

Feature Highlight: Fuzzy Matching

Stefanie Histed

When my parents decided to ignore common spelling conventions and go with “Stefanie”, they didn’t know they were setting me up for a lifetime of annoyance and inconvenience. As a child, gift shop key chains and stickers were not made for me. As an adult, all types of important documents have had my name spelled incorrectly—everything from bank forms to government paperwork. However, the most disappointing example was receiving my high school diploma with my name spelled with a "ph" instead of an "f"- *Stephanie*. If your name is spelled wrong on your diploma, did you even graduate?

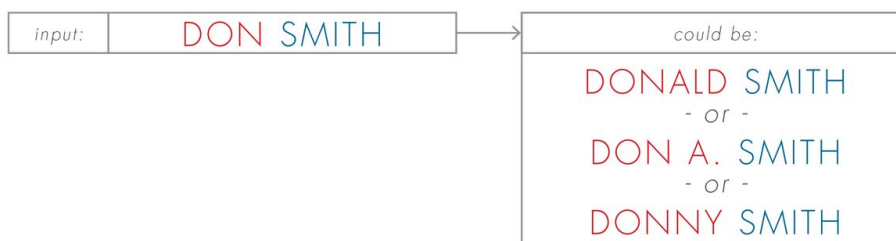
If I could get back all the time spent on the phone having the “well, did you spell my name with an F? I’m definitely in your system” conversation, I’d probably have enough time to have hobbies. If only all systems would utilize the modern magic of Fuzzy Matching, this dream could be achieved.

Fuzzy Matching is a way to process word-matching user or system queries to find a match in a database. When there is no exact match (eg “Kate”), fuzzy matching finds likely matches (e.g “Katie”).

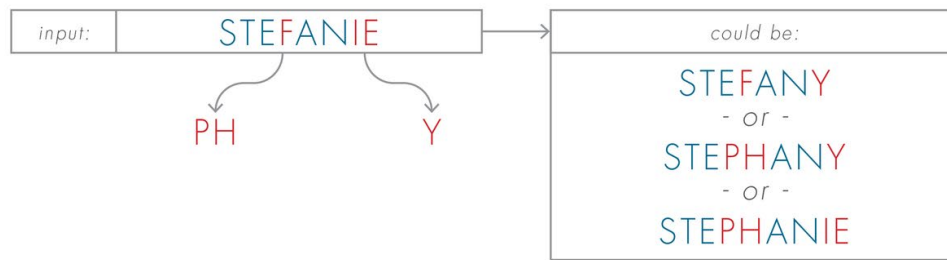
eLect VRS uses Elasticsearch, a distributed full-text search engine, to provide the Fuzzy Matching capabilities to the system. On top of the functional benefits, *eLect VRS*’s query times for voter records are 10x faster compared with legacy systems (5-10 seconds down to a ½ second in our tests).

Here are three types of Fuzzy matching used in modern VRS:

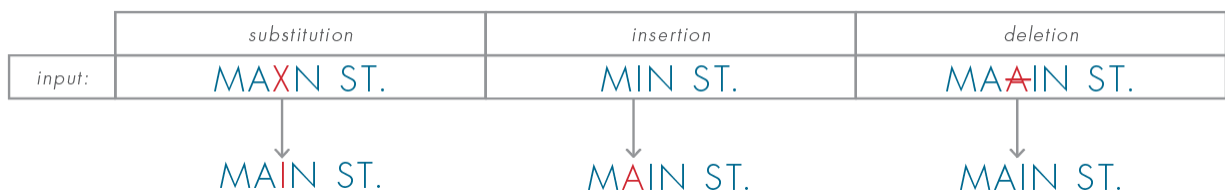
- *Name Matching* — Name matching tables can be set up to find similar names, including nicknames or abbreviations. With this type of matching, you’ll have a better chance of finding an original registration rather than entering a person under two different names.



- *Soundex* — An algorithm for indexing words by phonetic pronunciation. Helpful for a name like mine—if you type “Stephanie”, it knows that the “ph” could also be an “f”, and would offer you my correctly spelled voter record.



- *Levenshtein Distance* — This type of fuzzy matching is perfect for catching mistakes by fast (or bad) typers—by setting the “edit distance” of how many letters off something can be, you can find matches even with typos.



If these descriptions of Fuzzy Matching are leaving your brain a bit...fuzzy, let’s talk instead about why it’s important. Here are a few benefits the application brings to a voter registration system:

- *Database Interoperability* — Matching works not just in manual entry scenarios, but also while loading data from external databases—making sure any data you import is matched and vetted against information already in your system.
- *OCR/ICR Improvement* — When petitions (or other documents) are scanned, the results also run through matching programs, helping to find matches despite the quality of handwriting, squished letters, or differences in the nickname they wrote vs their name in the VRS.
- *Accuracy* — Catching misspellings and typos decreases the likelihood of creating duplicate invalid records.
- *Reduces Human Error* — Errors in manually transcribing data from hard-to-read handwritten registrations, address changes, or petition signatures are noticed quickly and easily.
- *Efficiency* — Fast search engine saves processing time, leaving administrators more time for other tasks.
- *Bypasses Future Problems* — Avoiding the entry of duplicate voter records in the first place not only keeps data rolls clean, but saves administrators from having to research and reconcile potential duplicates later. Avoiding misspellings saves voters from problems when checking in to polling places on voting day

As more databases turn to fuzzy matching for efficient processing speed and accuracy of data, and I spend less time on the phone with customer support, I think I’ll use my new free time to start a side-business...designing vanity keychains for all the Stefanies, Ericks, and Jennyfers of the world.