer true.

The match score for George Herbert Walker Bush vs. George Walker Bush is 0.8726			
bush	bush	MATCH	1.0000
walker	walker	MATCH	1.0000
george	george	MATCH	1.0000
herbert		DELETION	0.2690

The match score for George Herbert Walker Bush vs. George Bush Walker is 0.8042			
1.2889 in	bush	MATCH	1.0000
walker	walker	MATCH	1.0000
george	george	MATCH	1.0000
herbert		OUT_OF_ORDER DELETION	0.2250

Initials

The use of initials can affect a match score.

The match score for Richard William Smith vs. Richard W. Smith is 0.9093			
richard	richard	MATCH	1.0000
smith	smith	MATCH	1.0000
william	w	INITIAL_MATCH	0.5420

Name split inconsistently across database fields

Consider a simple database consisting of three name fields: first, middle, last. What happens when an input name has those values distributed differently than expected across the database fields? When one token matches another token from the wrong field, a penalty adjustment is applied.

The match score for Richard William-Smith vs. Richard William Smith is 0.988				
richard	richard	MATCH	1.0000	
smith	smith	MATCH	1.0000	
william	william	MATCH	Original 1.0000	Adjusted 0.9200

Special algorithms for challenging name variations

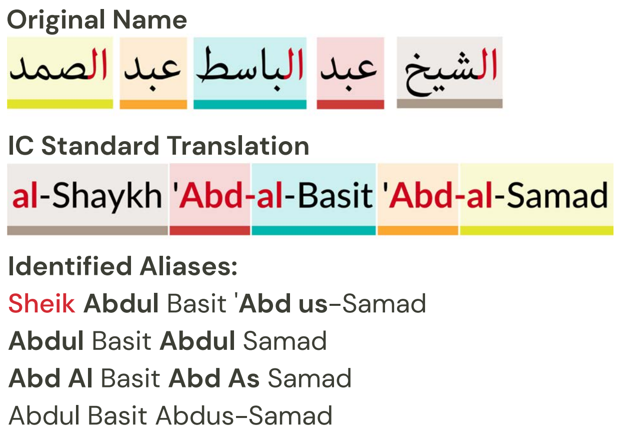
Language-specific algorithms

Users can always specify the language of origin of a name to increase the accuracy of Match. For example, although the name Norika Isoda is written in English characters, it is of Japanese origin. When not provided, Babel Street Match will guess the language of origin to enable it to apply language-specific algorithms to the matching. A few examples are described below.

Arabic transliteration spellings

In some languages, two different characters may be represented by one English character. Or, in an Arabic example, variations in transliterating a definite article changes how the name appears in English. The title "Sheikh" may sometimes be omitted.

- Sheikh Abdul Basit 'Abd us-Samad
- Abd Al Basit Abd As Samad
- Abdul Basit Abdul Samad
- Abdul Basit Abdus-Samad


Original Name
الشيخ عبد الباسط عبد الصمد
IC Standard Translation
al-Shaykh 'Abd-al-Basit 'Abd-al-Samad
Identified Aliases:
Sheik Abdul Basit 'Abd us-Samad
Abdul Basit Abdul Samad
Abd Al Basit Abd As Samad
Abdul Basit Abdus-Samad

Transliterated names from languages (such as Arabic) which have a richer set of sounds than English, will lose details in translation that hamper name matching efforts. Match has built-in knowledge of how Arabic sounds are expressed in English thus negating any loss of data in the transliterated name.

Matching names written in Chinese ideographs

Babel Street Match also implements the four-corner method (四角号码检字法 or 四角號碼檢字法) to improve the fuzzy name matching of names written in Chinese characters. This method represents the similarity of appearance between Chinese characters with a code for each corner of the character. Using these codes, Match adjusts the score for the comparison of two names when their four-corner method codes indicate that the characters do not resemble each other.

Furthermore, Match handles name matching between names in Chinese, Japanese, and Korean (which use Chinese ideographs in their names) as follows:

- Chinese simplified script ↔ Chinese traditional script
- Chinese simplified or traditional script ↔ Latin script (Romanization)
- Chinese simplified or traditional script ↔ Japanese Kanji
- Chinese simplified or traditional script ↔ Korean Hanja
- Korean Hanja ↔ Japanese Kanji
- Korean Hanja ↔ Latin script (Romanization)
- Japanese Kanji ↔ Latin script (Romanization)

Spanish-origin name matching

Match understands that names of Spanish origin frequently feature a two-part surname, such as "Gonzalez Lopez," where the father's surname (Gonzalez) is followed by the mother's surname (Lopez). Unlike names of English origin, where the "middle" name is frequently dropped, Spanish names are more likely to drop the maternal surname.

Cross-lingual name matching

While many names are written in Latin characters, many originate in other scripts. This can complicate matching names across languages.

Unlike systems that support only Latin characters, Babel Street Match processes names in native script for many languages including Chinese, Japanese, Korean, Arabic script languages, and Russian (language support details). In the process, it detects the language origin of the name (as it could be an English name written in Arabic script, for example). In those cases, Match also translates the name back to its language of origin using dictionary data, so that جورج بوش will be translated as "George Bush" instead of just transliterated to "Jurj Bush."

At indexing and for each query, each non-Latin name gets special treatment, including:

- If the name's language of origin is unknown, Match makes an educated guess.
- For names likely to be of native origin (e.g., a Japanese name written in Japanese Kanji), they are phonetically transliterated to generate a double metaphone key for each name. In the case of Japanese and Chinese where the script does not provide phonetic clues, dictionary data provides the "readings" of the characters.
- For names foreign to the script (for example, an English name written in Arabic script), Match translates the name using dictionary data.
- Where applicable, Match enriches the name with character normalization, such as adding vowels to an Arabic name.
- The product's algorithms are carefully curated to be invoked only for languages in which it has shown to help match accurately, as some algorithms may be counterproductive in other languages.

As with English names, matching is performed in two passes. The high recall first pass looks for matching phonetic keys, ranks them by confidence score and sends the top candidates to the second pass.

In the high precision second pass, the name in the original script is used in various matching techniques (depending on language) to ensure the most accurate results. At this stage, transliteration could result in loss of fidelity and interfere with finding the best matches. These techniques vary by language. For example, in one language, edit distance between characters is helpful, but not so in another language.

Long and multi-component names

Certainly not all names follow the three-field pattern of given name, middle name, surname. Many languages and cultures have different naming systems that include parents' names, places of origin, and other family names. These names may have six fields or more, along with multiple components that increase the number of potential variations and the complexity of matching names.

Arabic, Persian, Pashtun, and Urdu names typically have five to six fields.

Title	Given	Family	Grandfather's	Family
Al-Sheikh	Abdullah	Bin Hassan	Bin Mohammed	Al-Ashqar

Many traditional name matching systems try to anticipate these variations and create huge lists of name variants to match against. But with complex names, this is a losing proposition, and the systems are soon overwhelmed by the memory requirements and computing power needed to sort through all the permutations. For example, just one common six-component Arabic name generates nearly 3 million possible variants.

Babel Street approaches matching names through algorithms that compare name components, so matching a multi-component name like the one above does not take much more time than matching a two-component name. Most importantly, the solution can match a name despite never having seen it before, whereas systems that rely on pre-generated lists of name variants to match will be significantly less successful.

Semantic matching of company and organization names

Unlike personal names that mostly suffer from misspellings and typos, company names are difficult to match for other reasons. Company names are often composed of several common words, and synonyms of these words can be interchanged. Besides "Corp.," "Inc.," and "LLC," there are also synonyms like "Drug" vs. "Pharmaceutical." Additionally, when matching across languages, knowing what a word means is more important than how a word sounds. For example, **電信** in Japanese should be translated to "telegraph" and not "denshin."

Nippon Telegraph and Telephone Corporation vs. 日本電信電話株式会社			
Nippon (=Japan)	Telegraph	Telephone	Corporation
日本	電信	電話	株式会社

To accurately match names based on meaning, Match uses NLP techniques for word or text embeddings. This technique converts words into vectors (numbers) such that words with similar meanings are represented by vectors. The closer the vectors are in proximity, the closer the words are in meaning. A search for “PennyLuck Pharmaceuticals, Inc.” would likely find “PennyLuck Drugs, Co.” It also works to match the English translation of an organization’s name from another language.

Matching addresses and dates

Babel Street Match applies intelligent name matching algorithms to postal addresses. Match accepts fielded addresses or unstructured address strings, which it parses into fields. Depending on the type of field, the product applies the appropriate algorithm. For alphanumeric fields like postal code or street number, Match applies edit distance, which looks at character-level additions, substitutions, and deletions.

Specialized name matching algorithms compare text fields like “street name,” “house (aka, building name),” “city,” “province/state,” and “country.” Within each of the text fields, the solution matches with respect to:

Variation	Example
Phonetics and spelling differences	100 Montvale Ave vs. 100 Montvail Av
Missing address field components	100 Montvale Ave vs. 100 Montvale
Differences in upper and lowercase	100 Montvale Ave vs. 100 MONTVALE AVE
Reordered address components within a field	100 Montvale Ave. vs. 100 Avenue Montvale
Address field abbreviations	Montvale St. vs Montvale Street

When comparing two names, Match compares every field of one address against every field of the other address to look for the best match.

To further increase accuracy, Match groups related fields (such as “state, stateDistrict, island” or “city, cityDistrict, suburb”) so that if data in similar fields match, it can reduce the impact of misfielding during address parsing.

For example, if an address that includes the cityDistrict “Williamsburg” is parsed to assign “Williamsburg” to “city,” there will only be a small match penalty for the mismatched fields, because city and cityDistrict belong to the same address field group.

Field	Address 1	Address 2
House	Hawaii Paradise Apartments	
House Number	1351	1351
Road	Washington St.	Washington Street
Unit	312	312
cityDistrict	Brooklyn	
City	Williamsburg	Brooklyn
State	NY	New York

On the other hand, if the house field value “Hawaii Paradise Apartment” matches “Hawaii” in the state field of a different address, a large penalty will be assessed for these fields that don’t belong to the same address field group.

Match currently supports parsing of postal codes and abbreviations specific to the US, Canada, and UK locales and supports addresses written in Chinese.

Fuzzy date matching

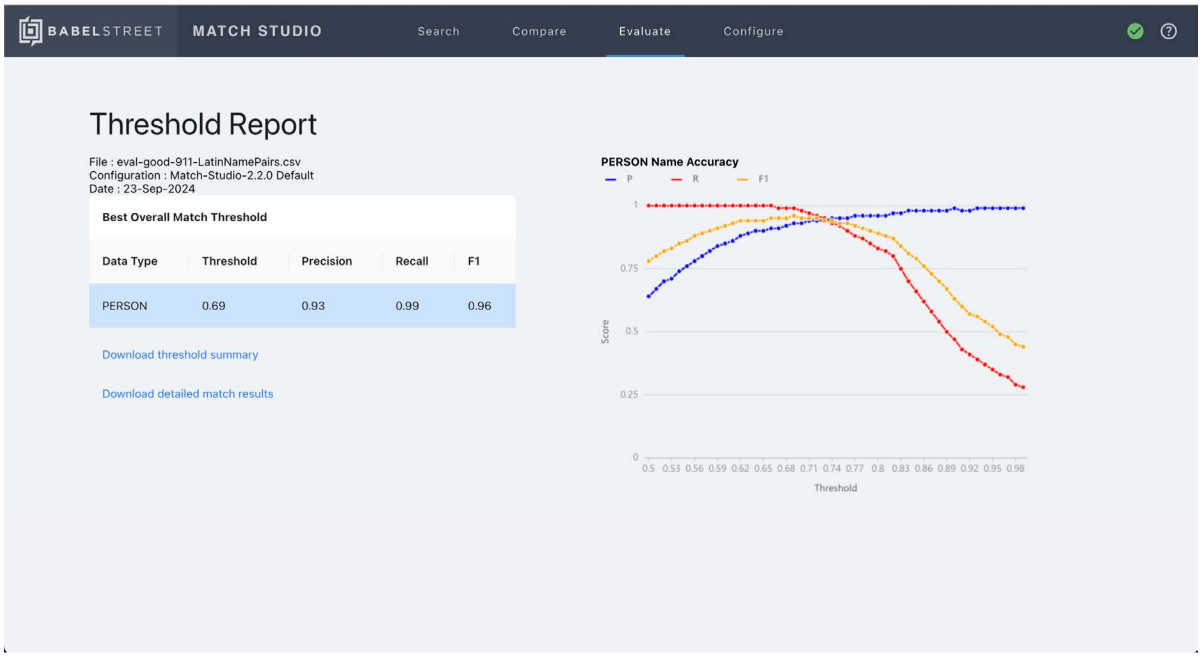
Fuzzy date matching by Babel Street Match complements address and name matching. It can compare partial dates and misordered date components (DDMMYY vs. MMDDYY) for the Gregorian calendar. In particular, the matching engine considers several aspects of dates:

- **Time** – The number of days between Date 1 and Date 2
- **Year** – The difference of the year fields of Date 1 and Date 2
- **Month** – The difference of the month fields of Date 1 and Date 2
- **Day** – The difference of the day fields of Date 1 and Date 2 (even if they are close in time, 1 and 30 are considered far apart)
- **String distance** – Date 1 and Date 2 to a standard format; then the string distance score is calculated based on the edit distance between the two strings.
- **Time proximity** – Based on a given interval of years, Match computes the chronological distance between dates in years to determine similarity.

Configurability and accuracy tuning

A key strength of Babel Street Match is its ability to be configured to match any customer's risk tolerance, adjust to new regulatory requirements, and tuned for greater accuracy on specific data. The carefully tested tuning parameters are what make every deployment of Match unique.

The user-friendly administrative interface, Match Studio shows non-technical users what factors went into each match score calculation. Further, in real time, Match Studio shows how changing different parameters (such as decreasing the penalty for a missing name component) affects match behavior. It also guides decision making on what the best "match threshold" (the match score above which a pair of names is considered "a match") is to meet business priorities by showing the threshold score at which precision and recall are balanced. A high threshold score reduces false positives, but increases the chance of false negatives.



Threshold Report

File : eval-good-911-LatinNamePairs.csv
Configuration : Match-Studio-2.2.0 Default
Date : 23-Sep-2024

Best Overall Match Threshold

Data Type	Threshold	Precision	Recall	F1
PERSON	0.69	0.93	0.99	0.96

Download threshold summary

Download detailed match results

While Match has nearly 100 configurable parameters, here are a few of the ones more commonly used for name matching.

- **Speed versus accuracy** — The Max Window Size parameter determines the percentage of match candidates that are sent to the second pass after being discovered in the first pass. The optimal size depends on the organization’s desired balance of accuracy/recall versus speed. Users can test the impact of smaller or larger windows sizes on the accuracy/recall of the matches returned. Smaller windows sizes mean faster execution. Larger window sizes mean more comprehensive coverage. Because of this parameter, every name queried is returned in the same amount of time, as controlled by max window size. In other words, speed is affected by the Max Window Size parameter, not the complexity of the names.
- **Reorder penalty** — This configuration penalizes the match score when name components appear out of order.
- **Deletion score** — A missing name component typically lowers the match score. This configuration allows a user to override the deletion penalty, enhancing the score even if a component is missing.
- **Boost weight at left/right end** — When users know that name components are in a particular order, for example “family name, given name, middle name,” a negative value will increase the weight of a name part at the left end of a name when contributing to the total match score. A positive value boosts the weight of the right end of the name. Identity fields can be similarly weighted to give more or less weight to fields such as name and date of birth.

Conclusion

Babel Street Match uses sophisticated AI and statistical models with carefully curated and tested algorithms to deliver accurate multilingual name matching of people, organizations, and locations. Match also intelligently fuzzy matches dates and addresses.

And while accuracy is important, so is processing speed in high-volume, high-stakes environments. Regardless of how common or complex a name is, or the number of name components, Match returns every query in a consistent amount of time using its two-pass approach.

A multitude of options lets users assign weights to name components and data fields according to the characteristics of the data and to fit a particular use case or risk profile. Match is easily deployed on-premises, in the cloud, or as a compatible integration with existing systems and search engines such as Elasticsearch and Apache Solr.

This makes Babel Street Match a top choice for identity resolution across government and commercial organizations.

Babel Street is the trusted technology partner for the world's most advanced identity intelligence and risk operations. The Babel Street Insights platform delivers advanced AI and data analytics solutions to close the Risk-Confidence Gap.

Babel Street provides unmatched, analysis-ready data regardless of language, proactive risk identification, 360-degree insights, high-speed automation, and seamless integration into existing systems. We empower government and commercial organizations to transform high-stakes identity and risk operations into a strategic advantage.

Learn more at **babelstreet.com.**

A-LIGN ISO 27001 CERTIFIED

© 2024 Babel Street. All Rights Reserved.
All names, companies, and incidents portrayed in this document are fictitious. No identification with actual persons (living or deceased), places, companies, and products are intended or should be inferred.

BABELSTREET