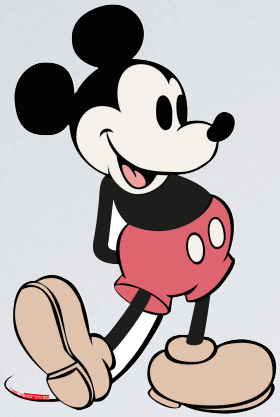
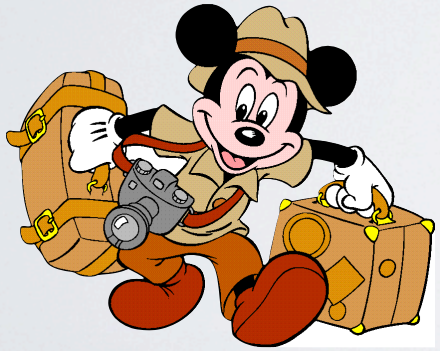





NAME MATCHING WITH PHYLOGENIES

Nicholas Andrews, Jason Eisner, Mark Dredze

Martin Freeman

Martin Freeman

M Freeman

Martin Freedman

Marty Freemen

Marty Freeman

Martin F

Entity Linking

Coref Resolution

Martin Freeman

M Freeman

Martin Freedman

Marty Freemen

Marty Freeman

Martin F

STRING COMPARISON

- Levenshtein distance
  - Edit distance between two strings
- Jaro Winkler
  - Measures matching characters and transpositions

STRING COMPARISON

- Levenshtein distance
  - Edit distance between two strings
- Jaro Winkler
  - Measures matching characters and transpositions


Mark Dredze vs. Mark Drezde (e.g. typo, name variant)


Mark Dredze vs. Benjamin Van Durme

NAME VARIATION

- Nicknames: Benjamin Van Durme vs. Ben Van Durme
- Aliases: Caryn Elaine Johnson vs. Whoopi Goldberg
- Chinese Names: Zhang Wei vs. Wei Zhang
- Arab Names:
  Muhammad ibn Saeed ibn Abd al-Aziz al-Filasteeni
  vs. Muhammad
  vs. Abu Kareem

OUR GOAL

LEARN HOW TO MATCH NAMES

FINITE STATE TRANSDUCERS

- Probabilistic finite state transducers encode a probability distribution over strings given a string
- Character operations: copy, substitute, delete, insert
- Train parameters on name pairs

Latent-Variable Modeling of String Transductions with Finite-State Methods*

Markus Dreyer and **Jason R. Smith** and **Jason Eisner**
Department of Computer Science
Johns Hopkins University
Baltimore, MD 21218, USA
{markus,jsmith,jason}@cs.jhu.edu

Abstract

String-to-string transduction is a central problem in computational linguistics and natural language processing. It occurs in tasks as diverse as name transliteration, spelling correction, pronunciation modeling and inflectional morphology. We present a conditional log-linear model for string-to-string transduction, which employs overlapping features over latent alignment sequences, and which learns latent classes and latent string pair regions from incomplete training data. We evaluate our approach on morphological tasks and demonstrate that latent variables can dramatically improve results, even when trained on small data sets. On the task of generating morphological forms, we outperform a baseline method reducing the error rate by up to 48%. On a lemmatization task, we reduce the error rates in Wicentowski (2002) by 38–92%.

1 Introduction

A recurring problem in computational linguistics and language processing is transduction of character strings, e.g., words. That is, one wishes to model some systematic mapping from an input string x to an output string y. Applications include:

- *phonology*: underlying representation ↔ surface representation
- *orthography*: pronunciation ↔ spelling
- *morphology*: inflected form ↔ lemma, or differently inflected form
- *fuzzy name matching* (duplicate detection) and *spelling correction*: spelling ↔ variant spelling
- *lexical translation* (cognates, loanwords, transliterated names): English word ↔ foreign word

We present a configurable and robust framework for solving such word transduction problems. Our results in morphology generation show that the presented approach improves upon the state of the art.

2 Model Structure

A weighted edit distance model (Ristad and Yianilos, 1998) would consider each character in isolation. To consider more context, we pursue a very natural generalization. Given an input x, we evaluate a candidate output y by moving a sliding window over the aligned (x, y) pair. More precisely, since many alignments are possible, we sum over all these possibilities, evaluating each alignment *separately*.[1]

At each window position, we accumulate log-probability based on the material that appears within the current window. The window is a few characters wide, and successive window positions *overlap*. This stands in contrast to a competing approach (Sherif and Kondrak, 2007; Zhao et al., 2007) that is inspired by phrase-based machine translation (Koehn et al., 2007), which *segments* the input string into substrings that are transduced *independently*, ignoring context.[2]

*This work was supported by the Human Language Technology Center of Excellence and by National Science Foundation grant No. 0347822 to the final author. We would also like to thank Richard Wicentowski for providing us with datasets for lemmatization, and the anonymous reviewers for their valuable feedback.

[1] At the other extreme, Freitag and Khadivi (2007) use no alignment; each feature takes its own view of how (x, y) relate.

[2] We feel that this independence is inappropriate. By analogy, it would be a poor idea for a language model to score a string highly if it could be *segmented* into independently frequent n-grams. Rather, language models use overlapping n-grams (indeed, it is the language model that rescues phrase-based MT from producing disjointed translations). We believe phrase-based MT avoids overlapping phrases in the *channel* model only because these would complicate the modeling of reordering (though see, e.g., Schwenk et al. (2007) and Casacuberta (2000)). But in the problems of section 1, letter reordering is rare and we may assume it is local to a window.

William Ronald Dodds Fairbairn Ronald Fairbairn

- Ideal: matched name pairs

William Ronald Dodds Fairbairn

Ronald Fairbairn

- Ideal: matched name pairs

Ronald Fairbairn

W. R. D. Fairbairn

William Ronald Dodds Fairbairn

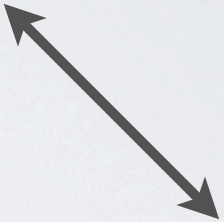
- Sets of matching names

William Ronald Dodds Fairbairn

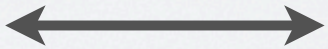
Ronald Fairbairn

W. R. D. Fairbairn

William Ronald Dodds Fairbairn

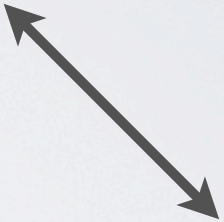


Ronald Fairbairn



W. R. D. Fairbairn

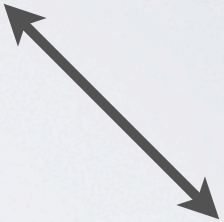
William Ronald Dodds Fairbairn



Ronald Fairbairn

W. R. D. Fairbairn

William Ronald Dodds Fairbairn



Ronald Fairbairn

W. R. D. Fairbairn

William Ronald Dodds Fairbairn

Ronald Fairbairn

- Ideal: matched name pairs

Ronald Fairbairn

W. R. D. Fairbairn

William Ronald Dodds Fairbairn

- Sets of matching names

William Ronald Dodds Fairbairn

Ronald Fairbairn

- Ideal: matched name pairs

Ronald Fairbairn

W. R. D. Fairbairn

William Ronald Dodds Fairbairn

- Sets of matching names

John Wilkins

Mikhail Dobuzhinsky

Samuel Loyd

James Beach Wakefield

Mstislav Dobuzhinsky

James Wakefield

- Unorganized set of names

William Ronald Dodds Fairbairn

Ronald Fairbairn

- Ideal: matched name pairs

Ronald Fairbairn

W. R. D. Fairbairn

- Sets

Key Insight
Learn name phylogenies

John Wilkins

Mikhail Dobuzhinsky

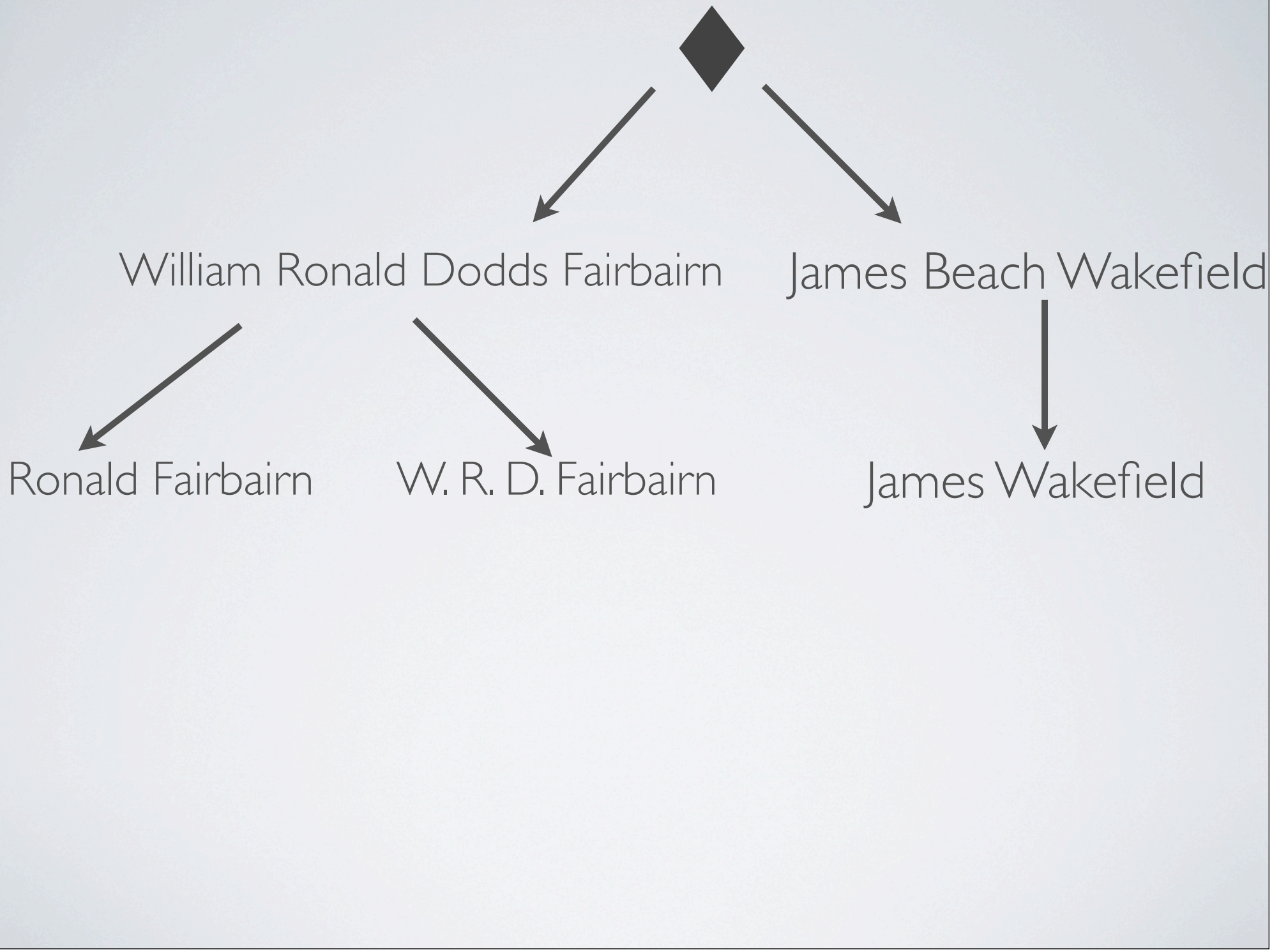
Samuel Loyd

James Beach Wakefield

Mstislav Dobuzhinsky

James Wakefield

- Unorganized set of names


William Ronald Dodds Fairbairn
James Beach Wakefield
Ronald Fairbairn
W. R. D. Fairbairn
James Wakefield

WHY A NAME PHYLOGENY?

- Aligns matching names for transducer
- Organizes names into connected components (clusters)
- We can jointly estimate a phylogeny and a mutation model (transducer)
  - A mutation model gives a phylogeny
  - A phylogeny provides training data for a mutation model

OUTLINE

- Generative model
- Inference
- Experiments

GENERATIVE MODEL

NAME VARIATION

- A generative model of strings that can explain observed name variation

  ...
  Mitt Romney
  President Barack Obama
  Barack Obama
  Secretary of State Hillary Clinton
  Hillary Clinton
  Barack Obama
  Clinton
  Obama
  ...

- What are the sources of variation for names?

GENERATIVE MODEL OF NAME VARIATION

- Suppose an author decides to write a name
  - Where do names come from?
    - Copy a previous mention
    - Mutate a previous mention
      - According to mutation model
    - Create a new name

COPY A PREVIOUS MENTION

- Select a previous mention at random (uniformly)
- Copy it with probability $1-\mu$

MUTATE PREVIOUS MENTION

- Select a previous mention at random (uniformly)
- Mutate it with probability μ
- Sample a new mutation from the mutation model given the mention

CREATE A NEW NAME

- Select the root of the phylogeny ♦ with probability proportional to α
- Sample a new name from a character language model

SUMMARY

- To generate the next mention
  - Pick an existing name mention w with probability $1/(\alpha + k)$
    - Copy w verbatim with probability $1 - \mu$
    - Mutate w with probability μ
  - Decide to talk about a new entity with probability $\alpha/(\alpha + k)$
    - Generate a name for it

INFERENCE

EM ALGORITHM

- E-step
  - Given mutation model θ, compute a distribution over phylogenies
- M-step
  - Re-estimate θ given marginal edge probabilities
    - Sum over alignments for all (x,y) string pairs via forward-backward
    - Each pair is training example weighted by the marginal probability

SUMMARY

- Learn a name matching algorithm
  - θ (transducer/mutation model)
  - Phylogeny: a means to an end
    - Part of the reason for a *distribution* over phylogenies
- Question: Is θ better than other name matching algorithms?
  - Can θ find matching names more accurately?

EXPERIMENTS

DATA

- English Wikipedia (2011) to create lists of name variants
  - Wikipedia redirects are human-curated pages to resolve common name variants to the correct page (unambiguously)
  - Use Freebase to restrict to redirects for Person entities
  - Applied some further filters to remove redirects that were clearly not names (e.g. numbers)
  - Use LDC Gigaword to obtain a frequency for each name variant

Khwaja Gharib Nawaz

Muinuddin Chishti

Khwaja Muin al-Din Chishti

Thomas Pynchon, Jr.

Thomas R. Pynchon Jr.

Khawaja Gharibnawaz Muinuddin Hasan Chisty

Thomas Ruggles Pynchon Jr..

Thomas Pynchon Jr.

Ghareeb Nawaz

Khwaja gharibnawaz

Thomas R. Pynchon

Khwaja Gharib Nawaz

Muinuddin Chishti

Khwaja Muin al-Din Chishti

Thomas Pynchon, Jr.

Thomas R. Pynchon Jr.

Khawaja Gharibnawaz Muinuddin Hasan Chisty

Thomas Ruggles Pynchon Jr..

Thomas Pynchon Jr.

Ghareeb Nawaz

Khwaja gharibnawaz

Thomas R. Pynchon

Our Algorithm

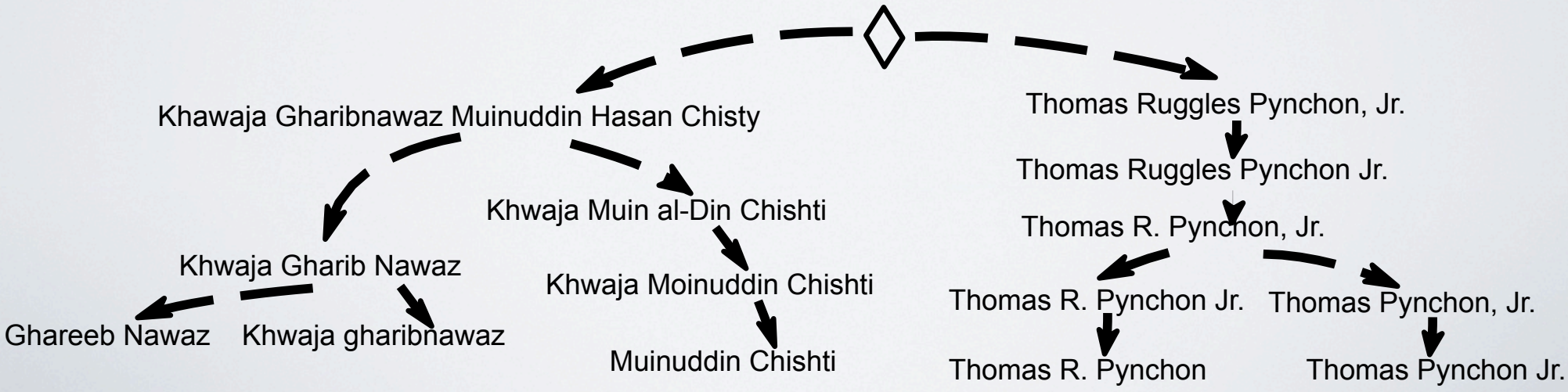
Our Algorithm

Our Algorithm

θ (Transducer)

Our Algorithm

θ (Transducer)



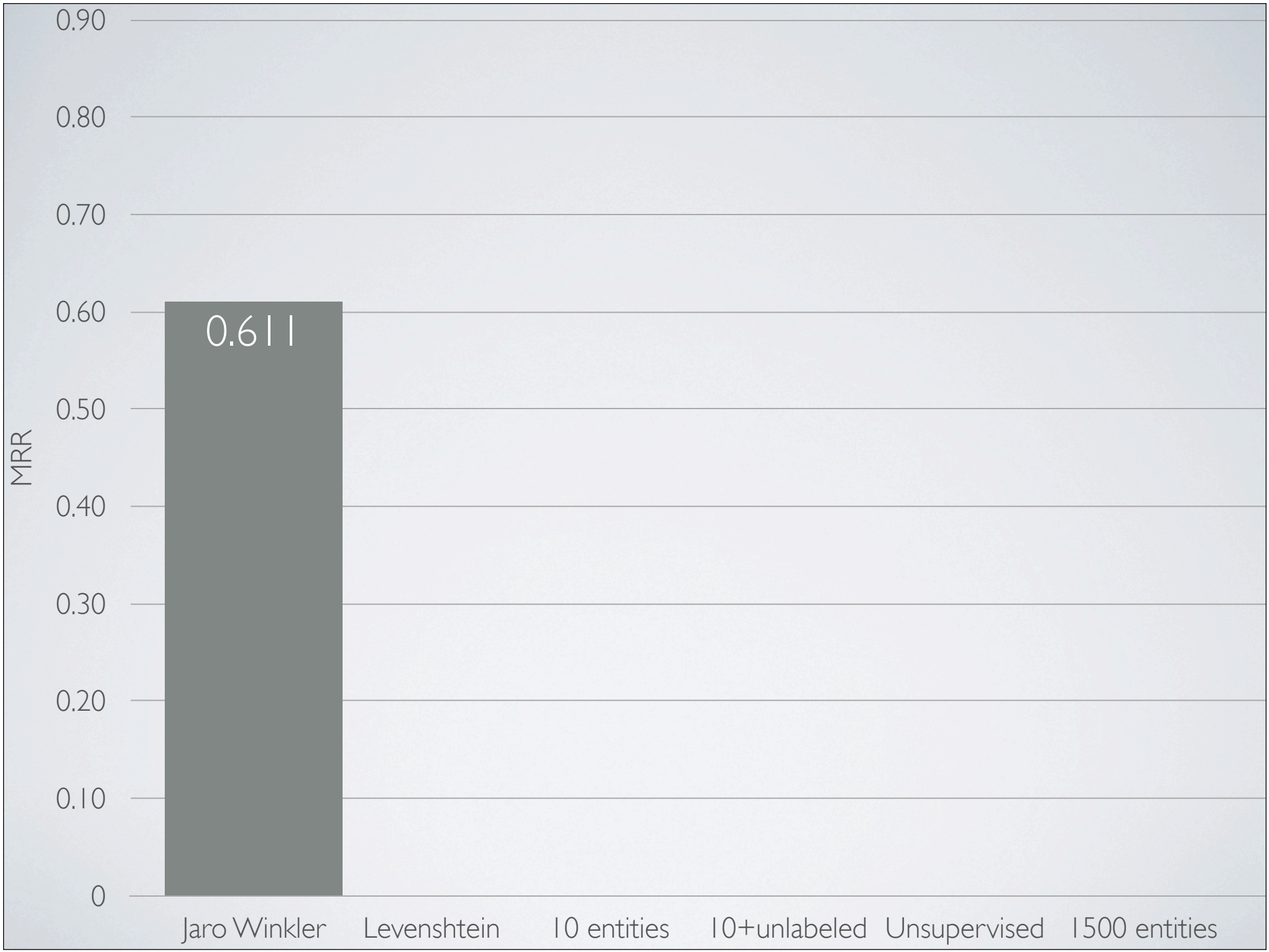
EXPERIMENT: RANKING

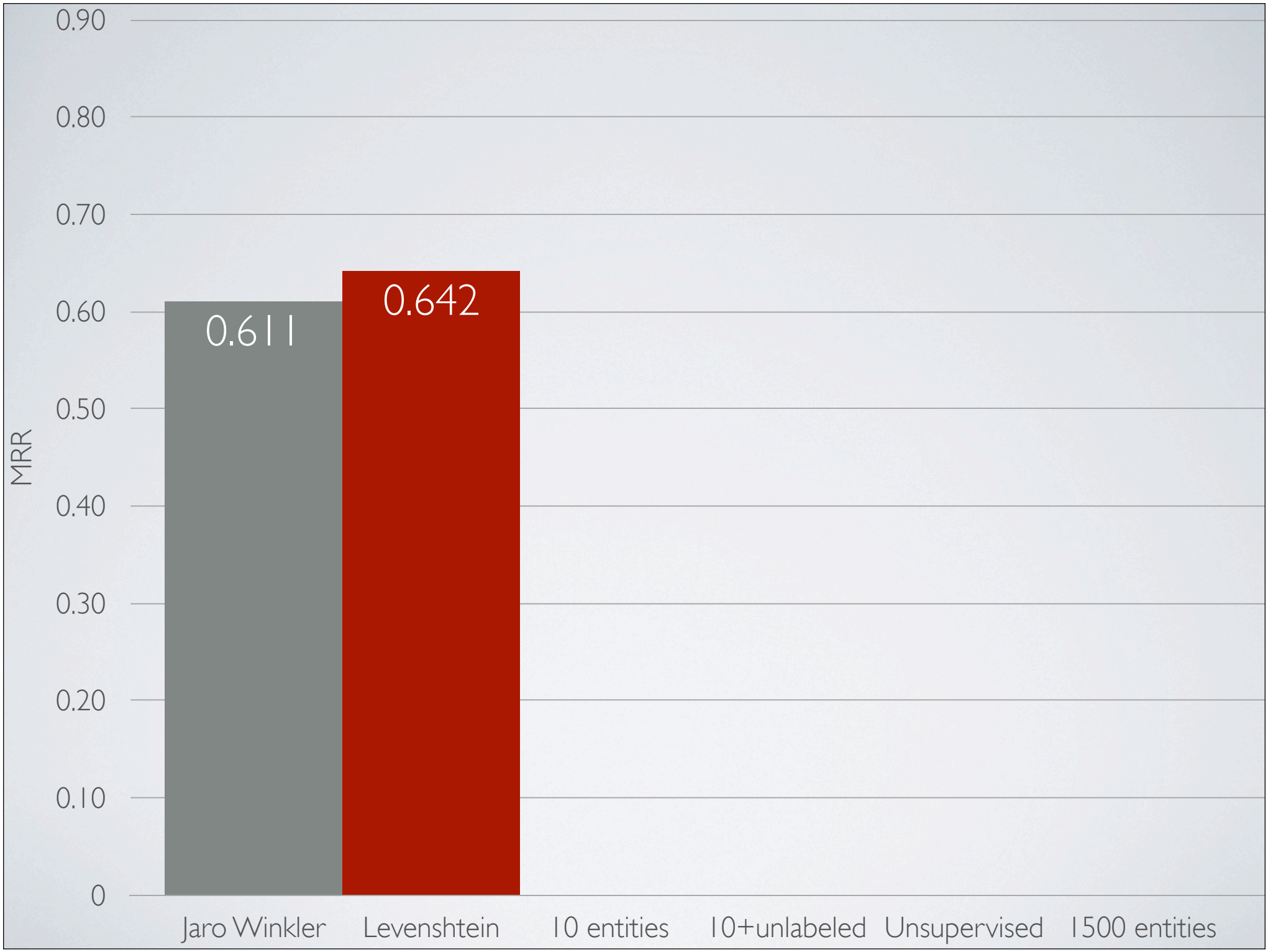
- Input: query (name)
- Output: ranked list of possible aliases
- Evaluation: where is correct alias in list?
  - Mean Reciprocal Rank (MRR) (higher is better)

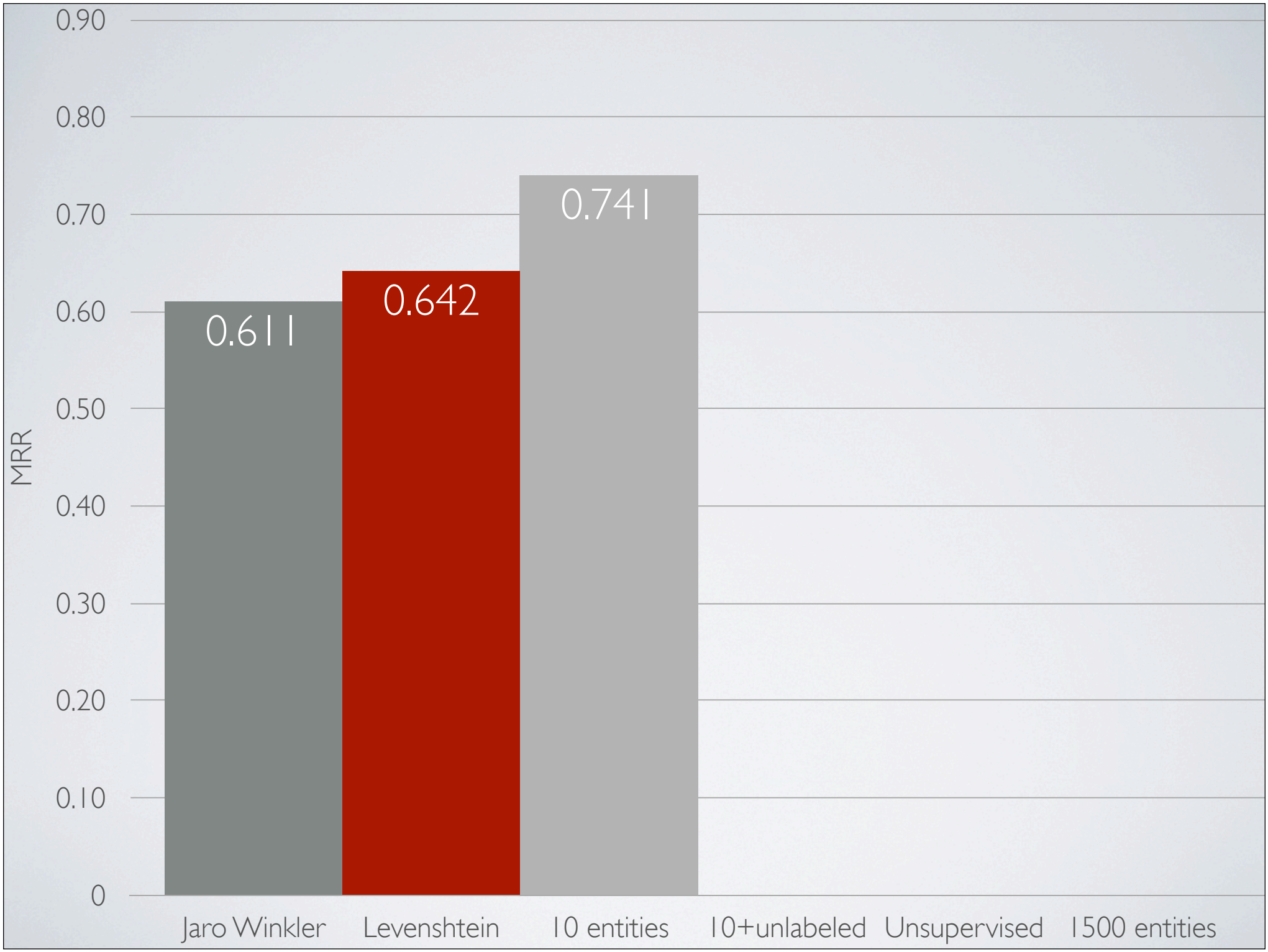
SETUP

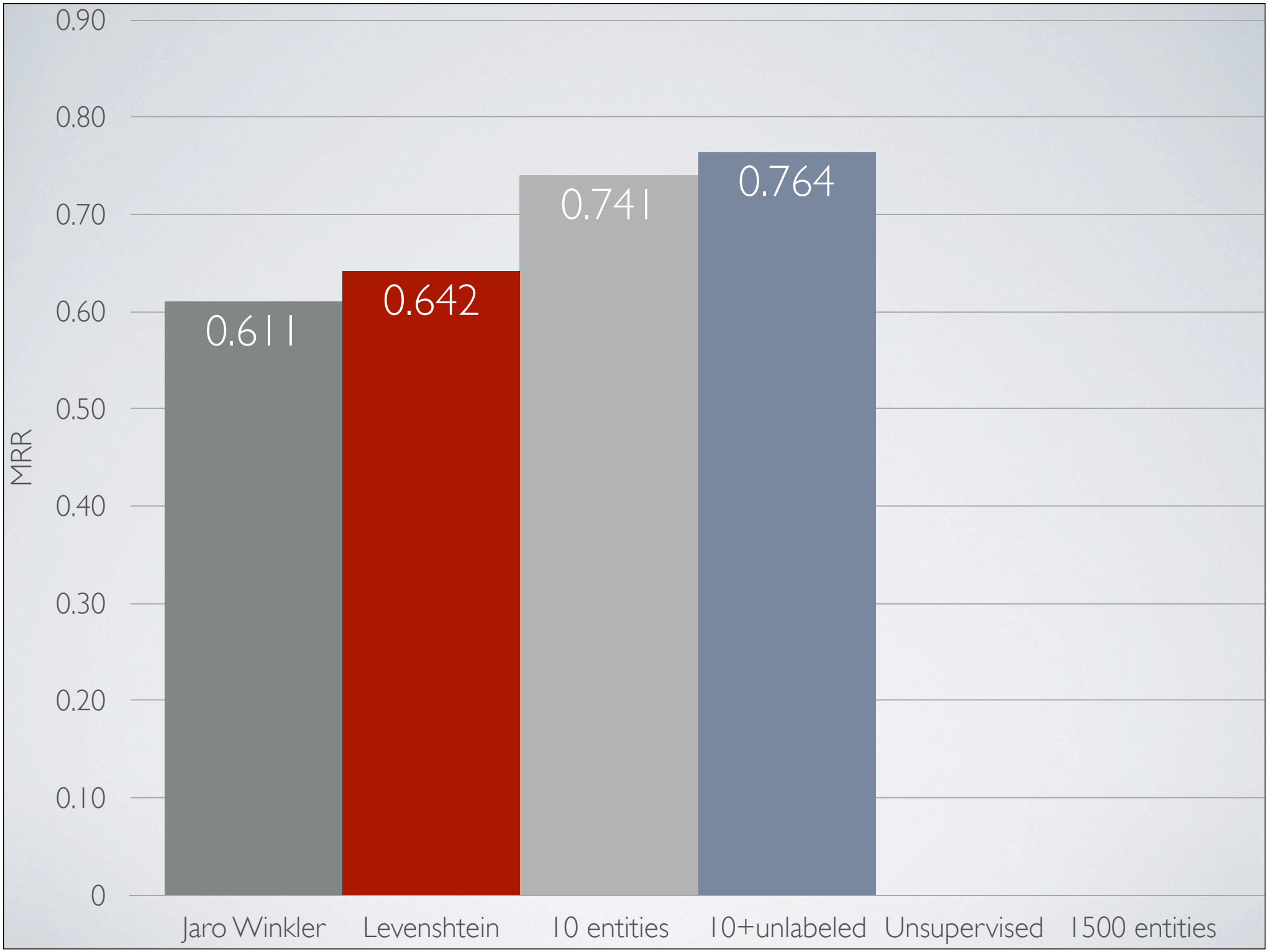
- Data
  - Train: 1500 entities (~6000 names)
  - Test: 1500 different entities (~6000 names)
- Settings
  - Train θ on a set of "supervised" pairs (varying levels of training)
- Baselines: other name matching algorithms

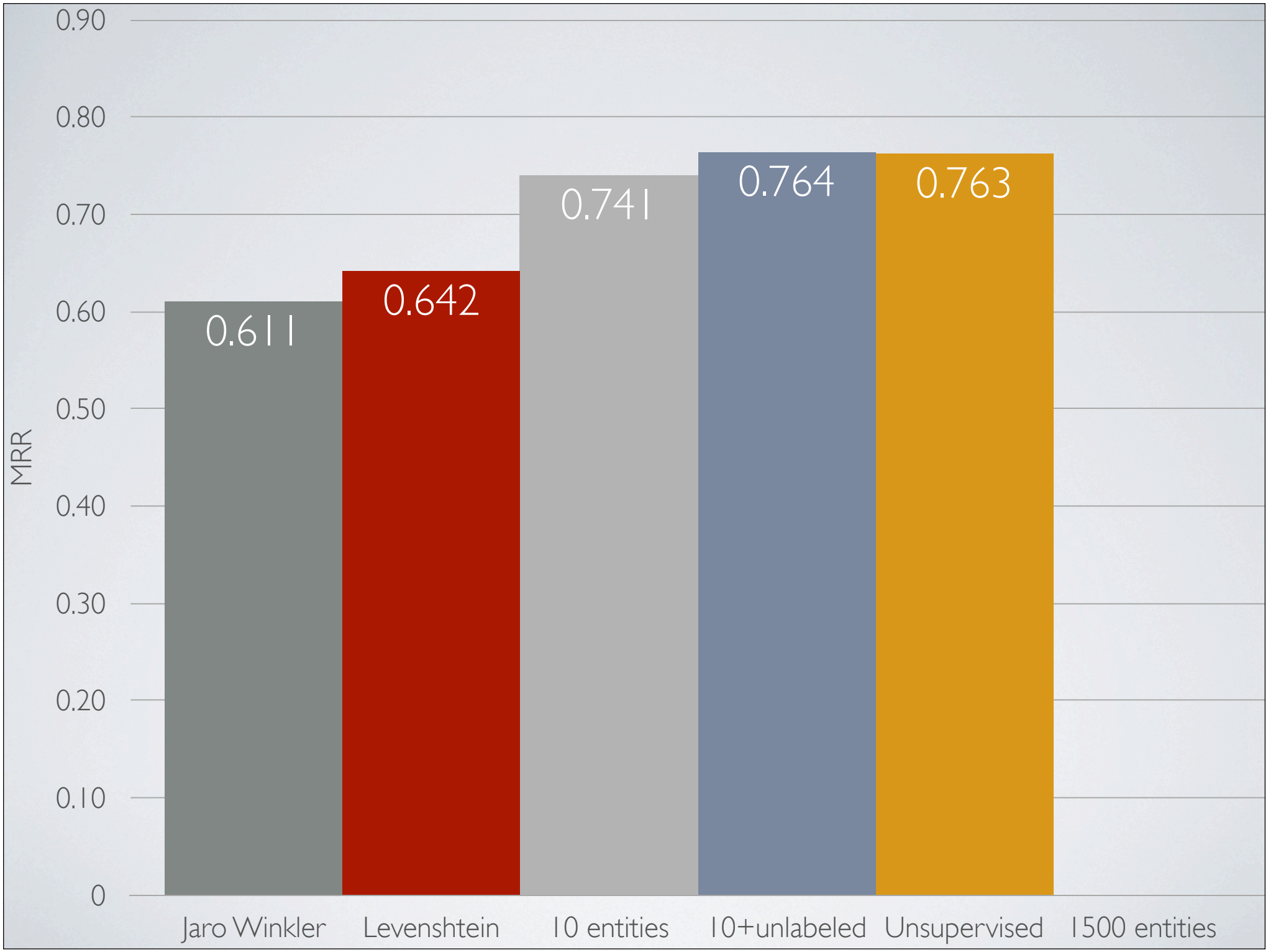

0.90
0.80
0.70
0.60
0.50
0.40
0.30
0.20
0.10
0
MRR
Jaro Winkler
Levenshtein
10 entities
10+unlabeled
Unsupervised
1500 entities

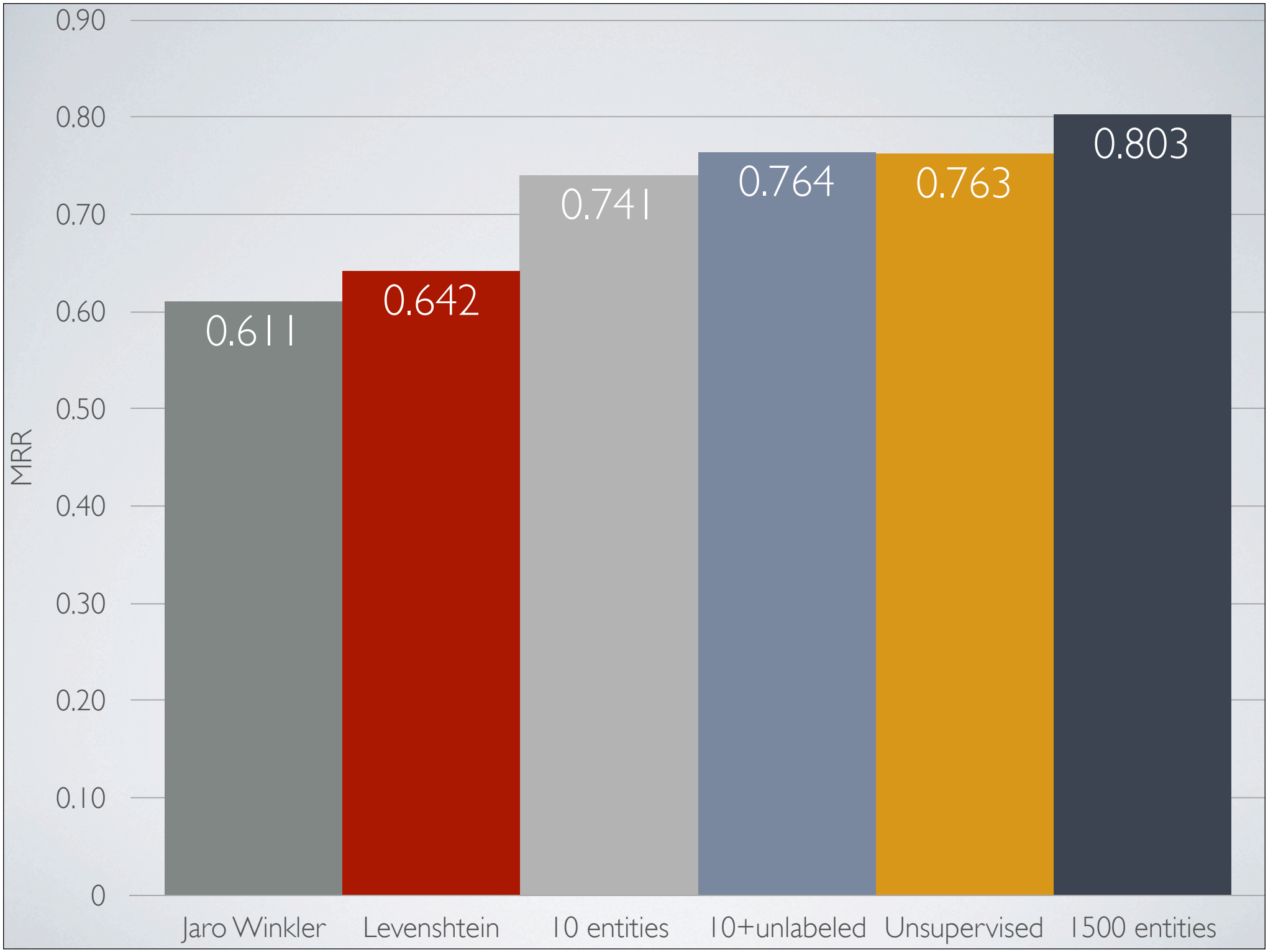

0.90
0.80
0.70
0.60
0.50
0.40
0.30
0.20
0.10
0
MRR
0.611
Jaro Winkler
Levenshtein
10 entities
10+unlabeled
Unsupervised
1500 entities


0.90
0.80
0.70
0.60
0.50
0.40
0.30
0.20
0.10
0
MRR
0.611
0.642
Jaro Winkler
Levenshtein
10 entities
10+unlabeled
Unsupervised
1500 entities


0.90
0.80
0.70
0.60
0.50
0.40
0.30
0.20
0.10
0
MRR
0.611
0.642
0.741
Jaro Winkler
Levenshtein
10 entities
10+unlabeled
Unsupervised
1500 entities


0.90
0.80
0.70
0.60
0.50
0.40
0.30
0.20
0.10
0
MRR
0.611
0.642
0.741
0.764
Jaro Winkler
Levenshtein
10 entities
10+unlabeled
Unsupervised
1500 entities


0.90
0.80
0.70
0.60
0.50
0.40
0.30
0.20
0.10
0
MRR
0.611
0.642
0.741
0.764
0.763
Jaro Winkler
Levenshtein
10 entities
10+unlabeled
Unsupervised
1500 entities


0.90
0.80
0.70
0.60
0.50
0.40
0.30
0.20
0.10
0
MRR
0.611
0.642
0.741
0.764
0.763
0.803
Jaro Winkler
Levenshtein
10 entities
10+unlabeled
Unsupervised
1500 entities

FUTURE WORK

- Include context for full entity disambiguation
- Increase matching speed
- More sophisticated mutation models
  - Incorporate internal name structure
- Informal genres
- Cross lingual data

QUESTIONS

Nicholas Andrews, Jason Eisner, Mark Dredze. **Name Phylogeny: A Generative Model of String Variation**. *Empirical Methods in Natural Language Processing (EMNLP)*, 2012.

Name Phylogeny: A Generative Model of String Variation

Nicholas Andrews and **Jason Eisner** and **Mark Dredze**
Department of Computer Science and Human Language Technology Center of Excellence
Johns Hopkins University
3400 N. Charles St., Baltimore, MD 21218 USA
{noa,eisner,mdredze}@jhu.edu

Abstract

Many linguistic and textual processes involve transduction of strings. We show how to learn a stochastic transducer from an unorganized collection of strings (rather than string pairs). The role of the transducer is to organize the collection. Our generative model explains similarities among the strings by supposing that some strings in the collection were not generated *ab initio*, but were instead derived by transduction from other, "similar" strings in the collection. Our variational EM learning algorithm alternately reestimates this phylogeny and the transducer parameters. The final learned transducer can quickly link any test name into the final phylogeny, thereby locating variants of the test name. We find that our method can effectively find name variants in a corpus of web strings used to refer to persons in Wikipedia, improving over standard untrained distances such as Jaro-Winkler and Levenshtein distance.

1 Introduction

Systematic relationships between pairs of strings are at the core of problems such as transliteration (Knight and Graehl, 1998), morphology (Dreyer and Eisner, 2011), cross-document coreference resolution (Bagga and Baldwin, 1998), canonicalization (Culotta et al., 2007), and paraphrasing (Barzilay and Lee, 2003). Stochastic transducers such as probabilistic finite-state transducers are often used to capture such relationships. They model a conditional distribution $p(y \mid x)$, and are ordinarily trained on input-output *pairs* of strings (Dreyer et al., 2008).

In this paper, we are interested in learning from an *unorganized* collection of strings, some of which might have been derived from others by *transformative linguistic processes* such as abbreviation, morphological derivation, historical sound or spelling change, loanword formation, translation, transliteration, editing, or transcription error. We assume that each string was derived from at most one parent, but may give rise to any number of children.

The difficulty is that most or all of these parent-child relationships are unobserved. We must reconstruct this evolutionary phylogeny. At the same time, we must fit the parameters of a model of the relevant linguistic process $p(y \mid x)$, which says what sort of children y might plausibly be derived from parent x. Learning this model of $p(y \mid x)$ helps us organize the training collection by reconstructing its phylogeny, and also permits us to generalize to new forms.

We will focus on the problem of name variation. We observe a collection of person names—full names, nicknames, abbreviated or misspelled names, etc. Some of these names can refer to the same person; we hope to detect this. It would be an unlikely coincidence if two mentions of John Jacob Jingleheimer Schmidt referred to different people, since this is a long and unusual name. Similarly, John Jacob Jingelhimer Smith and Dr. J. J. Jingleheimer may also be related names for this person. That is, these names may be derived from one another, via unseen relationships, although we cannot be sure.

Readers may be reminded of unsupervised clustering, in which "suspiciously similar" points can be explained as having been generated by the same cluster. Since each name is linked to at most one parent, our setting resembles single-link clustering—with a learned, asymmetric distance measure $p(y \mid x)$.

We will propose a generative process that makes explicit assumptions about how strings are copied with mutation. It is assumed to have generated all the names in the collection, in an unknown order. Given learned parameters, we can ask the model whether a name Dr. J. J. Jingelheimer in the collection is more likely to have been generated from scratch, or derived from some previous name.