

Understanding Match Scoring in Babel Street Match

A look at how Babel Street Match scores name matches, why it does it that way, and how your organization can determine its optimal match threshold.

Introduction

Imagine that it is your responsibility to determine whether the name Jesús Alfonso López Díaz matches a name on a list.

- You need software to do this accurately because your list contains thousands of names.
- The software must do this flexibly because some of the names on the list are variations that an exact name search would miss completely.
- The software must do this quickly because you have hundreds more names to match after Jesús Alfonso López Díaz.

As you evaluate different software products, you see that they generate completely different results when they try to match Jesús Alfonso López Díaz against your list. One product returns “strong match,” but does not explain what that means. Another product yields a score of 13.103. Babel Street Match generates a score of 91%.

“Why the differences?” you ask. “How do these products score matches? What does each score mean? Which scoring method best fits my business and use case? Will each product find variations on the name? Can I specify the kinds of variations I want to include and exclude?”

We’ve written this paper to address those questions. It is designed to:

- Explain how Babel Street Match performs scoring
- Differentiate Match from scoring methods used in other products (in particular, Elasticsearch)
- Describe how to optimize match parameters and threshold in Match for more automated matching with higher confidence

You will gain an understanding of the steps Babel Street Match takes when scoring and the criteria it applies when searching for matches. You’ll also see how to use Match Studio to understand how Match calculates the score for names and learn how it can help optimize the match threshold and individual match parameters for your business objectives.

Approaches to Match Scoring — an Overview

Why score at all? Why not just determine “match/no match?”

The roots of scoring lie in how difficult it can be to match certain entities, such as people’s names, organization names, addresses, locations, and dates.

When you’re trying to determine whether the name Jesús Alfonso López Díaz occurs in your list, suppose your software returns “match,” with no other information. If your list is small and the risk of getting it wrong is low, you may be satisfied with the result. You may assume that Jesús Alfonso López Díaz matches the name on your list and act accordingly.

But if your list is very large and the risk of a mistake is high, you may ask yourself whether the list contains more than one Jesús Alfonso López Díaz. You may think about the possibility of other ways to spell the name. You’ll then want more insight into how your software has determined that there is a match.

Or, suppose your software returns “no match.” Again, with a large list and high stakes, you may wonder whether your list actually contains the right name, but spells it differently. Could a surname be missing? Could the name appear on the list, but with a nickname?

Entities, especially person names, are rarely an exact match

Thus, the binary approach of match/no match is not well suited to most high-velocity, high-risk use cases. In business situations, the decision process usually boils down to a few numbers:

- True positives (correct matches found)
- True negatives (correct non-matches found)
- False positives (non-matches labeled as matches) that you are willing to vet manually after the software has done its work
- False negatives (missed matches) that you may be willing to ignore, depending on your use case

In any business situation, name matching involves trade-offs among those numbers. The first two numbers indicate high quality in name matching software and the latter two indicate low quality.

Some organizations and tasks, such as border security, tolerate more false positives. They don't want to miss potential matches because of the high risk of admitting a dangerous person. They are willing to manually vet the search results from the software. For other tasks, such as Know-Your-Customer financial onboarding or transaction

linking, organizations prefer fewer false positives because the risk is lower and most mistakes can be rectified later. They are less willing to manually vet results and they want to see only likely matches.

Since decision-makers with different goals assess those trade-offs differently, name matching software that simply returns "match" or "no match" actually does a disservice. The real value of software lies in equipping the human decision-maker for the next step in the decision process. The software can do that by returning matches and indicating how confident it is in the strength of the match.

That's where scoring comes in. Since entities rarely match exactly, scoring gives an objective, consistent way to measure the likelihood, or confidence, that two entities are similar. Similarity scores help automate decision-making when the names aren't an exact match. In the example of Jesús Alfonso López Díaz, software could produce scoring results like those in Table 1, showing degraded variations with correspondingly lower confidence.

Score	Matching Name	Variation
100	Jesus Alfonso Lopez Diaz	Exact match
91	Jesus Alfonso Lobez Deaz	Misspelled family name
83	Jesus Alfonso Deaz	Mother's father's name removed
80	Jesus A. Deaz	Middle name replaced with an initial
77	Chuy A. Deaz	Given name replaced with a nickname
63	Deaz, Chuy A.	Reordered name components

Table 1: Sample scoring results ranked by descending likelihood of match. (Spellings are normalized to remove accent marks.)

Decision-makers and analysts might specify a threshold requiring a minimum match score of, say, 85 percent (to eliminate false positives), or 65 percent (to capture more false negatives). Then, armed with the scores and search results from the software, they could decide the next step in the decision process.

Which Methods are Used to Compute Match Scores?

Several approaches have evolved in using software to match names. In all cases, the goal is to quantify the difference, compute a score, and rank results, as in Table 1.

Edit distance

How many characters would you have to change to make the names match? In principle, fewer changes should indicate greater likelihood of a match.

Levenshtein distance¹ is one way to implement edit distance, which is a measure of similarity based on the number of one-character changes needed to make one string identical to another. The method is easy to implement and fast with short strings.

However, edit distance is better suited to comparing strings of text than entities. If you were querying for the name Jack, both Jac and Mack would yield a Levenshtein distance of 1². Human review, of course, would determine that, even misspelled, Jac is more likely a match than Mack.

The roots of scoring lie in how difficult it can be to match certain entities, such as people's names, organization names, addresses, locations, and dates.

How does Elasticsearch calculate a match score?

Elasticsearch is a search and analytics engine designed for a wide variety of data types, including structured and unstructured data. Because of its powerful capabilities in searching text and documents, it is sometimes applied to name matching.

Elasticsearch is built on Lucene, an open source project used by search engines such as Solr. Lucene uses term frequency-inverse document frequency (TF-IDF) and an algorithm called BM25 to estimate the relevance of documents to a query term, which it returns as a 32-bit digit. Elasticsearch implements fuzzy matching by taking the search results from Lucene, applying Levenshtein distance and phonetic analysis (see "Edit distance") and reordering the results by the number of edits required to match.

¹ https://en.wikipedia.org/wiki/Levenshtein_distance

² <https://planetcalc.com/1721/>

Rules-based: list method

How many different ways are there to spell and transliterate each component in the name? This method calls for making a list of all variations, then querying it against all possible combinations of the components. It's one way of generating results like those in Table 1.

But creating and maintaining a comprehensive list is resource-intensive. For example, the common Arabic name عبد الرشيد has thousands of variations, from abdal-rashid to 'abd-errchiyd. Searching the list is computationally intensive. The method does not handle names that are not on the list and the rules have to account for differences between languages and scripts.

Rules-based: common key

How similar do the names sound? Common key methods, like Soundex, reduce names to a key or code. The idea is to compare the common key of the query term against the common keys of names in the list.

Although the method is fast, it depends on how the name sounds in a reference language (usually English). Many sounds in other languages need to be approximated to the reference language or converted to the reference script, and that requires more matching rules to account for transliteration.

Statistical similarity

How statistically similar are the names? A statistical method uses a list of hundreds or thousands of matching name pairs to train an artificial intelligence (AI) model to recognize similarity among names. The resulting model calculates the probability that two names are a match and assigns a similarity score. It can directly match names written in different languages without first transliterating them to a reference script.

Of course, the greater the quantity of name pairs in the data set used for training, the higher the statistical accuracy of the resulting model. But that entails considerable manual effort before the software can do its work. Plus, a system using an AI model alone to search a large list for matches may be too slow for high-transaction environments.

Hybrid

No single method suffices for high-accuracy, high-velocity name matching. A successful software approach combines multiple methods.

Why is scoring in Babel Street Match different from Elasticsearch scoring?

While higher scores from Elasticsearch (Lucene) indicate better matches, the scores are not along a consistent continuum (such as 0 to 100). That makes it impossible to establish that scores above a given threshold, such as 80, are of a particular match confidence. In other words, it is difficult to guarantee that the likeliest matches will be at the top.

Even though the Elasticsearch approach of Lucene, BM25, and edit distance operate at the token level, they are better suited to computing token frequency. Babel Street Match is designed for aligning tokens. It is the difference between answering questions like, "How many times does Martin Luther King occur in these documents?" and "How closely does Martin Luther King match King, Martin Luther?" That is why high-recall, high-precision name matching requires the approach found in Match.

How Babel Street Match Performs Scoring: AI-powered, Hybrid Two-pass

Match fills in the gaps of traditional approaches.

It is:

- **AI-powered**, dynamically and simultaneously considering all the ways that names can vary, not just the ways found in the list
- **Hybrid**, applying the strength of one approach to overcome the weakness of another
- **Two-pass**, using the common key method to quickly eliminate obvious non-matches and statistical similarity to intelligently score each remaining pair of names.

Name match scoring by aligning tokens

As described above, the task of matching entities like people's names, organization names, and locations is different from that of matching text. So, instead of operating at the level of overall text similarity or phonetic components, Match operates at the level of the token, or portion of the entity.

In effect, Match is a token alignment engine, generating a match score that reflects how closely the tokens in the query entity — taken together — align with those on the list. Calculating the match score involves the following steps and algorithms:

1. Transform names into constituent tokens, then normalize and transliterate them. Use all similarity techniques to determine the best possible alignment.
2. Match and score every token in Name 1 against every token in Name 2, looking for the highest total score for each token pair. Each token aligns with some method — exact match, phonetic match, or distance match — and outputs a score.

3. Select the highest-scoring alignment of tokens.
4. Score deletions and conflicts because some tokens do not align at all (for example, a missing middle name). Lack of alignment reduces the strength of the match.
5. Calculate weighted score based on the weight of each token. The weights determine how important the token pair match is in calculating the final score.
6. Adjust final score. For example, apply a penalty if names do not appear to have the same gender.

The match scoring/token alignment process accounts for the many match phenomena that are unique to entities like proper nouns. It accounts for:

- The presence of stop words (Dr., Mrs., General)
- Nicknames and truncation (Vladimir ↔ Vlad)
- Languages and scripts (一郎 ↔ Ichiro)
- Acronyms and initials/initialisms (ABC ↔ Australian Broadcasting Corporation)
- Gender mismatch (John Smith is a closer match to Joe Smith than Joan Smith)

The process yields the kind of results shown in Table 1.

The real value of name matching software lies in equipping the human decision-maker for the next step in the decision process.

Using Match Studio to Analyze Match Scoring

Match Studio is an interactive tool for exploring name matching in Babel Street Match. The Compare tool displays the details of a pairwise match, including the algorithms Match uses to calculate the match scores.

Let's start by comparing the person name Jesús Alfonso López Díaz with itself:



Home > Compare

Compare

Person Name Organization Name Location Name Dates Addresses

Left Name

Full Name

Spanish Jesús Alfonso López Díaz

Right Name

Full Name

Spanish Jesús Alfonso López Díaz

+ Show Configurations: Custom

Clear Compare

As expected, Match computes a 100% match; in other words, 100% confidence that the two names refer to the same person:

Match Score	Input Summary	Left Name	Right Name
	Data	Jesús Alfonso López Díaz	Jesús Alfonso López Díaz
	Normalized Data	jesus alfonso lopez diaz	jesus alfonso lopez diaz
	Latin Data	jesus alfonso lopez diaz	jesus alfonso lopez diaz
	Script	Latin	Latin
	Language of Use	Spanish	Spanish
	Language of Origin	Spanish	Spanish
	Entity Type	Person	Person

	jesus	alfonso	lopez	diaz	Matchtype	Raw Score	Context Score
jesus					MATCH	1.000	1.000
alfonso					MATCH	1.000	1.000
lopez					MATCH	1.000	1.000
diaz					MATCH	1.000	1.000

The match score computation presented by the Compare tool shows how Match has transformed the name into tokens. In this case, each token corresponds to one of the four parts of the person's name. It also shows normalized spellings, without accent marks, punctuation or capital letters. The table shows the statistical weight given to each token (as a percentage) and the match reason for the token pair (here, exact

match). It also shows the match score for each of the four token pairs: in this case, 1, meaning 100% confidence. Beneath the table is an optional explanation of how Match calculated the match score.

Continuing with the name variations in Table 1, a comparison with Jesus Alfonso Lobez Deaz yields this result:

Match Score

92.2%

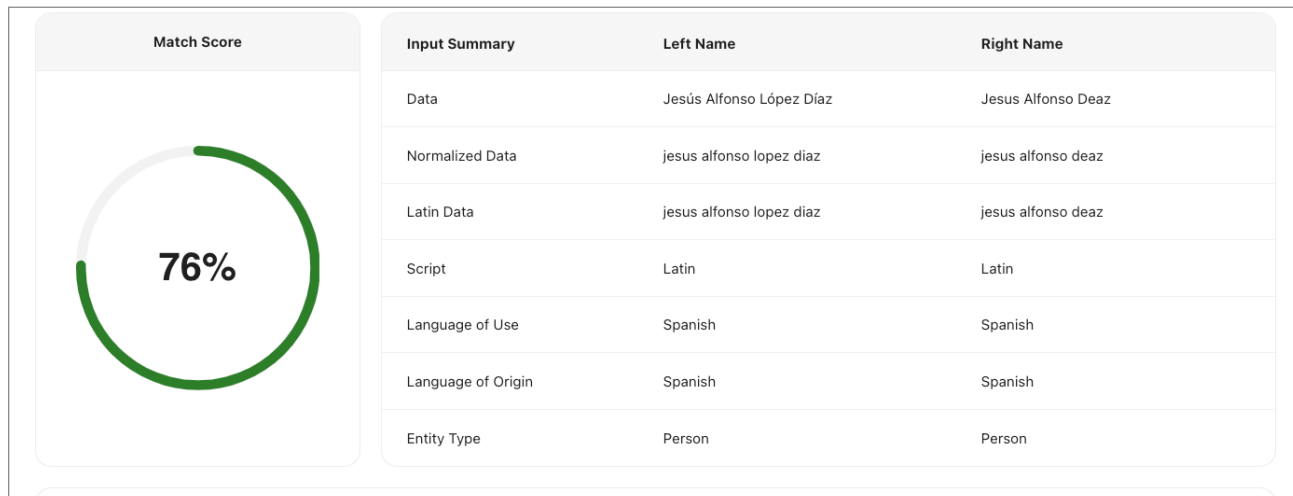
Input Summary	Left Name	Right Name
Data	Jesús Alfonso López Díaz	Jesus Alfonso Lobez Deaz
Normalized Data	jesus alfonso lopez diaz	jesus alfonso lobez deaz
Latin Data	jesus alfonso lopez diaz	jesus alfonso lobez deaz
Script	Latin	Latin
Language of Use	Spanish	Spanish
Language of Origin	Spanish	Spanish
Entity Type	Person	Person

	jesus	alfonso	lobez	deaz	Matchtype	Raw Score	Context Score
jesus					MATCH	1.000	1.000
alfonso					MATCH	1.000	1.000
lopez					HMM_MATCH	0.803	0.803
diaz					HMM_MATCH	0.673	0.673

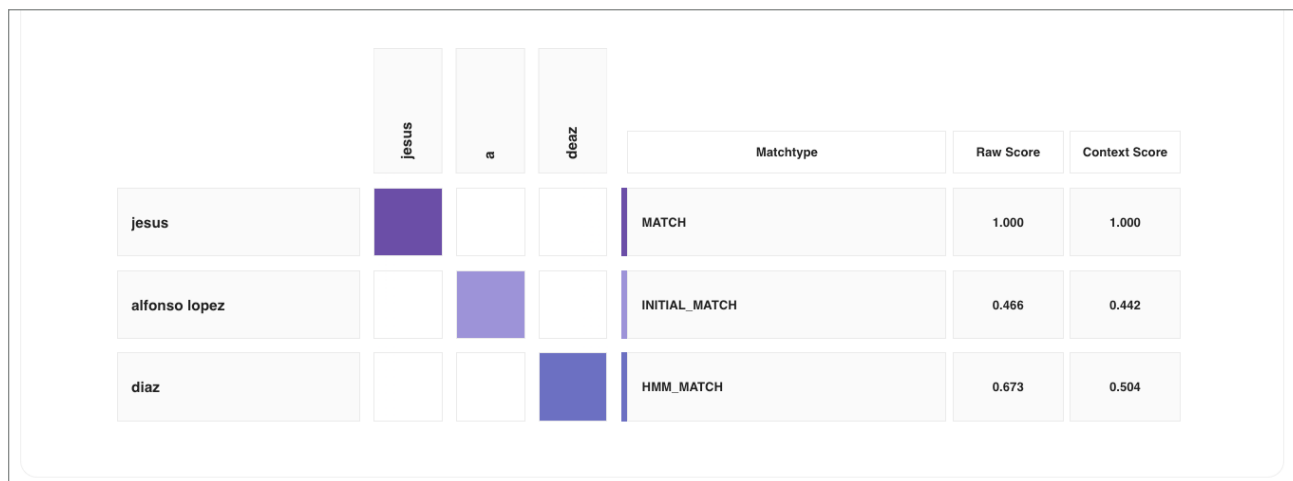
Because Lobez ↔ Lopez and Deaz ↔ Diaz are not exact matches, the Compare tool shows that Match evaluated other ways to compare them. Match selected a fuzzy-match statistical model, according to which the tokens were scored as .803 and .673, respectively. Unusual tokens receive more weight than common names because it is more significant when they match.

The final score of .922 reflects 92% confidence that the names refer to the same person.

Comparing with Jesus Alfonso Deaz shows the effect of a missing surname (token without a corresponding match), pushing match confidence down to 76%:



Next, the Compare tool shows how Match compares Jesus A. Deaz by aligning and weighting the initial "A." for Alfonso–Lopez:

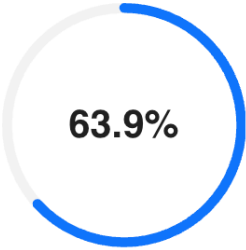


In a comparison with Chuy A. Deaz, Rosette uses an override file and returns .758 or 75.8% confidence that Chuy is a nickname for Jesus:

	chuy	a	deaz		Matchtype	Raw Score	Context Score
jesus					OVERRIDE	0.740	0.740
alfonso					INITIAL_MATCH	0.542	0.493
lopez					DELETION	0.269	0.269
diaz					HMM_MATCH	0.673	0.673

Deaz, Chuy A. is an example of reordering name components:

Match Score



63.9%

Input Summary	Left Name	Right Name
Data	Jesús Alfonso López Díaz	Deaz, Chuy A.
Normalized Data	jesus alfonso lopez diaz	deaz, chuy a.
Latin Data	jesus alfonso lopez diaz	deaz, chuy a.
Script	Latin	Latin
Language of Use	Spanish	Spanish
Language of Origin	Spanish	English
Entity Type	Person	Person

Match can correctly align Jesus ↔ Chuy, “A” ↔ Alfonso Lopez and Diaz ↔ Deaz, but the confidence on each token pair is low, affecting the overall match score.

Taking advantage of scores generated by Match

Babel Street Match produces an overall score between 0 and 1 to reflect the system's confidence that two names match. Scoring names along the range from 0 (low confidence) to 1 (high confidence) has several advantages:

- **You can easily understand the score and convey it.** The preceding series of examples shows pairwise matching between Jesús Alfonso López Díaz and variations — humans can understand each score as an indicator of probable match. In a scenario of index matching, humans can infer from the score that the closest match found in the list refers to the same person.
- **You can integrate Match without having to replace your existing search engine.** The token alignment and match scoring in Match complement the functions of search engines with high-speed, scalable, cross-language, and cross-script name searches.
- **You can automate a greater share of the work of matching entities** (person names, organization names, locations, dates, and addresses). That pays off in tasks such as processing customer applications, screening names, and researching news.


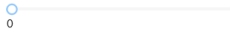
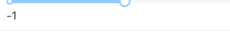



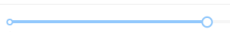


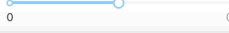

Because scores go beyond the match/no match dichotomy to indicate confidence, you have the flexibility to determine the optimal match threshold for your specific use case, as described below.

Using Match Studio to Experiment with Parameter Settings and Match Threshold

One way to experiment with the calculation of match scores in Match Studio is through the Show Configurations pane in the Compare tool. Show Configurations makes it possible to fine-tune individual match parameters and gauge the effects on match score computation.

Match scoring depends on more than 100 parameters, of which Compare exposes approximately 20 (depending on the language pair). That is how Match Studio implements the transparency and adjustable scoring that are important differentiators between Match and other name matching products.

Of course, the default settings used in actual Match deployments are based on decades of research and tens of millions of successfully matched names in hundreds of environments. Still, Match Studio makes it possible to experiment with those parameters and see how they can increase or decrease the match score between two names.

Parameter	Set Value	Modified
adjustOneSidedDeletionScores	 0.5 1.5	1.131
boostWeightATBothEnds	 0 5	0.044
boostWeightATRightEnd	 -1 1	0.000
conflictScore	 0 0.5	0.180
deletionScore	 0.001 0.5	0.269
finalBias	 0.1 10	2.400
initialsScore	 0 1	0.542
joinedTokenInitialsPenalty	 0 1	0.859
joinedTokenPenalty	 0 1	0.950
outOfOrderDeletionScore	 0.000001 0.5	0.225
reorderPenalty	 0 0.5	0.237

To follow the earlier example of Jesús Alfonso López Díaz, suppose that you wanted to see only high-confidence matches. You might specify a match threshold of 80% to reduce the number of false positives you had to review manually. In that case, Jesus A. Deaz would not appear as a match because Jesús Alfonso López Díaz ↔ Jesus A. Deaz yields a score of only 73.8%.

Conversely, suppose that you tolerated false positives and wanted to reduce the number of false negatives — potential matches you didn't want to miss. To see Jesus A. Deaz and similar names, you could increase the Initials Score in Show Configurations from .542 to .90. That would raise the overall match score for Jesús Alfonso López Díaz ↔ Jesus A. Deaz to 80.8%. Therefore, Jesus A. Deaz would appear as a match because the new score exceeds your match threshold.

Because Match Studio generates a score rather than match/no match, you have the flexibility to determine the correct match threshold for your organization and then automate matching based on that. The Evaluate tool in Match Studio is designed for computing the match accuracy and corresponding match threshold of a data set you provide. With the right threshold, Match can perform AI-powered, hybrid, two-pass match scoring that is ideal for your business environment.

Step 1: Set priorities and objectives for your business

Determining the right threshold is more than a function of mathematics. It also involves the priorities you set for your use case. The main factors in evaluating the quality of name matching include:

- **Precision** — The number of correct items out of the total number of items found. Likelier matches mean fewer time-wasting false positives. This is well-suited to use cases like patient record searches or bank compliance screenings.

- **Recall** — The number of correct items found out of all possible correct items. Less-likely matches mean casting a wider net. This is well-suited to high-stakes situations, such as border security and no-fly lists.
- **F-score or F1** — A measure that attempts to balance precision and recall. Often called the accuracy of the system.
- **Speed** — The importance of performance in match scoring. In environments like transportation security, the speed of processing is also a priority.

The goal — for both the organization and Babel Street Match — is to strike a balance among precision, recall, and speed to achieve optimal accuracy for your use case.

In most organizations, it's important to establish whether precision or recall is more important before evaluating a data set and determining the ideal match threshold.

With the right threshold, Match can perform AI-powered, hybrid, two-pass match scoring that is ideal for your business environment

Step 2: Collect and prepare the data set

Prepare these text files:

- **“Gold standard” list** — This list includes approved, annotated name pairs. The file should contain both positive (you consider the names a match) and negative (you do not consider the names a match) name pairs. Your choice of positive and negative name pairs should be consistent with the priorities and objectives you set in step 1. The text file

represents your “gold” evaluation data set because it indicates desirable match results for your organization and use case.

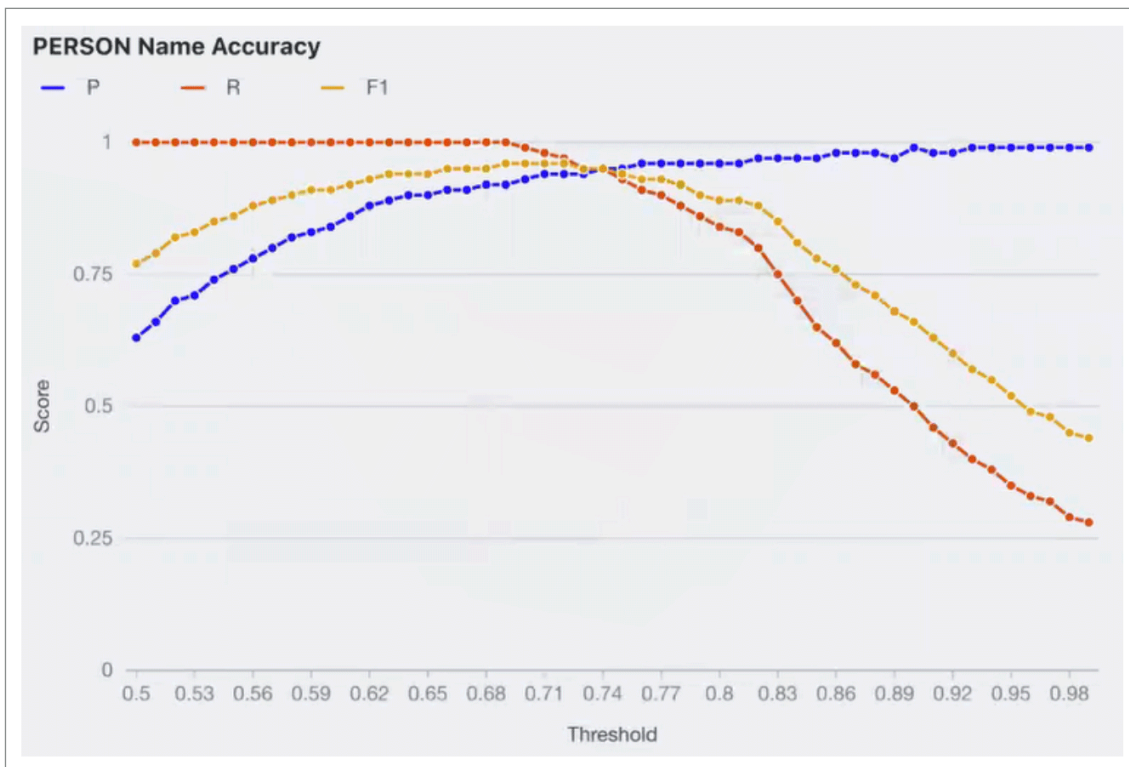
- **Match configuration** — This contains the configuration parameters for your test. Match Studio ships with a default configuration.

Follow the formatting and header guidelines shown in the Evaluate tool.

Step 3: Upload the files

Upload the files to the Evaluate tool. As shown in the example below, the tool plots precision, recall,

and F-score in a threshold report based on your data set:



It also computes the best overall match threshold — the one resulting in the best score of all those evaluated:

Best Overall Match Threshold				
Data Type	Threshold	Precision	Recall	F1
PERSON	0.71	0.94	0.98	0.96

Based on the gold data set used in this example, Match recommends a match threshold of .71 for matching similar person names in the future. From the Search menu in Match Studio, you can modify your search configuration so that results with scores greater than or equal to .71 are considered matches. You can also use the Evaluate tool to experiment with different gold data sets and search configurations.

Conclusion

There are several approaches to matching entities. As described in this paper, entities like person names and organization names vary in countless, unexpected ways. They have characteristics that defy the techniques used for search or document comparison. The essential problem is matching all desirable, potential candidates for a given use case and computing a score that explains the similarity meaningfully.

Babel Street Match embodies three important tenets of identity matching:

- Instead of operating at the level of overall text similarity or phonetic components, Match operates at the level of the token, or portion of the entity.
- It fills in the gaps of traditional approaches with an AI-powered, hybrid, two-pass method of matching.
- Unlike products that return match/no match, Match generates transparent, explainable scores that can be compared across multiple queries and around which business logic can be configured.

Of greatest importance, the Babel Street Match approach offers unique value by factoring in the trade-off between precision and recall and by allowing you to tailor individual parameters and match thresholds to use cases. That flexibility enables you to implement name matching technology ideally suited to the business risk you face.

Because scores go beyond the match/no match dichotomy, you have the flexibility to determine the optimal match threshold for your specific use case.

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.