Web-derived Pronunciations *for* Spoken Term Detection

Authors

Doğan Can	Boğaziçi University
Erica Cooper	MIT
Arnab Ghoshal	Johns Hopkins University
Martin Jansche	Google Inc.
Sanjeev Khudanpur	Johns Hopkins University
Bhuvana Ramabhadran	IBM T. J. Watson Research
Michael Riley	Google Inc.
Murat Saraçlar	Boğaziçi University
Abhinav Sethy	IBM T. J. Watson Research
Morgan Ulinski	Cornell University
Christopher White	Johns Hopkins University

BOĞAZİÇİ ÜNİVERSİTESİ
1863

CORNELL UNIVERSITY
FOUNDED A.D. 1865
Cornell University

Google™

IBM®

THE JOHNS HOPKINS UNIVERSITY · BALTIMORE · 1876
VERITAS VOS LIBERABIT

Authors

Doğan Can	Boğaziçi University
Erica Cooper	MIT
Arnab Ghoshal	Johns Hopkins University
Martin Jansche	Google Inc.
Sanjeev Khudanpur	Johns Hopkins University
Bhuvana Ramabhadran	IBM T. J. Watson Research
Michael Riley	Google Inc.
Murat Saraçlar	Boğaziçi University
Abhinav Sethy	IBM T. J. Watson Research
Morgan Ulinski	Cornell University
Christopher White	Johns Hopkins University

Overview

Spoken Term Detection (STD): open-vocabulary search over spoken document collections

Classic Large-Vocabulary Continuous Speech Recognition (LVCSR) assumes a closed vocabulary

Overview

Speech signal

Sampled waveform

Waveform windows

Cepstral features

Hidden Markov model states

Contextual phones

Phones

Words

Pronunciation model

Overview

Speech signal

Sampled waveform

Waveform windows

Cepstral features

Hidden Markov model states

Contextual phones

Phones

Overview

Spoken Term Detection (STD): **open-vocabulary** search over spoken document collections

Build phone index instead of word index

Search by (approximate) phonetic match

Need word pronunciations during search

Overview

Need word pronunciations during search

For an open-ended vocabulary

For proper names from a variety of origins

Continually evolving

Ahmadinejad, Blagojevich, Sotomayor, ...

Models

Models over pairs of strings:

- Letter-to-phone (L2P, pronunciation) models
- Phone-to-phone (P2P) model
- Letter-to-letter (L2L, transliteration) models

Models

Latent alignment models, like in SMT

$$\Pr[\lambda, \pi] = \sum_a [\lambda, \pi \mid a]$$

Alignments a assumed to be monotonic

Train on parallel data $(\lambda_1, \pi_1), \ldots, (\lambda_n, \pi_n)$:

- Impute latent alignments with a 1-gram model, EM trained from flat start
- Train n-gram language model on imputed alignments ($n = 2, 3, 4, 5$)

Models

Call these "pair *n*-gram models"

All models are joint models $\Pr[\lambda, \pi]$

For 1-gram models, can derive conditional models $\Pr[\lambda \mid \pi]$ or $\Pr[\pi \mid \lambda]$ from joint ones in closed form

Expressed as finite-state transducers (FSTs) using the OpenFst library (openfst.org)

Operations on models are well-known FST manipulations

The Web is a rich source of pronunciations:

IPA transcripton

The Ctenophora (pronounced /tɨˈnɒfərə/, singular **ctenophore**, pronounced /ˈtɛnəfɔər/ or /ˈtiːnəfɔər/), commonly known as comb jellies, are a phylum of animals that live in marine waters worldwide.

en.wikipedia.org

Ad-hoc transcription

Two species of **ctenophores** (pronounced TEN-uh-fores), can be found just off shore in the Chesapeake Bay: *Mnemiopsis* and *Beroe*.

nationalzoo.si.edu

The Moonjelly is a small sea creature about the size of a child's hand. It looks like a blob of clear, colorless jelly. Its scientific name is "**Ctenophore**" (pronounced tee-ne-for.)

markshasha.com

The Web is a rich source of pronunciations

Finding them involves:

- **Extracting** a superset of candidates
- **Validating** the extracted candidates
- **Normalizing** the pronunciations

Extraction

Find candidate pronunciations by pattern matching over billions of Web pages:

… (pronounced …)

… pronounced "… "

…, pronounced …,

… [… ə …]

… / … ə … /

… \ … ə … \

Extraction

IPA predates computers, the Web, and modern notions of phonetics/phonology

IPA is difficult to use even by experts

IPA symbols are scattered across several Unicode code blocks

Cannot tell just by looking at a character whether it is part of an IPA transcription

IPA characters are often misappropriated

sıɥʇ ǝʞıן uʍop ǝpısdn ǝʇıɹʍ uɐɔ noʎ

Extraction

For each pronunciation candidate, find the most likely matching orthographic string

- The Ctenophora (pronounced /tɨˈnɒfərə/, singular **ctenophore**, pronounced **/ˈtɛnəfɔər/**

Use a very simple pronunciation model to score orthographic strings

Validation

Extraction had to be simple and fast to allow it to run at Web scale

Extraction validation examines a few million (orthography, pronunciation) candidates and

- removes candidates with invalid or undesirable pronunciations
- removes candidates with wrong or undesirable orthographies

Validation

Rain Water, the product, comes from Dripping Springs, where it is collected and bottled by Richard Heinichen, a 57-year-old former blacksmith. … Mr. **Heinichen (pronounced like the beer)** said he sold about 170,000 16-ounce bottles last year…

nytimes.com

So, that said, I thought I'd talk a little about the towns of Dharamsala (pronounced Dar-am-Shala) and **Pushkar (pronounced like the thing you would do when your automobile breaks down)**.

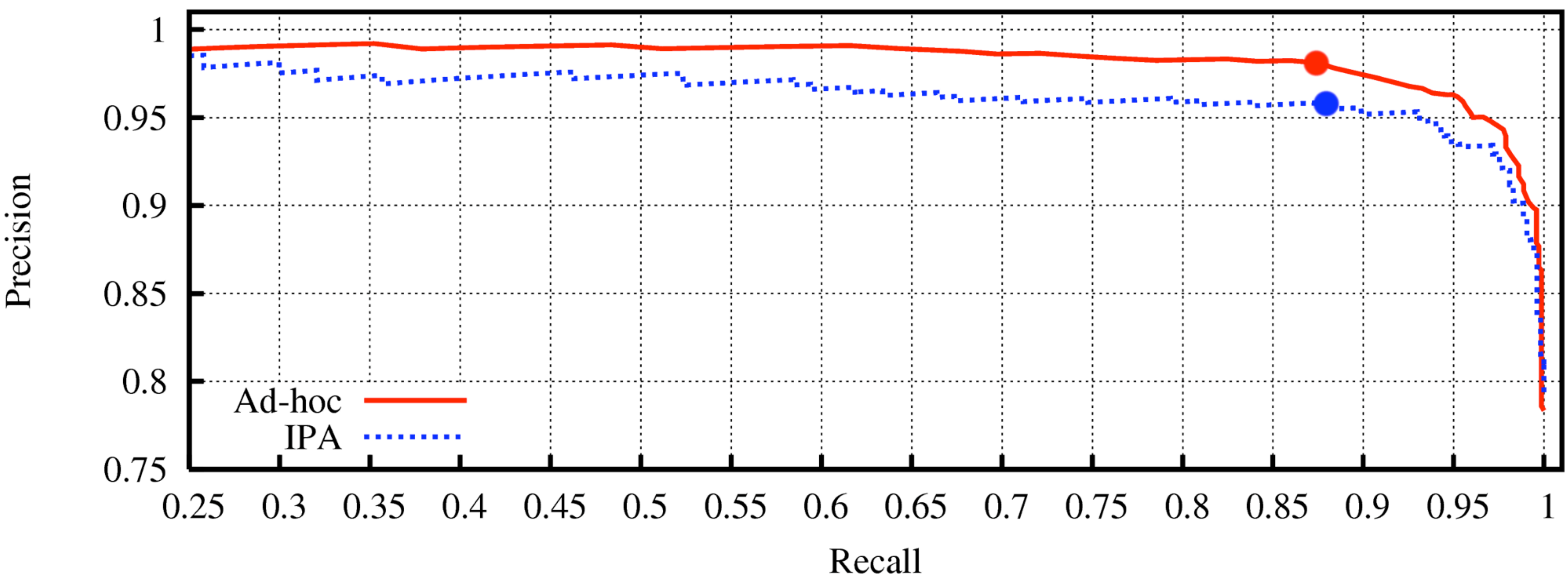
strangebenevolent.blogspot.com

Validation

Annotate a few hundred candidates

Extract a few dozen features, in particular alignment-based features that count e. g. vowel mismatches or consonant matches

Train and apply Support Vector Machine (SVM) classifiers



Precision
Recall
Ad-hoc
IPA
1
0.95
0.9
0.85
0.8
0.75
0.25
0.3
0.35
0.4
0.45
0.5
0.55
0.6
0.65
0.7
0.75
0.8
0.85
0.9
0.95
1

Normalization

Normalization is necessary to homogenize the extracted raw pronunciations

- For IPA pronunciations, transcription conventions and/or skills vary
- For ad-hoc pronunciations, need to generate phones

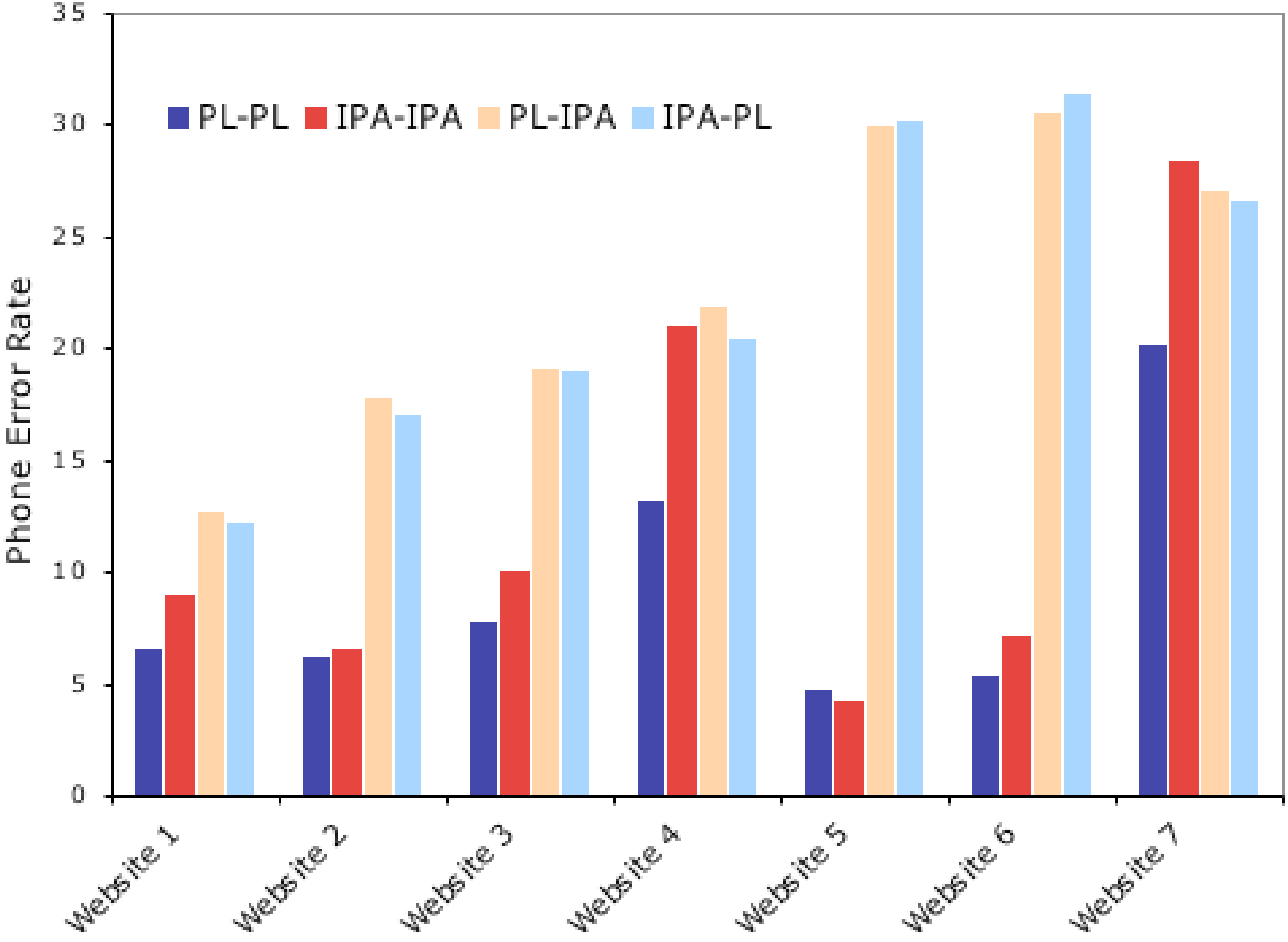
Normalization

For extracted IPA pronunciations, consider the subset of words found in Pronlex (PL)

Check what happens when we train L2P models on one source (PL, IPA) and evaluate it on another

Compute phone error rate (PhER) by 5-fold parallel cross-validation

Do this for the top 7 websites in our data



PL-PL
IPA-IPA
PL-IPA
IPA-PL
Phone Error Rate
0
5
10
15
20
25
30
35
Website 1
Website 2
Website 3
Website 4
Website 5
Website 6
Website 7

Normalization

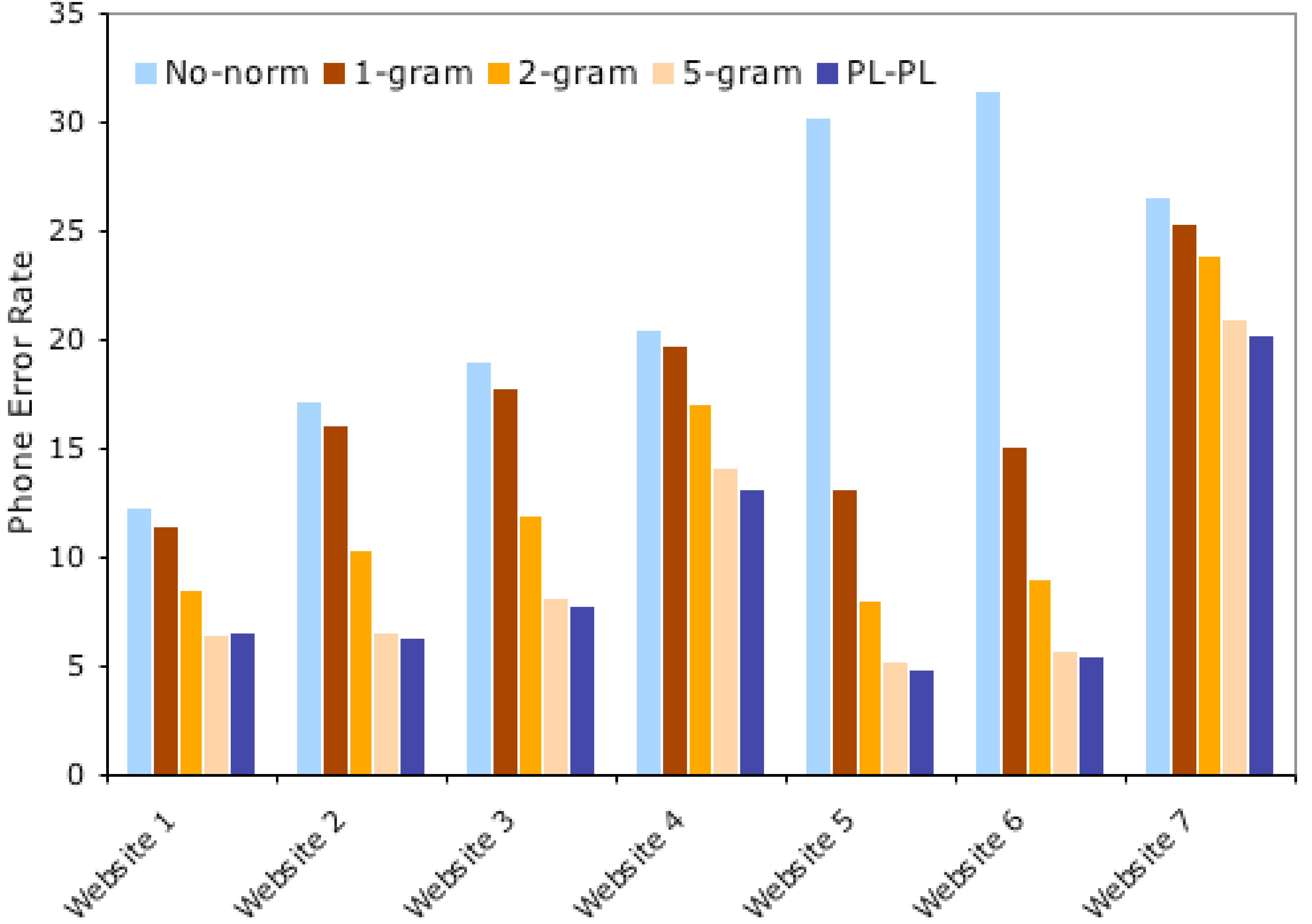
Focus on the IPA-PL evaluation

Train phone-to-phone (P2P) normalization models on parallel (IPA, Pronlex) data

Vary the *n*-gram order of the P2P models

Use P2P models to normalize IPA data, train L2P models on normalized IPA

Compare with L2P model trained directly on Pronlex



No-norm
1-gram
2-gram
5-gram
PL-PL
Phone Error Rate
0
5
10
15
20
25
30
35
Website 1
Website 2
Website 3
Website 4
Website 5
Website 6
Website 7

Normalization

Phonetic transcription conventions vary by data source

Website-specific IPA normalization makes extracted pronunciations look more like those found in Pronlex

L2P models trained on normalized Web-IPA pronunciations are as good as models trained on comparable amounts of Pronlex

Normalization

For extracted ad-hoc pronunciations, we need to derive phones from the two available orthographies

From last Wednesday's *New York Times*:

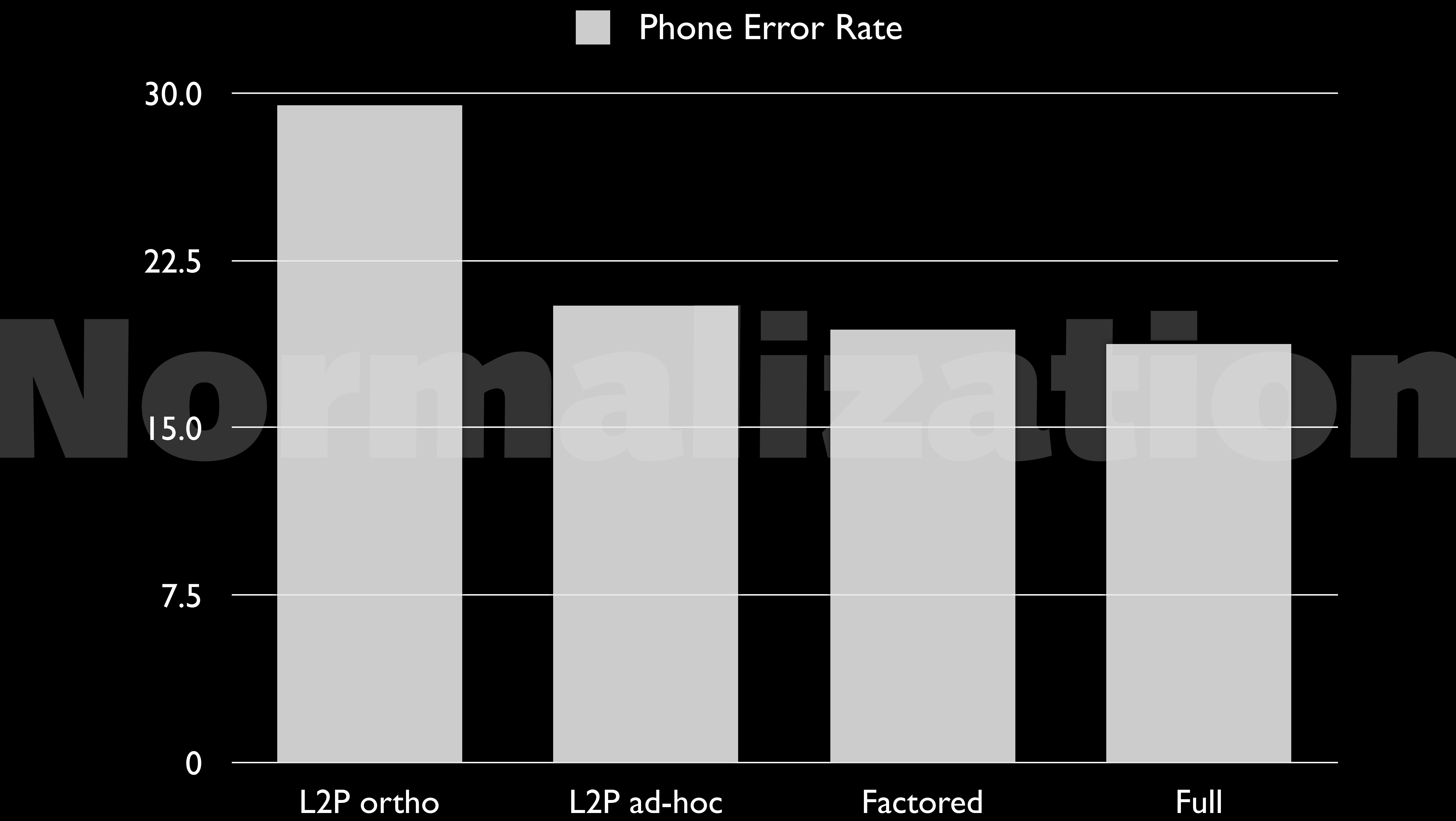
Phthalates (pronounced THAL-ates) are among the most common endocrine disruptors, and among the most difficult to avoid.

Ambiguities remain in the simplified orthography (which *th* sound?)

Normalization

Experiment with 4 ways of generating phones for ad-hoc pronunciations

- L2P model trained on orthography
- L2P model trained on ad-hoc prons
- Factored generative model with conditional independence
- Full model over aligned triples


Normalization
Phone Error Rate
30.0
22.5
15.0
7.5
0
L2P ortho
L2P ad-hoc
Factored
Full

Normalization

Ad-hoc transcriptions are easier to produce than IPA transcriptions

We found 80% more ad-hoc transcriptions than IPA on the Web

L2P models trained on ad-hoc data are better than L2P models trained on comparable amounts of data in standard orthography

Indexation

- Indexation of weighted finite automata
- Used in Spoken Utterance Retrieval and Spoken Term Detection
- Related to suffix and factor automata
- Implemented with OpenFst
- Also see *Spoken Information Retrieval for Turkish Broadcast News* by Parlak and Saraçlar in tonight's poster session

Indexation

Goal of Spoken Term Detection is to find the time interval containing the query, for each occurrence of the query

Retrieval is based on the posterior probability of substrings (factors) in a given time interval

Need to index the (preprocessed) output lattices of an automatic speech recognition (ASR) system

Indexation

Preprocessing of ASR output lattices:

- Cluster non-overlapping occurrrences of each word (or sub-word)
- Assign other occurrences to the cluster with which they maximally overlap
- Time interval of each cluster is the union of all its members
- Adaptively quantize the time intervals

Indexation

Index construction:

- Union of preprocssed FSTs
- Optimized for efficiency
- Factor-automaton introduces a new start state and a new final state, plus transitions to and from every other state
- Normalized to form a proper posterior probability distribution

Indexation

Searching for a user query is as simple as:

- Representing the query as an FSA, which may represent multiple pronunciations
- Composing the query FSA with the index FST
- Projecting onto the output labels (time intervals) and ranking by best path

Produces results ordered by decreasing posterior probability

Experiments

Analyze the impact of web-derived pronunciations on the retrieval of out-of-vocabulary (OOV) queries in an STD task

Held out 1290 names of persons and places and rare or foreign words with 5+ occurrences in the Broadcast News corpus

Removed those words from the vocabulary of the speech recognizer

Removed all utterances containing the held-out data from the BN training data

Experiments

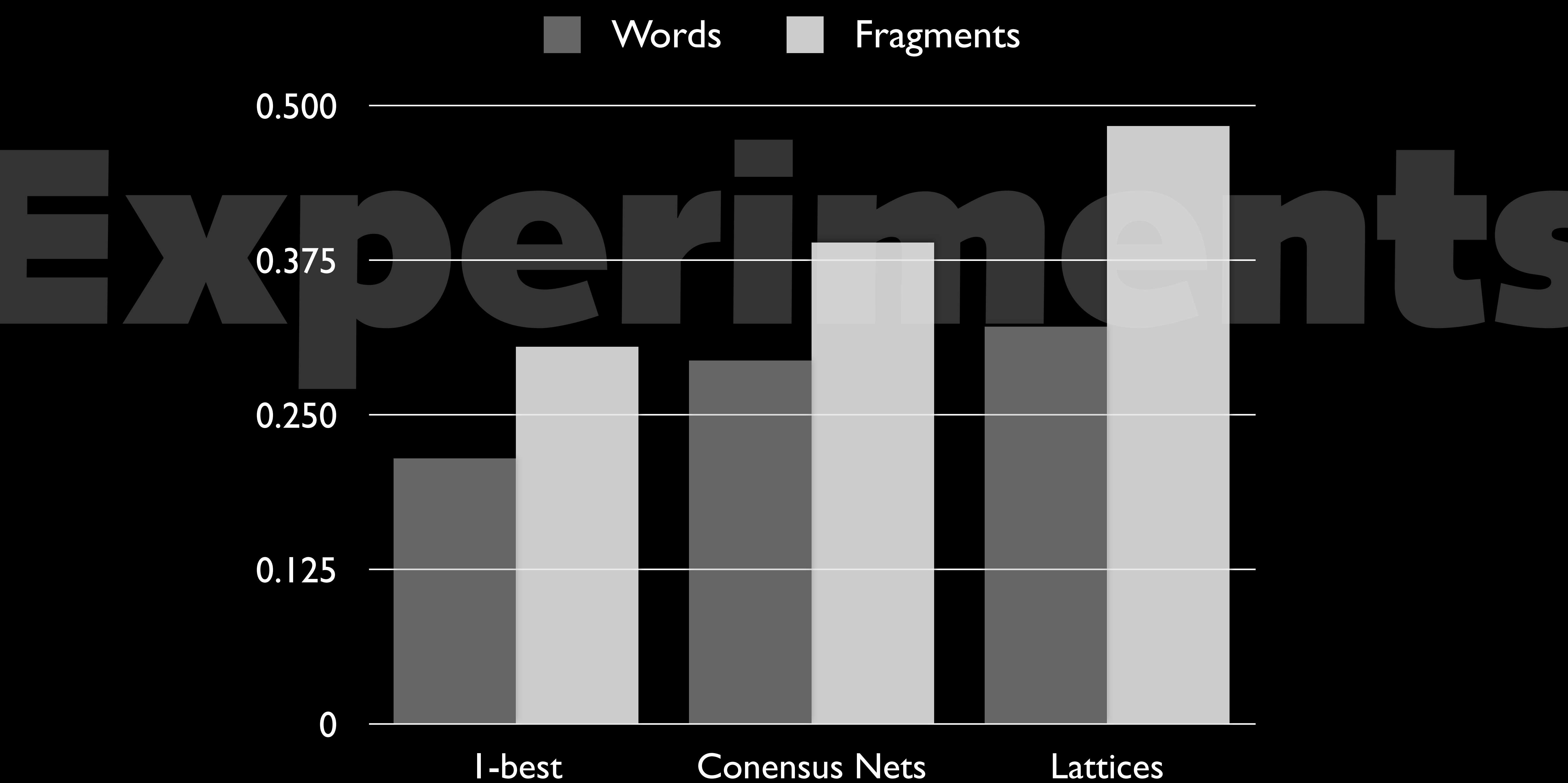
Trained a recognizer using the IBM Speech Recognition Toolkit on 300 hours of BN

Word error rate on standard BN test set was 19.4%

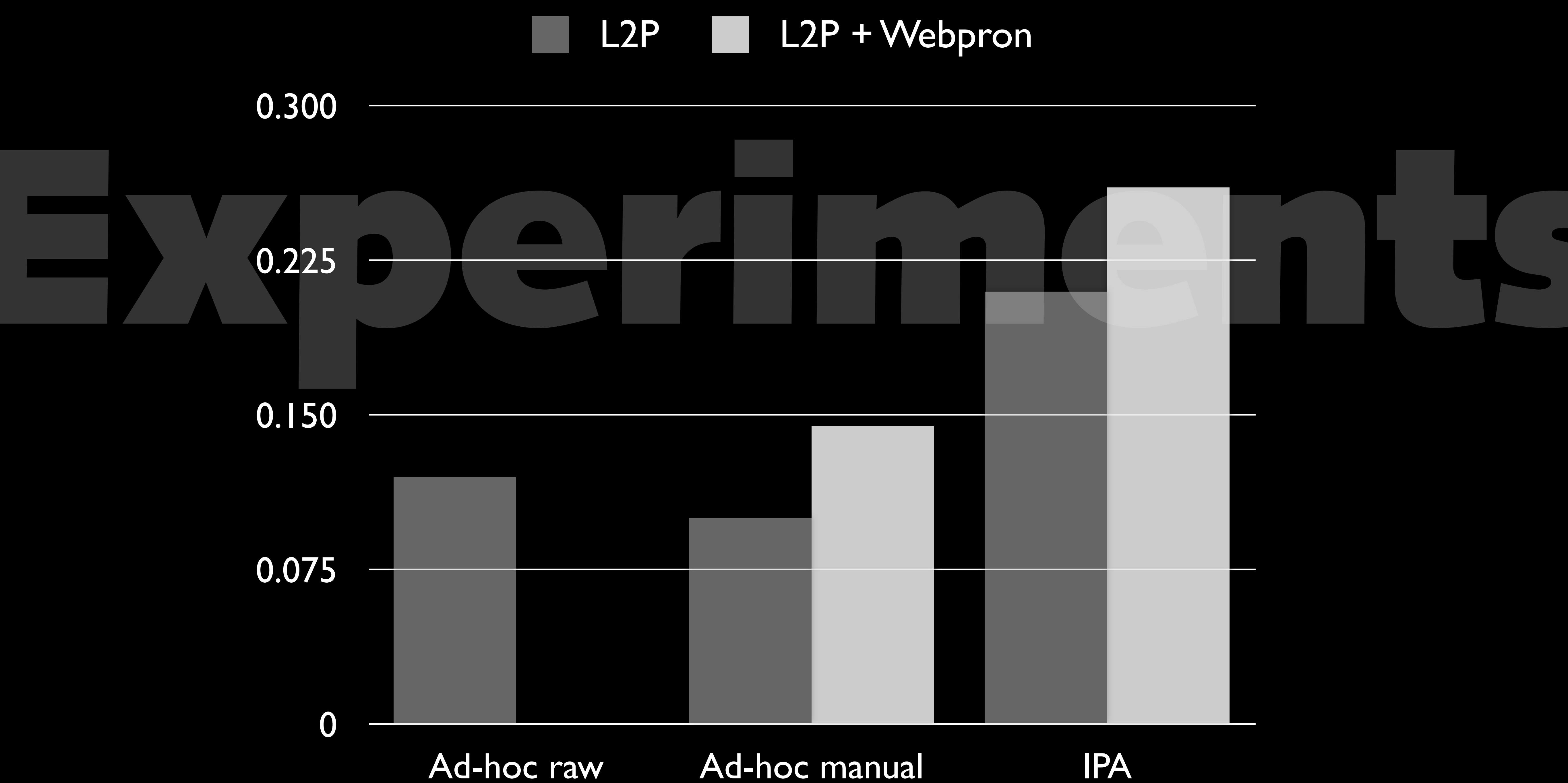
100 hours containing OOV terms held out for further experiments, transcribed by the recognizer and indexed by the STD system

Experiment with different pronunciations during retrieval, report ATWV metric from NIST 2006 STD Evaluation

Results with **reference pronunciations** in terms of ATWV (higher is better)



Experiments with Web-derived pronunciations added to a baseline L2P system

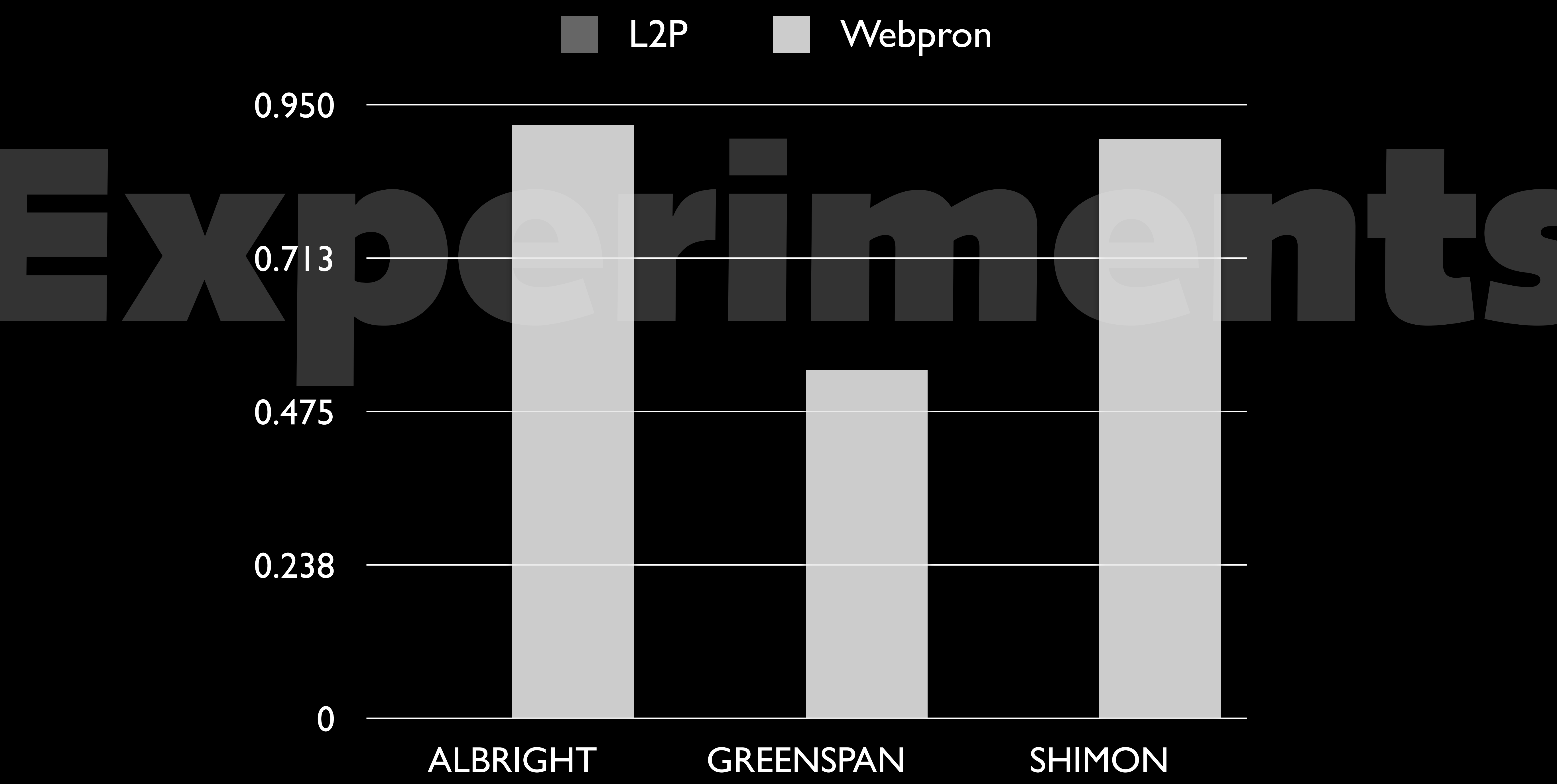


Examples of Webprons with positive impact

	L2P	Webpron
ALBRIGHT	ae l b r ay t	ao l b r ay t
GREENSPAN	g r iy n s p aa n	g r iy n s p ae n
SHIMON	sh ih m ax n	sh ih m ow n

Experiments

Fraction of correctly detected occurrences

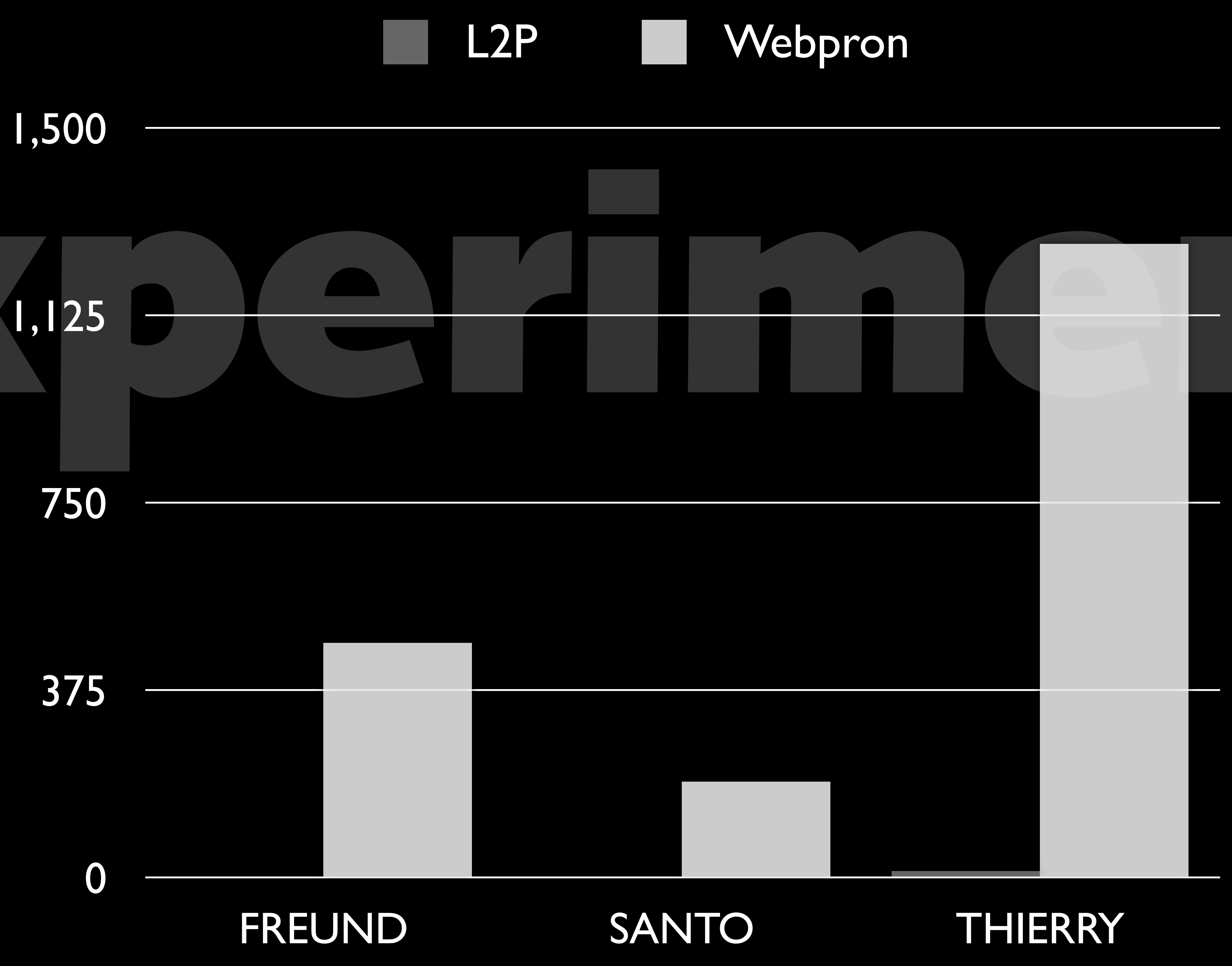


Experiments

Examples of Webprons with negative impact

	L2P	Webpron
FREUND	f r oy n d	f r eh n d
SANTO	s ae n t ow	s ax/ey/eh n t
THIERRY	th iy ax r iy	t eh r iy

Number of false alarms



Experiments

People sometimes use nearest-neighbor pronunciations, where the pronunciation of a familiar word is used for a similar unfamiliar word

For cases like Thierry / Terry, which occurs as a suffix in *military*, or *voluntary*, false alarms increase dramatically

Overall, Web-derived pronunciations have a net positive impact

Conclusion

Large quantities of human-supplied pronunciations are available on the Web

Our methods yield more than 7M occurrences of raw English pronunciations

After validation and normalization, extracted pronunciations have a positive impact on a Spoken Term Detection task

Our approach can be used to bootstrap pronunciation dictionaries for other tasks and languages