JRC-Names-Retrieval

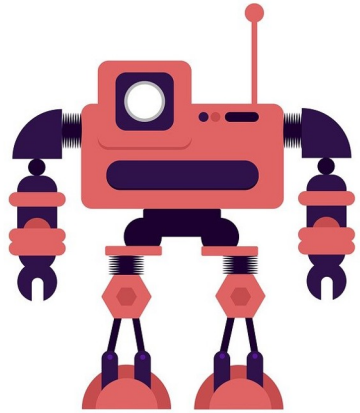
A Standardized Benchmark for Name Search

BABEL STREET

Philip Blair & Kfir Bar

22-24 May 2024 | Torino, IT

Name Search is a Common Problem



Query

Match?

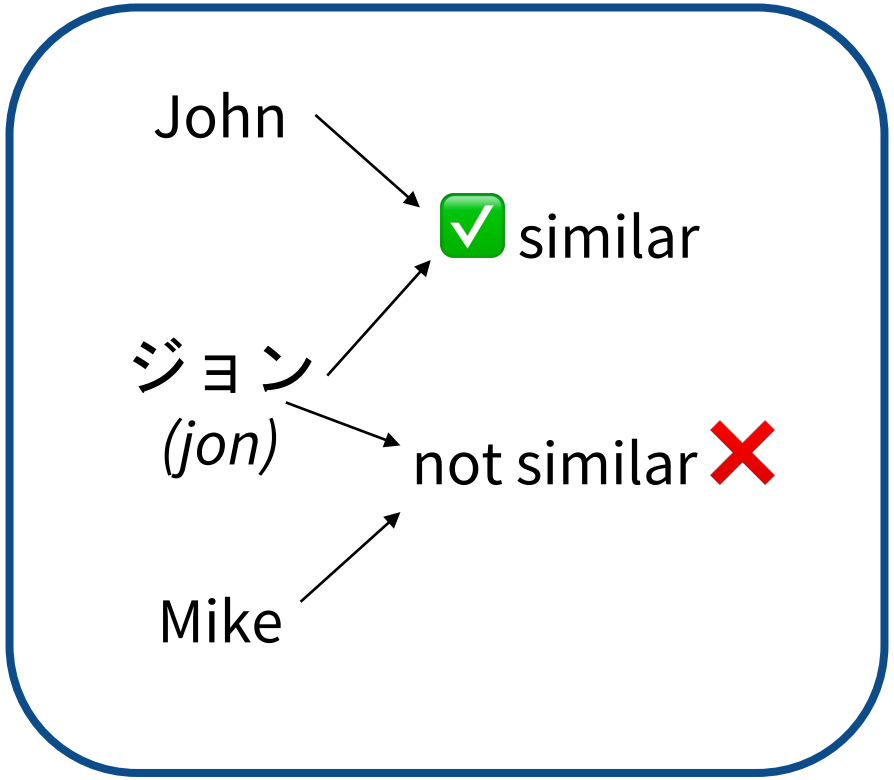
Database

Sanction List

Useful across many industries:

- Banks
- Electronic Health Records
- Governments
- …and more!

Cross-Lingual Name Search is Hard!



What types of variations are we interested in?

- **Phonetically/Lexically Similar Names,** e.g. “John Smith” and “Jon Smith”, or “Phillippe Jones” and “Felipe Jones”
- **Initials,** e.g. “George Walker Bush” and “George W. Bush”
- **Missing Components,** e.g. “John A. Macdonald” and “John Macdonald”
- **Similar Names/Nicknames,** e.g. “William Clinton” and “Bill Clinton”, or “Michael Scott” and “Prison Mike”
- **Out-of-Order Components,** e.g. “François Mitterand” and “MITTERAND François”
- **Titles,** e.g. “Sir Tony Blair” and “Tony Blair”
- **Different Writing Scripts,** e.g. “Hu Jintao” and “胡锦涛”

Particularities for Organizations

- **Semantically Similar Components,** e.g. “Raven Train Company” and “Raven Locomotive, Inc.”.
- **Semantically Similar Components in Different Languages,** e.g. “株式会社京都アニメーション” (*Kabushiki-gaisha Kyōto Animēshon*) and “Kyoto Animation Co. Ltd.”.

Problem:
No Benchmark!

Why a Standard Benchmark is Useful

1. Consistency
   - All systems measure the same query/result pairs
   - Future works able to compare performance without needing to run existing systems
2. Multilinguality
   - We can design multiple (balanced) data splits to look at different types of multilingual support
   - Four writing script combinations published:
     - Latin, Cyrillic, and Arabic
     - Latin and Hani

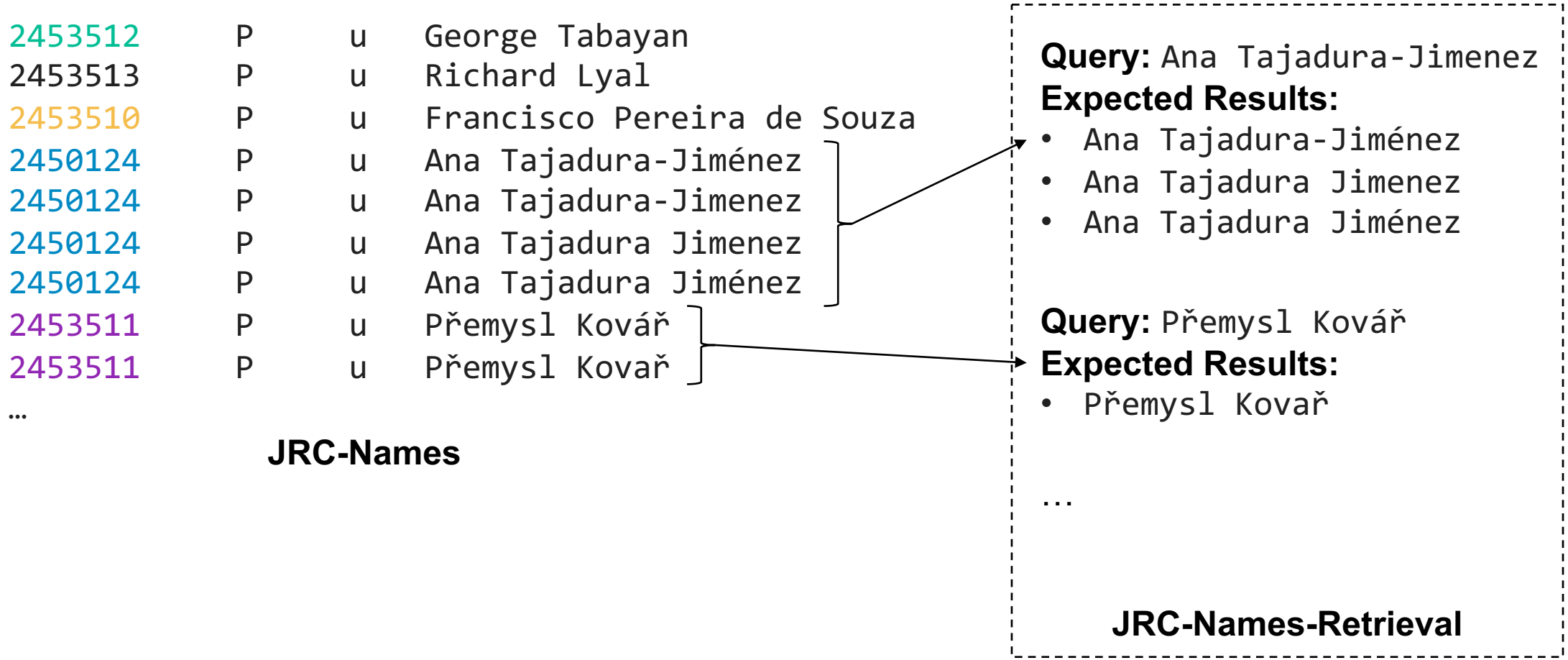
     Realistic
     - Hangul and Hebrew

     Stress Test
     - Devanagari and Kana (Hiragana+Katakana)

Background: JRC-Names


```
2453512     P     u   George Tabayan
2453513     P     u   Richard Lyal
2453510     P     u   Francisco Pereira de Souza
2450124     P     u   Ana Tajadura-Jiménez
2450124     P     u   Ana Tajadura-Jimenez
2450124     P     u   Ana Tajadura Jimenez
2450124     P     u   Ana Tajadura Jiménez
2453511     P     u   Přemysl Kovář
2453511     P     u   Přemysl Kovař
…
```

JRC-Names-Retrieval: Basic Idea



Data Statistics

Type	Script	# Names	# Clusters
PER	Latn+Cyrl+Arab	1,178,357	737,361
PER	Latn+CJK	97,077	38,268
PER	Hang+Hebr	1,331	640
PER	Deva+Kana	1,792	454
ORG	Latn+Cyrl+Arab	30,113	2,942
ORG	Latn+CJK	27,702	2,857
ORG	Hang+Hebr	494	224
ORG	Deva+Kana	380	122

Table 1: JRC-Names-Retrieval training subsplit sizes for each entity type/script combination split.

Entity Type	Script	Database Size	# Queries	Avg. # Results
PER	Latn+Cyrl+Arab	5602	1885	2.97
PER	Latn+CJK	1457	268	5.43
PER	Hang+Hebr	316	266	1.18
PER	Deva+Kana	550	160	3.43
ORG	Latn+Cyrl+Arab	841	152	5.53
ORG	Latn+CJK	738	135	5.51
ORG	Hang+Hebr	119	97	1.23
ORG	Deva+Kana	90	38	2.36

Table 2: JRC-Names-Retrieval evaluation subsplit sizes for each entity type/script combination split.

Baselines

Preexisting Baselines

String-Based

- Double Metaphone
  - Approximates pronunciation of a string (European language-centric)
  - Index names by taking bigrams
- Lucene Fuzzy-Query
  - Damerau-Levenshtein edit distance

Neural Network-Based

- Pretrained ByT5
- SimCSE
- Sentence-T5

Additional Baseline: Metric Learning

- Architecture: ByT5 encoder, followed by linear layer and mean pool (across all characters) into single vector

$$\mathcal{L}_{contrast}(n_a, n_+) = \frac{\delta(n_a, n_+)^2}{2}, \; \mathcal{L}_{contrast}(n_a, n_-) = \frac{\left[m - \delta(n_a, n_-)^2\right]^+}{2}$$

$$\mathcal{L}_{triplet}(n_a, n_+, n_-) = \left[\delta(n_a, n_+)^2 - \delta(n_a, n_-)^2 + m\right]^+$$

Figure 1: Loss functions used in our metric learning baseline.

- Negative mining based on NSEEN (Fakhraei et al., 2019)
  - After each epoch, create an approximate nearest neighbor index with each name
  - Look up each query in the ANN index and take the closest negative(s)

Results (Person, Realistic)

Latin, Cyrillic, and Arabic

Algorithm	MAP	Recall@1	Recall@5	Recall@10	Recall@50	Recall@100
Double Metaphone†	0.16427	0.08546	0.16556	0.16968	0.17626	0.18214
+ pre-transliteration	0.91355	0.50809	0.89106	0.94381	0.97900	0.98530
Sentence-T5†	0.29894	0.17942	0.30980	0.32471	0.35936	0.37559
+ pre-transliteration	0.84549	0.48011	0.83323	0.88937	0.94981	0.96674
ByT5	0.06841	0.04018	0.07354	0.09702	0.15778	0.19513
+ pre-transliteration	0.23765	0.17140	0.24862	0.28274	0.38930	0.44801
SimCSE†	0.22508	0.11528	0.24852	0.29057	0.36891	0.39210
+ pre-transliteration	0.74304	0.42829	0.74217	0.80665	0.88724	0.91324
Lucene FuzzyQuery†	0.39133	0.24901	0.39832	0.41042	0.42456	0.428090
+ pre-transliteration	0.76186	0.44020	0.75026	0.80762	0.88085	0.89819
Ours (Latin-only)†	0.50 ± 0.01	0.289 ± 0.00	0.47 ± 0.02	0.50 ± 0.03	0.62 ± 0.01	0.66 ± 0.01
+ pre-transliteration	0.93 ± 0.01	0.513 ± 0.00	0.91 ± 0.01	0.95 ± 0.00	0.97 ± 0.00	0.98 ± 0.00
Ours (La+Ar+Cy)	**0.95 ± 0.01**	**0.521 ± 0.01**	**0.93 ± 0.00**	**0.97 ± 0.00**	**0.99 ± 0.00**	**0.99 ± 0.00**
+ pre-transliteration	0.94 ± 0.01	0.520 ± 0.00	0.92 ± 0.01	0.96 ± 0.01	0.98 ± 0.00	0.98 ± 0.00

Latin and Hanzi

Algorithm	MAP	Recall@1	Recall@5	Recall@10	Recall@50	Recall@100
Double Metaphone†	0.31134	0.10988	0.30951	0.31853	0.33370	0.33619
+ pre-transliteration	0.74852	0.17531	0.65205	0.77245	0.87325	0.91044
Sentence-T5†	0.31130	0.11126	0.30871	0.31692	0.33371	0.34055
+ pre-transliteration	0.68389	0.17071	0.60647	0.69882	0.79372	0.83053
ByT5	0.07486	0.03837	0.07718	0.10056	0.18022	0.24148
+ pre-transliteration	0.15977	0.06965	0.14502	0.16835	0.27705	0.37376
SimCSE†	0.42119	0.17357	0.41617	0.43420	0.45417	0.46238
+ pre-transliteration	0.63187	0.16978	0.55939	0.64179	0.73881	0.78750
Lucene FuzzyQuery†	0.26278	0.09944	0.26380	0.27437	0.28613	0.29670
+ pre-transliteration	0.58183	0.15951	0.52562	0.60317	0.65018	0.66013
Ours (Latin-only)†	0.45 ± 0.00	0.17 ± 0.00	0.44 ± 0.00	0.45 ± 0.00	0.46 ± 0.00	0.48 ± 0.00
+ pre-transliteration	0.73 ± 0.03	0.17 ± 0.00	0.64 ± 0.03	0.73 ± 0.03	0.83 ± 0.03	0.86 ± 0.03
Ours (Latin+CJK)	**0.88 ± 0.01**	**0.183 ± 0.00**	**0.76 ± 0.00**	**0.88 ± 0.01**	**0.94 ± 0.01**	**0.95 ± 0.00**
+ pre-transliteration	0.82 ± 0.00	0.180 ± 0.00	0.72 ± 0.00	0.83 ± 0.00	0.90 ± 0.00	0.93 ± 0.00

Takeaway:
Fine-tuned ByT5 neural network performs comparably to systems which rely on transliteration.

Results (Organization, Realistic)

Latin, Cyrillic, and Arabic

Algorithm	MAP	Recall@1	Recall@5	Recall@10	Recall@50	Recall@100
Double Metaphone†	0.35200	0.11590	0.34803	0.37544	0.44287	0.47522
+ pre-transliteration	0.61014	0.15976	0.53487	0.63575	0.78958	0.83476
Sentence-T5†	0.40391	0.13180	0.38311	0.43399	0.51118	0.54079
+ pre-transliteration	**0.61739**	**0.16086**	**0.53640**	**0.64846**	**0.79956**	**0.85362**
ByT5	0.08393	0.03191	0.06985	0.10614	0.24485	0.33048
+ pre-transliteration	0.10012	0.03355	0.08542	0.11469	0.22763	0.34737
SimCSE†	0.37026	0.13180	0.35461	0.40450	0.47401	0.50581
+ pre-transliteration	0.53519	0.15099	0.47588	0.55888	0.71107	0.76173
Lucene FuzzyQuery†	0.26533	0.10186	0.26447	0.28202	0.31272	0.32149
+ pre-transliteration	0.38923	0.11754	0.35471	0.42708	0.48684	0.50164
Ours (Latin-only)†	0.41 ± 0.02	0.13 ± 0.00	0.37 ± 0.02	0.42 ± 0.02	0.54 ± 0.03	0.61 ± 0.04
+ pre-transliteration	0.57 ± 0.03	0.15 ± 0.00	0.51 ± 0.03	0.59 ± 0.03	0.73 ± 0.04	0.79 ± 0.04
Ours (La+Ar+Cy)	0.49 ± 0.04	0.14 ± 0.00	0.44 ± 0.04	0.52 ± 0.05	0.67 ± 0.03	0.75 ± 0.04
+ pre-transliteration	0.49 ± 0.01	0.14 ± 0.00	0.43 ± 0.01	0.51 ± 0.02	0.65 ± 0.01	0.73 ± 0.02

Latin and Hanzi

Algorithm	MAP	Recall@1	Recall@5	Recall@10	Recall@50	Recall@100
Double Metaphone†	0.43352	0.13000	0.41173	0.46605	0.55247	0.59346
+ pre-transliteration	0.47043	0.13580	0.43247	0.49914	0.64037	0.69901
Sentence-T5†	0.51244	0.14235	0.47728	0.52914	0.59580	0.60963
+ pre-transliteration	**0.54248**	**0.14963**	**0.49852**	**0.55840**	**0.65346**	**0.69815**
ByT5	0.08231	0.02790	0.06728	0.09321	0.22617	0.34309
+ pre-transliteration	0.08535	0.02605	0.07432	0.09802	0.22506	0.35148
SimCSE†	0.47097	0.14136	0.43519	0.49074	0.57617	0.60457
+ pre-transliteration	0.48958	0.14593	0.44420	0.50593	0.63889	0.67284
Lucene FuzzyQuery†	0.34896	0.10309	0.34235	0.37568	0.40037	0.40901
+ pre-transliteration	0.38029	0.10889	0.37691	0.40901	0.44728	0.45593
Ours (Latin-only)†	0.48 ± 0.02	0.14 ± 0.00	0.45 ± 0.03	0.49 ± 0.03	0.58 ± 0.01	0.64 ± 0.01
+ pre-transliteration	0.50 ± 0.03	0.14 ± 0.00	0.46 ± 0.03	0.51 ± 0.04	0.62 ± 0.02	0.68 ± 0.01
Ours (Latin+Hanzi)	0.18 ± 0.00	0.07 ± 0.00	0.16 ± 0.00	0.20 ± 0.00	0.33 ± 0.00	0.42 ± 0.00
+ pre-transliteration	0.18 ± 0.00	0.07 ± 0.00	0.17 ± 0.00	0.21 ± 0.00	0.33 ± 0.00	0.42 ± 0.01

Takeaway:
Sentence-T5 is a hard baseline to beat when transliteration is available, but fine-tuning ByT5 is better when it is not.

Thank You!