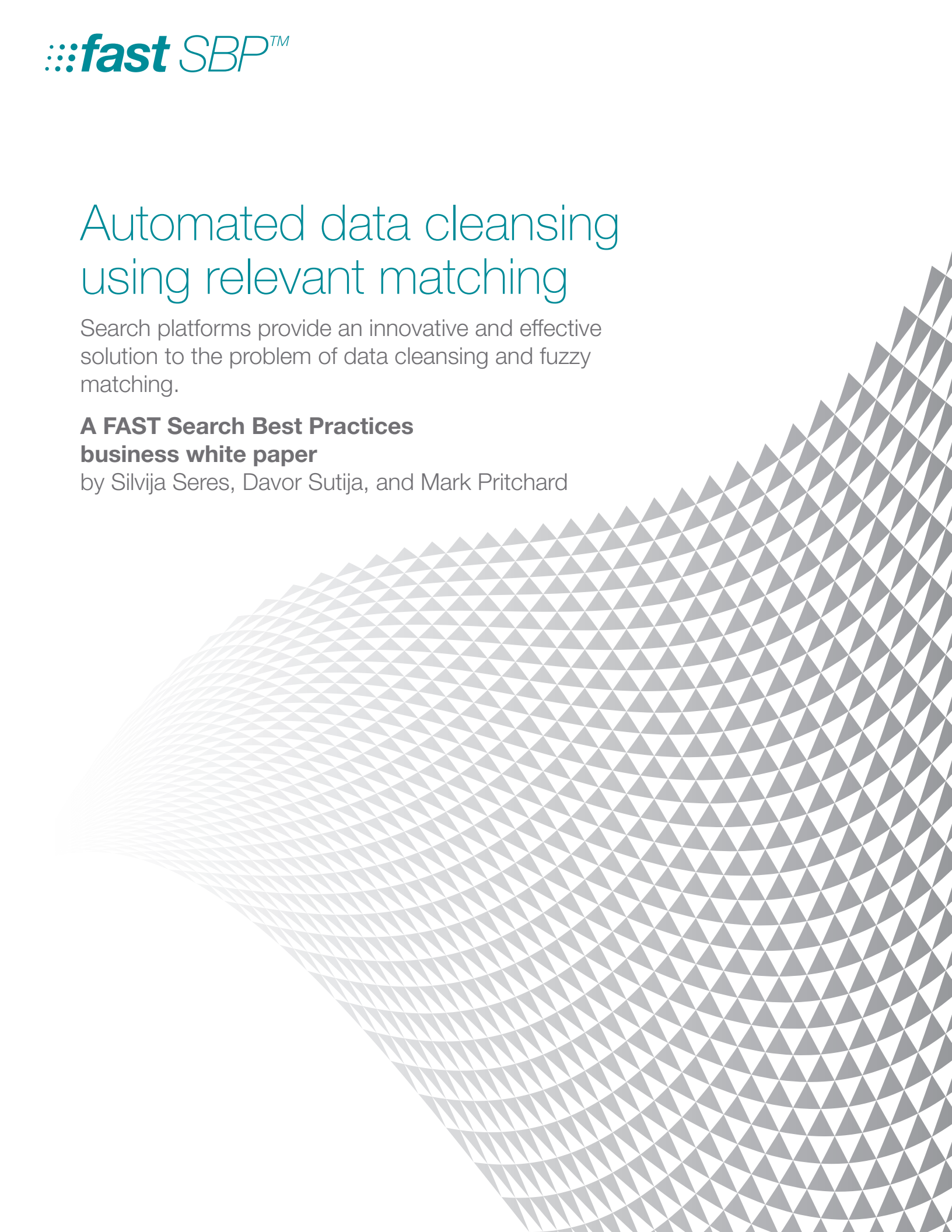
fast SBP™

Automated data cleansing using relevant matching

Search platforms provide an innovative and effective solution to the problem of data cleansing and fuzzy matching.

A FAST Search Best Practices business white paper

by Silvija Seres, Davor Sutija, and Mark Pritchard



Introduction

Integrating and merging data across multiple silos is a costly but necessary business for most large organizations. There are many reasons: re-organizations, mergers, compliancy requirements, re-use of information, cutting operating costs, or simply a need for a richer picture of customers or situations. As systems grow and their intended use changes over time, there is a regular need to join partially overlapping data sources, such as multiple customer registers without a common key.

Such unified picture of data across a number of silos requires significant data normalization with cleansing and matching capabilities. No matter how disciplined and organized the content providers are, data in different silos often evolves in different directions. Over time, the business-critical data becomes increasingly ambiguous: sources from different parts of the organization vary in formats, models, spelling, or local conventions. Data supplied by customers is even less clean, with broad variances caused by lack of knowledge about the accepted formats, expected keys or simply proper spelling. A traditional manual "normalization" effort takes a long time, has high associated costs, and is itself ridden with the same imperfection problems it is trying to solve. The alternative is to turn to modern search engines for rescue, with approximate matching capabilities and weighted multi-field comparison.

FAST has multiple live implementations in this area. Examples include several major telcos with phone-book databases that need cleansing of 'yellow pages' and 'white pages', and multinational banks with databases that need cleansing of customer information after mergers, or black-listing and cleansing of undesirable or dangerous customers. In this whitepaper we will explore a case with one of the major Telcos, and their use of data cleansing.

Search-based data cleansing is used together with 'fuzzy' or imprecise matching logic, for cases where there is no exact match for a given search. This automatic approach removes the need for costly, slow and often incorrect manual corrections that are used in traditional solutions. The ambiguities that are automatically resolved occur in two separate problem domains:

Parsing of imprecise input – data to be inserted in the system, that is either incorrect and should be deleted, or is misspelt and should be corrected

Matching of ambiguous results – consistent results when faced with choices among alternative answers

We will review each of these areas in some detail later. Together, the solutions to these two problems enable rich reuse of imperfect information – a pragmatic, practical and cost-effective alternative to the utopian vision of perfect data, enforced through perfect discipline and precision among all data producers, data classifiers, and data integrators. The search-enabled solution works, and provides rich and correct information in a world where such imperfect and overlapping data needs to put in context, improved by technology, and revealed in all its strategic power.

Case Study Overview

One of world's largest telcos has turned to enterprise search for improvement of the ambiguous data in their database of customer addresses.

This company had two main business reasons for this data improvement exercise: one was to improve the "Front of House" services, in their customer facing shops and call centres, and the other was to improve "Back of House" services, in their internal operations such as billing, customer analysis, and dispatching of service personnel.

The implementation was planned in several phases. In the first phase, this telco wanted to cut costs in a couple of concrete areas:

- Improved dispatch precision for services, such as installers and equipment send-outs;
- Cheaper postal services for mass mailing such as bills, product brochures or other marketing, through less returned mail, and cheaper bulk mailing agreements with guaranteed addresses correctness (and a premium charge for wrong addresses)

In the front of house, their problems were introduced primarily by the interactive nature of telco customer services, where data feeds and answers are expected in real time. A significant portion of data entry and querying is expected to be entered by customers. This often leads to misspelled data and vague queries. The quality problems

are often intensified by poor mobile lines that affect correct phonetic understanding of names and addresses, and by the fact that a significant proportion of customers and call centre employees have English as a second language. Usual mistakes in this area include: wrong synonyms (e.g. homestead instead of station), alternative spellings (e.g. Jeffrey instead of Geoffrey) and wrong use of bi-grams (e.g. Westlake instead of West Lake).

In the back of house, the problems were mainly in the area of matching conflicting addresses. For example, a given street may change its name as it continues through different parts of a city, e.g. through different localities. Some customers may have provided a wrong street name but the correct locality (often the case with residential customers, who may be used to outdated street naming conventions) or they may give the true street with the wrong locality (a common case with business customers wishing to appear to be located in the neighbouring, more upmarket, suburb). In these cases, capabilities such as calculation of geographic distance from addresses become crucial precision factors.

On the technical side, the address database contains approximately 14 million hierarchically structured addresses stored in an IBM DB2 database on a Z/OS mainframe. Query rate is approximately 15 queries per second. The database is well structured, with fields for "main address", "sub address", "property" and so on, but this internal structure allows for the creation of ambiguous address terms, through multiple possible representations for the same address. For example, the address "Caltex House, 170 Anne St" can be represented in two ways, as a subaddress "Caltex", with type 'HSE' or 'house' at '170 Anne St', or as a property "Caltex House", at 170 Anne St.

At a high-level summary, this telco's data-quality challenges can be grouped in three categories:

Address correctness issues – incorrect information where names have been misspelled or shortened, and missing information, such as "170 Anne St Brisbane" (no state or postcode), and database-specific ambiguities as shown above

Address parsing ambiguities – for example, in Five Level House, 170 Anne St, Melbourne 3000, is the word 'level' specifying a level (e.g. Level 5), or the name of a property (five level house), or other cases where lacking information leads to multiple possible interpretations

Lack of separating commas to delineate depth – the database requires this to be specifically (and correctly) split out into appropriate levels before it can be matched (e.g. Suite 501 L 5 shell house 170 north tc Adelaide)

The new search based solution allowed a correct parsing of duplicate addresses, cleansing of incorrect data, and automatic steps for evaluating ambiguous queries.

Traditional Approach to Data Cleansing

This telco has previously used an ETL-based approach to clean their data. The features of the underlying database were used directly for e.g. substring search in fields, creating new fields containing phonetic encodings and breaking the query down to the smallest details to allow exact matches on this highly structured data.

This ETL-based solution had several shortcomings.

One problem was a poor user experience and ultimately poor data quality because user needed to be trained on the different types of address so they could enter them correctly. For example, they needed to differentiate between physical addresses, parcels, postal addresses, lots etc: "14A Brogil Road North Warrandyte" is a physical address, while "45FG Backwater Close Nagambie" is a postal address. This is not obvious, and the user often doesn't know how to choose (or even what the choices are), and has to try the different options.

Another problem was that implementation of advanced free text functionality is not a core competency of database products, as databases tend to have a focus on transactions, not rich retrieval. Yet advanced search across random textual data, with capabilities such as edit distance, spell correction, synonyms etc, is crucial for imprecise matching.

A new solution, where offloading the database search to a true search platform, is used in complement with the solid transactional security of the database still in use, has proved to be the cheapest and fastest approach to clean, unified data with easy and precise access.

Toolbox for Data Cleansing

The search-based data cleansing solution has been con-

structed as a Service Oriented Architecture, with the following groups of capabilities across multiple languages:

1. Structured data search, where data is extracted from a database in a model-sensitive way, so that the fields are properly tagged and can be used in a biased way, depending on the matching logics. This basically entails contextual awareness for the search engine, where localities are differentiated from street names and postal addresses are differentiated from physical addresses. Addresses are typically hierarchical, for example a part of the address may be <primary> 525 Collin Street <sub> Level 27, South Tower <sub> <primary>, and it is useful to be able to weight the primary part more heavily, or to search for a South Tower on the Collin Street. In effect, the advanced search engine is able to model the data structure in its memory, rather than normalize it too early and thus lose valuable contextual info.

2. Dictionaries of names, phrases, or other key concepts are used to detect spelling errors and variations. Known variants of names are used by the search engine to ensure matching in case a name has many alternative spellings. An advanced search engine will take this into account when performing matching. In addition to spelling variations, typing errors might also occur in the original documents and in the user queries. These are handled by the spelling correction mechanism to ensure correct matching.

3. Character normalization is used to solve the challenge caused by diacritical characters (á, à, ä, å, etc.). Diacritical characters are often typed wrong or omitted. By performing character normalization, all diacritical characters are mapped to one normalized form on both query and content side, ensuring that one can obtain exact matches despite any differences in the use of diacritics.

4. Fuzzy and phonetic matching is used at times where character normalization and dictionary approaches do not give us 100% match. Fuzzy matching applies functions such as spell checking, lemmatization, stemming and synonyms to ensure that the search engine recognizes words and terminology that are considers alike. On the other hand, phonetic matching will detect words and names that sound similar although spelled differently (e.g. “Cheap” and “Sheep”).

Linguistic Tool Examples

1. Search for words anywhere within a search field (e.g. “John Lucas Dr”, “Lucas John Dr”)
2. Synonym dictionaries (e.g. ‘Panther St’ and ‘Leopard St’ or ‘Homestead’ and ‘Station’)
3. Edit distance corrections through Levenstein’s algorithm (e.g. missing letters in ‘Melborne’ instead of ‘Melbourne’)
4. Bi-gram and N-gram support (e.g. recognising that Westlake is very similar to “West Lake”)
5. Geo-matching support (geographical awareness)

5. Data Corrections, where the power of advanced linguistics of free-text search is used to define and apply rules that fix addresses and names. The linguistic tools in use include spellchecking, phonetics, bi-grams, n-grams and wild-carding, proximity boosting and forward or-boosting. In short, these tools first identify ambiguous overlapping data, they subsequently introduce a controlled level of linguistic chaos (by breaking the words up in parts, for example, or dropping letters) in this data, finally, they use search engine’s fuzzy matching and flexible relevancy ranking to tidy up and find the correct normalized version of the data.

To ensure secure identification, these matching mechanisms should be employed across multiple fields of the merging systems. This is used to help decide if two similar looking records are the same person/company, for example: let us say that the last names are identical and the first names close (Pam vs. Pamela), then matching the address-field might help determine if the two records are in fact the same person/company.

Other tools that have proved central in this data improvement exercise involve easy management of synonym dictionaries through web-based tools for business-managers, and reporting tools to detail the types of address errors that are not being detected and corrected by the system.

On the query side, automated functionality to try alternative queries is being used. It has proved useful that the system allows a 'query completion' feature, where possible results are proposed or "auto-completed" rather than the operator having to type the entire word. Use of this feature is desirable as it improves the accuracy and decreases the time required to enter an address by an operator, however each key press generates a query to the underlying engine, thus increasing query rates and scalability requirements. Given the potentially large number of operators with such auto-complete capability, it is important that the system scales linearly and inexpensively.

How Search Platforms Cleanse Data

Parsing

The initial parsing stage can potentially generate multiple interpretations of a given textual address. In addition, a single address can generate multiple interpretations. An example could be "44 Westlake Drive, River Hills". This could mean: 44 West Lake Dr, River Hills OR 44 Westlake Dr, Riverhills OR 44 Westlake Dr, River Hills (as originally given). The detailed output of the last of these parses is displayed in the box on the right.

Parsed Output

Streetname: Westlake
Streettype: Dr
Streetnum: 44
Locality: River hills

To solve this problem, the search platform includes a variety of parsers:

- RMB Parser
- Level Parser (multiple levels) – e.g. Level 5, EMI Building, 160 Anne st, Brisbane
- Simple address parser
- Property address parser
- Plan address parser
- Term splitter parser – generates alternate interpretations of an address based on words such as "westlake" by splitting and / or joining words.

Matching

Matching utilises a range of underlying features of the search platform. These features are key to the performance and ease-of-implementation of these matchers. They ensure the ability to deal with structured textual data, and thus enhance the platform's ability to delivery highly accurate "fuzzy" matching capabilities.

Exact matcher: As the name suggests, this will exactly match each of the input criteria against the structured data within the search index.

Advanced text matcher: Utilising a range of built-in search features such as:

- Levenstein edit distance algorithms – will automatically try 'brisbane' if only 'brsbane' is given (letter missing)
- Word synonyms – Will automatically try a range of words that are similar in meaning to the given word – for example: "fern tree gully", "fern tree gulch"
- Words "anywhere" within a text field – "john lucas", "lucas johns" can be easily matched
- Remove parts of query matcher: Removes elements of a query to determine if parts are completely incorrect. This is configurable in the order in which items are removed.
- Remove locality
- Remove postcode

- Wildcarding matcher: Remove sections of the words and replace these with wildcards
- Sounds like matcher: Generate soundex-like summaries of the given textual words and match these. For example: "Computa Dr" à "Cmpta". Search platforms utilise a pluggable 'soundslike' framework currently utilising the double-metaphone algorithm that is believed to be higher quality than standard Soundex. Nysiis and Soundex are also available, and other algorithms can be plugged in within 1-2 days.
- Ngram matcher: A matching approach that was not

included due to time constraints is the n-gram matcher, in which original words are broken up into ngrams by the search index (e.g. the word "west lake" would generate 3-letter ngrams of [wes, est, lak, ake]). When the user types in "westlake" the matcher would match ngrams such as "wes est stl lak ake". The "stl" ngram is the only ngram that would not match, indicating that Westlake is very similar to "West Lake" as all other ngrams did match.

- Geo matcher: The built-in ability to 'understand' the location of a given address, for example to recognise that an address for a suburb is 'near' another suburb can be utilised to generate matches that would otherwise not be possible.

Summary

The telco in this case has had two main goals:

The primary goal was to reduce error in addressing that leads to downstream costs such as re-work, returned mail and address sanitisation.

The secondary goal was to Improve the accuracy with which correct addresses are proposed in the case where an exact match finds no address.

Both of these goals were achieved in a cost- and time-effective way through the use of a rich data cleansing capability of a search platform.

Ultimately, the ability to perform an automatic data cleansing with significant success requires 'fuzzy' matching; this capability, in turn, revolves around the underlying functionality inherent in the search platform. Search engines provide a wide range of "out of the box" features that support fuzzy text matching such as "any word location", synonyms, lemmatization, ngrams, geomatching and Levenstein edit-distance. These features are not trivial to implement and provide considerable underlying power to the search for an accurate match to an inexact address specification.

Search platforms also contain a range of matching-related features that are business-user oriented, such as the interactive web-based "search business centre" allowing the reporting of unsuccessful searches, and subsequent easy tuning. This tool also allows the modification of synonym and other lists, essentially providing a complete web-based front end to management of this integrated data improvement tool.

In summary, search platform's matching features and quality, provided this telco with an innovative and effective solution to the problem of data cleansing and fuzzy matching.

A rose by any other name... How many ways can users spell Britney Spears?

488941 britney spears
40134 brittany spears
36315 brittney spears
24342 britany spears
7331 britny spears
6633 briteny spears
2696 britteny spears
1807 briney spears
1635 brittny spears
1479 brintey spears
1479 britanny spears
1338 britiny spears
1211 britnet spears
1096 britiney spears
991 britaney spears
991 britnay spears
811 brithney spears
811 britney spears
664 birtney spears
664 brintney spears
664 briteney spears
601 bitney spears
601 brinty spears
544 brittaney spears
544 brittnay spears
364 britey spears
364 brittiny spears
329 brtney spears
269 bretney spears
269 britneys spears
244 britne spears
244 brytney spears
220 breatney spears
220 britiany spears
199 britnney spears
163 britnry spears
147 breatny spears

147 brittiney spears
147 britty spears
147 brotney spears
147 brutney spears
133 britteney spears
133 briyney spears
121 bittany spears
121 bridney spears
121 britainy spears
121 britmey spears
109 brietney spears
109 brithny spears
109 britni spears
109 brittant spears
98 bittney spears
98 brithey spears
98 brittiany spears
98 btitney spears
89 brietny spears
89 brinety spears
89 brintny spears
89 britnie spears
89 brittey spears
89 brittnet spears
89 brity spears
89 ritney spears
80 bretny spears
80 britnany spears
73 brinteny spears
73 brittainy spears
73 britney spears
66 brintany spears
66 britnery spears
59 briitney spears
59 britinay spears
54 britneay spears
54 britner spears

54 britney's spears
54 britnye spears
54 britt spears
54 brttany spears
48 bitany spears
48 briny spears
48 brirney spears
48 britant spears
48 britnety spears
48 brittanny spears
48 brttney spears
44 birttany spears
44 brittani spears
44 brityney spears
44 brtitney spears
39 brienty spears
39 brritney spears
36 bbritney spears
36 briitany spears
36 britanney spears
36 briterny spears
36 britneey spears
36 britnei spears
36 britniy spears
32 britbey spears
32 britneu spears
29 britent spears
29 brittnany spears
29 britttany spears
29 btiney spears
26 birttney spears
26 breitney spears
26 brinity spears
26 britenay spears
26 britneyt spears
26 brittan spears
26 brittne spears

26 btittany spears
24 beitney spears
24 birteny spears
24 brightney spears
24 brintiny spears
24 britanty spears
24 britenny spears
24 britini spears
24 britnwy spears
24 brittni spears
24 brittnie spears
21 biritney spears
21 britany spears
21 biteny spears
21 bratney spears
21 britani spears
21 britanie spears
21 briteany spears
21 brittay spears
21 brittinay spears
21 brtany spears
21 brtiany spears
19 birney spears
19 brirtney spears
19 britnaey spears
19 britnee spears
19 britony spears
19 brittanty spears
19 britttney spears
17 birtny spears
17 brieny spears
17 brintty spears
17 brithy spears
17 brittanie spears
15 brinney spears
15 briten spears
15 briterney spears

15 britheny spears
15 britneny spears
15 brittamy spears
15 brittmey spears
15 brytnei spears
15 btirney spears
15 rittney spears
14 brinet spears
14 britneyy spears
14 britten spears
12 beritney spears
12 bretiny spears
12 briatny spears
12 brieney spears
12 brinany spears
12 britaany spears
12 britan spears
12 britine spears
12 britnea spears
12 britnes spears
12 britnez spears
12 britnny spears
12 brittenay spears
12 brittneys spears
12 brittony spears
12 brteny spears
12 btitany spears
12 btittney spears
12 btriney spears
12 btritney spears
12 rittany spears
11 braitney spears
11 brettney spears
11 britamy spears
11 britery spears
11 britnary spears
11 brittent spears

11 bruttney spears
11 pritny spears
10 bitaney spears
10 brenty spears
10 bristney spears
10 britay spears
10 britinny spears
10 brittaany spears
10 brittanys spears
10 brittini spears
10 brittniy spears
10 brtieny spears
10 brutany spears
9 bitteny spears
9 brinttany spears
9 britanay spears
9 britinany spears
9 britn spears
9 britnew spears
9 britneyn spears
9 britrney spears
9 brtiny spears
9 brtittney spears
9 brtny spears
9 brytny spears
9 rbitney spears
8 birtiny spears
8 bithney spears
8 brattany spears
8 breitny spears
8 breteny spears
8 brightny spears
8 brintay spears
8 brinttey spears
8 briotney spears
8 britanys spears
8 britley spears

8 britneyb spears
8 britnrey spears
8 britnty spears
8 brittner spears
8 brottany spears
7 baritney spears
7 birntey spears
7 biteney spears
7 bitiny spears
7 breateny spears
7 brianty spears
7 brintye spears
7 britianny spears
7 britly spears
7 britnej spears
7 britneyu spears
7 britniey spears
7 britnnay spears
7 brittian spears
7 briyny spears
7 brrittany spears
7 brttiney spears
7 btiteny spears
7 btrittany spears
6 beritny spears
6 bhritney spears
6 birthney spears
6 breathney spears
6 breaty spears
6 bretany spears
6 briatany spears
6 brint spears
6 britenney spears
6 britian spears
6 britinty spears
6 britiitney spears
6 britnsy spears

6 britrey spears
6 britsny spears
6 brittine spears
6 brittnry spears
6 brittsny spears
6 brtanny spears
6 brtittany spears
6 bruttany spears
6 brytany spears
6 btitny spears
6 pretny spears
6 ritany spears
5 bbritany spears
5 bbrittney spears
5 biitney spears
5 bitny spears
5 breathny spears
5 breney spears
5 brethney spears
5 brettny spears
5 bridny spears
5 brigney spears
5 briittney spears
5 brinaty spears
5 brinay spears
5 brinetty spears
5 brinitty spears
5 brinny spears
5 brintery spears
5 britary spears
5 britbney spears
5 brithany spears
5 britieny spears
5 britiey spears
5 britnt spears
5 britnys spears
5 britrny spears

About FAST SBP (Search Best Practices)

FAST SBP is a highly focused transfer of search knowledge and experience from FAST to its prospects and customers. FAST SBP workshops aim to help enterprises realize the full potential of search, by creating optimal strategic, functional and technical roadmaps, delivered in the form of business model, solution and architecture designs.

For any feedback or questions related to this paper, please contact us at sbp@fastsearch.com.

FAST™
www.fastsearch.com
info@fastsearch.com

Regional Headquarters

The Americas
+1 781 304 2400

Europe, Middle East & Africa (EMEA)
+47 23 01 12 00

Japan
+81 3 5511 4343

Asia Pacific
+612 9929 7725

SWP.019.B.02.102406