Sign constraints on feature weights improve a joint model of word segmentation and phonology

Mark Johnson
Macquarie University

Joint work with Joe Pater, Robert Staubs and Emmanuel Dupoux
Summary

• Background on word segmentation and phonology
  ▶ Liang et al and Berg-Kirkpatrick et al MaxEnt word segmentation models
  ▶ Smolensky’s Harmony theory and Optimality theory of phonology
  ▶ Goldwater et al MaxEnt phonology models

• A joint MaxEnt model of word segmentation and phonology
  ▶ because Berg-Kirkpatrick’s and Goldwater’s models are MaxEnt models, and MaxEnt models can have arbitrary features, it is easy to combine them
  ▶ Harmony theory and sign constraints on MaxEnt feature weights

• Experimental evaluation on Buckeye corpus
  ▶ better results than Börschinger et al 2014 on a harder task
  ▶ Harmony theory feature weight constraints improve model performance
Outline

Background

A joint model of word segmentation and phonology

Computational details

Experimental results

Conclusion
Word segmentation and phonological alternation

- Overall goal: model children’s acquisition of words
- Input: phoneme sequences with *sentence boundaries* (Brent)
- Task: identify *word boundaries* in the data, and hence *words* of the language

```
ju want tu si ðə bʊk
“you want to see the book”
```

- But a word’s pronunciation can vary, e.g., final /t/ in /want/ can delete
  - can we identify the *underlying forms* of words
  - can we learn how pronunciations alternate?
Prior work in word segmentation

- Brent et al 1996 proposed a Bayesian unigram segmentation model
- Goldwater et al 2006 proposed a Bayesian non-parametric bigram segmentation model that captures word-to-word dependencies
- Johnson et al 2008 proposed a hierarchical Bayesian non-parametric model that could learn and exploit phonotactic regularities (e.g., syllable structure constraints)
- Liang et al 2009 proposed a maximum likelihood unigram model with a word-length penalty term
- Berg-Kirkpatrick et al 2010 reformulated the Liang model as a MaxEnt model
The Berg-Kirkpatrick word segmentation model

- Input: sequence of utterances $D = (w_1, \ldots, w_n)$
  - each utterance $w_i = (s_{i,1}, \ldots, s_{i,m_i})$ is a sequence of (surface) phones

- The model is a unigram model, so probability of word sequence $w$ is:
  \[
P(w | \theta) = \sum_{s_1 \ldots s_\ell \text{ s.t. } s_1 \ldots s_\ell = w} \prod_{j=1}^{\ell} P(s_j | \theta)
  \]

- The probability of a word $P(s | \theta)$ is a MaxEnt model:
  \[
P(s | \theta) = \frac{1}{Z} \exp(\theta \cdot f(s)), \text{ where:}
  \]
  \[
  Z = \sum_{s' \in S} \exp(\theta \cdot f(s'))
  \]

- The set $S$ of possible surface forms is the set of all substrings in $D$ shorter than a length bound
Aside: the set $S$ of possible word forms

$$P(s \mid \theta) = \frac{1}{Z} \exp(\theta \cdot f(s)),$$ where:

$$Z = \sum_{s' \in S} \exp(\theta \cdot f(s'))$$

- Our estimators can be understood as adjusting the feature weights $\theta$ so the model doesn’t “waste” probability on forms $s$ that aren’t useful for analysing the data
- In the generative non-parametric Bayesian models, $S$ is the set of all possible strings
- In these MaxEnt models, $S$ is the set of substrings that actually occur in the data
- How does the difference in $S$ affect the estimate of $\theta$?
- Could we use the negative sampling techniques of Mnih et al 2012 to estimate MaxEnt models with infinite $S$?
The word length penalty term

• Easy to show that the MLE segmentation analyses each sentence as a single word
  ▶ the MLE minimises the KL-divergence between the data distribution and the model’s distribution

⇒ Liang and Berg-Kirkpatrick add a double-exponential word length penalty

\[
P(w \mid \theta) = \sum_{s_1 \ldots s_\ell \text{s.t.} s_1 \ldots s_\ell = w} \prod_{j=1}^{\ell} P(s_j \mid \theta) \exp(-|s_i|^d)
\]

⇒ \( P(w \mid \theta) \) is deficient (i.e., \( \sum_w P(w \mid \theta) < 1 \))
  ▶ because we use a word length penalty in the same way, our models are deficient also

• The loss function they optimise is an \( L_2 \) regularised version of:

\[
L_D(\theta) = \prod_{i=1}^{n} P(w_i \mid \theta)
\]
Sensitivity to word length penalty factor $d$

Data

| Word length penalty | Surface token f-score |
|---------------------|-----------------------|
| 0.3                 |                       |
| 0.4                 |                       |
| 0.5                 |                       |
| 0.6                 |                       |
| 0.7                 |                       |
| 0.8                 |                       |
| 0.9                 |                       |
| 1.4 1.5 1.6 1.7     |                       |

Data

- Brent
- Buckeye
Phonological alternation

- Words are often pronounced in different ways depending on the context.
- Segments may *change* or *delete*.
  - Here we model *word-final /d/ and /t/ deletion*.
  - E.g., \( /\text{w a n t t u}/ \Rightarrow [\text{w a n t u}] \)
- These alternations can be modelled by:
  - Assuming that each word has an *underlying form* which may differ from the observed *surface form*.
  - There is a set of *phonological processes* mapping underlying forms into surface forms.
  - These phonological processes can be *conditioned* on the context.
    - E.g., /t/ and /d/ deletion is more common when the following segment is a consonant.
  - These processes can also be *nondeterministic*.
    - E.g., /t/ and /d/ deletion doesn’t always occur even when the following segment is a consonant.
Harmony theory and Optimality theory

- Harmony theory and Optimality theory are two models of linguistic phenomena (Smolensky 2005)
- There are two kinds of constraints:
  - **faithfulness constraints**, e.g., underlying /t/ should appear on surface
  - universal **markedness constraints**, e.g., \(*tC\)
- Languages differ in the importance they assign to these constraints:
  - in Harmony theory, violated constraints incur **real-valued costs**
  - in Optimality theory, constraints are **ranked**
- The grammatical analyses are those which are **optimal**
  - often not possible to simultaneously satisfy all constraints
  - in Harmony theory, the optimal analysis minimises the sum of the costs of the violated constraints
  - in Optimality theory, the optimal analysis violates the lowest-ranked constraint
    - Optimality theory can be viewed as a discrete approximation to Harmony theory
Harmony theory as Maximum Entropy models

- Harmony theory models can be viewed as Maximum Entropy a.k.a. log-linear a.k.a. exponential models

| Harmony theory                  | MaxEnt models                   |
|---------------------------------|---------------------------------|
| underlying form $u$ and surface form $s$ | event $x = (s, u)$            |
| Harmony constraints             | MaxEnt features $f(s, u)$       |
| constraint costs                | MaxEnt feature weights $\theta$|
| Harmony                         | $-\theta \cdot f(s, u)$        |

$$P(u, s) = \frac{1}{Z} \exp -\theta \cdot f(s, u)$$
Learning Harmonic grammar weights

- Goldwater et al 2003 learnt Harmonic grammar weights from (underlying,surface) word form pairs (i.e., supervised learning)
  - now widely used in phonology, e.g., Hayes and Wilson 2008
- Eisenstadt 2009 and Pater et al 2012 infer the underlying forms and learn Harmonic grammar weights from surface paradigms alone
- Linguistically, it makes sense to require the weights $-\theta$ to be negative since Harmony violations can only make a $(s, u)$ pair less likely (Pater et al 2009)
Integrating word segmentation and phonology

- Prior work has used *generative models*
  - generate a sequence of underlying words from Goldwater’s bigram model
  - map the underlying phoneme sequence into a sequence of surface phones
- Elsner et al 2012 learn a finite state transducer mapping underlying phonemes to surface phones
  - for computational reasons they only consider simple substitutions
- Börschinger et al 2013 only allows word-final /t/ to be deleted
- Because these are all generative models, they can’t handle arbitrary feature dependencies (which a MaxEnt model can, and which are needed for Harmonic grammar)
Outline

Background

A joint model of word segmentation and phonology

Computational details

Experimental results

Conclusion
Possible (underlying, surface) pairs

- Because Berg-Kirkpatrick’s word segmentation model is a MaxEnt model, it is easier to integrate it with Harmonic Grammar/MaxEnt models of phonology.
- $P(x)$ is a distribution over surface form/underlying form pairs $x = (s, u)$ where:
  - $s \in S$, where $S$ is the set of length-bounded substrings of $D$, and
  - $s = u$ or $s \in p(u)$, where $p \in P$ is a phonological alternation
    - our model has two alternations, word-final /t/ deletion and word-final /d/ deletion
  - we also require that $u \in S$ (i.e., every underlying form must appear somewhere in $D$)
- Example: In Buckeye data, the candidate $(s, u)$ pairs include $([l.ih.v], /l.ih.v/)$, $([l.ih.v], /l.ih.v.d/)$, and $([l.ih.v], /l.ih.v.t/)$$
  - these correspond to “live”, “lived” and the non-word “livet”
Probabilistic model and optimisation objective

- The probability of word-final /t/ and /d/ deletion depends on the following word contexts $C = \{C, V, \#\}$:

$$P(s, u | c, \theta) = \frac{1}{Z_c} \exp(\theta \cdot f(s, u, c)),$$

where:

$$Z_c = \sum_{(s, u) \in \mathcal{X}} \exp(\theta \cdot f(s, u, c)) \text{ for } c \in C$$

- We optimise an $L_1$ regularised log likelihood $Q_D(\theta)$, with the word length penalty applied to the underlying form $u$:

$$Q(s | c, \theta) = \sum_{u: (s, u) \in \mathcal{X}} P(s, u | c, \theta) \exp(-|u|^d)$$

$$Q(w | \theta) = \sum_{s_1 \ldots s_\ell} \prod_{j=1}^{\ell} Q(s_j | c, \theta)$$

$$Q_D(\theta) = \sum_{i=1}^{n} \log Q(w_i | \theta) - \lambda \|\theta\|_1$$
MaxEnt features

- Here are the features \( f(s, u, c) \) where \( s = [l.ih.v] \), \( u = /l.ih.v.t/ \) and \( c = C \)
  - **Underlying form lexical features**: A feature for each underlying form \( u \). In our example, the feature is \( <U l i h v t> \). These features enable the model to learn language-specific lexical entries. There are 4,803,734 underlying form lexical features (one for each possible substring in the training data).
  - **Surface markedness features**: The length of the surface string (\( <#L 3> \)), the number of vowels (\( <#V 1> \)), the surface prefix and suffix CV shape (\( <CVPrefix CV> \) and \( <CVSuffix VC> \)), and suffix+context CV shape (\( <CVContext _C> \) and \( <CVContext C _C> \)). There are 108 surface markedness features.
  - **Faithfulness features**: A feature for each divergence between underlying and surface forms (in this case, \( <*F t> \)). There are two faithfulness features.
$L_1$ regularisation and sign constraints

- We chose to use $L_1$ regularisation because it promotes *weight sparsity* (i.e., solutions where most weights are zero)
  - rather than assigning every possible lexical entry and constraint a non-zero weight (as $L_2$ would), we may identify the subset of lexical entries and constraints relevant to the language
  - in turns out that $L_1$ and $L_2$ regularisation produce similair results

- The $L_1$ regularised log-likelihood is discontinuous at zero
  - gradient-based methods like LBFGS can’t handle this discontinuity
  - the OWLQN extension of LBFGS stops the line minimisation whenever it crosses an *orthant boundary* (Andrew et al 2007)
  - easy to extend this to impose sign constraints on weights

- Sign constraints we explored:
  - Lexical entry weights must be positive (i.e., you learn what words are in the language)
  - Harmony faithfulness and markedness constraint weights must be negative
Outline

Background

A joint model of word segmentation and phonology

Computational details

Experimental results

Conclusion
Determining the possible surface and underlying forms

- The set of possible surface forms $S$ is the set of all substrings in the training data of length $\leq 15$
- $\mathcal{X}$ contains \textit{possible (surface, underlying) word pairs}. For each $s \in S$, $(s, s) \in \mathcal{X}$, and $(s, s + /d/) \in \mathcal{X}$ if $s + /d/ \in S$ (same for /t/)

$$P(s, u \mid c, \theta) = \frac{1}{Z_c} \exp(\theta \cdot f(s, u, c)),$$

where:

$$Z_c = \sum_{(s, u) \in \mathcal{X}} \exp(\theta \cdot f(s, u, c)) \text{ for } c \in \mathcal{C}$$

$$Q(s \mid c, \theta) = \sum_{u : (s, u) \in \mathcal{X}} P(s, u \mid c, \theta) \exp(-|u|^d)$$

$$\frac{\partial \log Q(s \mid c, \theta)}{\partial \theta} = \mathbb{E}[f(s, u, c) \exp(-|u|^d) \mid s, c, \theta] - \mathbb{E}[f(s, u, c) \mid c, \theta]$$

- The first expectation sums over underlying forms $u : (s, u) \in \mathcal{X}$, while the second expectation sums over all $(s, u) \in \mathcal{X}$
Dynamic programming for log $Q(w \mid \theta)$

$$
Q(w \mid \theta) = \sum_{s_1 \ldots s_\ell} \prod_{j=1}^{\ell} Q(s_j \mid c, \theta)
\quad \text{s.t. } s_1 \ldots s_\ell = w
$$

$$
Q_D(\theta) = \sum_{i=1}^{n} \log Q(w_i \mid \theta) - \lambda \|\theta\|_1
$$

- We can sum/maximise over all $s_1 \ldots s_\ell$ such that $s_1 \ldots s_\ell = w$ by using *dynamic programming*

- A *forward-backward type calculation* calculates the expectations required to compute $\partial \log Q(w) / \partial \theta$
Outline

Background

A joint model of word segmentation and phonology

Computational details

Experimental results

Conclusion
Data preparation procedure

- Data from *Buckeye corpus* of conversational speech (Pitt et al 2007)
  - provides an underlying and surface form for each word
- Data preparation as in Börschinger et al 2013
  - we use the Buckeye underlying form as our underlying form
  - we use the Buckeye underlying form as our surface form as well . . .
  - except that if the Buckeye underlying form ends in a /d/ or /t/ and the surface form does not end in that segment our surface form is the Buckeye underlying form with that segment deleted
- Example: if Buckeye $u = /l.ih.v.d/$ “lived”, $s = [l.ah.v]$ then our $u = /l.ih.v.d/$, $s = [l.ih.v]$
- Example: if Buckeye $u = /l.ih.v.d/$ “lived”, $s = [l.ah.v.d]$ then our $u = /l.ih.v.d/$, $s = [l.ih.v.d]$
Data statistics

- The data contains 48,796 sentences and 890,597 segments.
- The longest sentence has 187 segments.
- The “gold” segmentation has 236,996 word boundaries, 285,792 word tokens, and 9,353 underlying word types.
- The longest word has 17 segments.
- Of the 41,186 /d/s and 73,392 /t/s in the underlying forms, 24,524 /d/s and 40,720 /t/s are word final, and of these 13,457 /d/s and 11,727 /t/s are deleted.
- All possible substrings of length 15 or less are possible surface forms $S$
- There are 4,803,734 possible word types and 5,292,040 possible surface/underlying word type pairs.
- Taking the 3 contexts derived from the following word into account, there are 4,969,718 possible word+context types.
- When all possible surface/underlying pairs are considered in all possible contexts there are 15,876,120 possible surface/underlying/context triples.
Overall segmentation scores

|                           | Börschinger et al. 2013 | Our model |
|---------------------------|-------------------------|-----------|
| Surface token f-score     | 0.72                    | **0.76** (0.01) |
| Underlying type f-score   | —                       | 0.37 (0.02) |
| Deleted /t/ f-score       | 0.56                    | **0.58** (0.03) |
| Deleted /d/ f-score       | —                       | 0.62 (0.19) |

- Results summary for our model compared to Börschinger et al (2013)
  - their model only recovers word-final /t/ deletions and was run on data without word-final /d/ deletions, so it is solving a simpler problem
- Surface token f-score is the standard token f-score
- Underlying type or “lexicon” f-score measures the accuracy with which the underlying word types are recovered.
- Deleted /t/ and /d/ f-scores measure the accuracy with which the model recovers segments that don’t appear in the surface.
- These results are averaged over 40 runs (standard deviations in parentheses) with the word length penalty \(d = 1.525\) applied to underlying forms
The effect of feature weight constraints

- The effect of constraints on feature weights on surface token f-score.
- “OT” indicates that the markedness and faithfulness features are required to be non-positive.
- “Lexical” indicates that the underlying lexical features are required to be non-negative.
The effect of feature weight constraints on the number of deleted underlying /d/ and /t/ segments posited by the model (d = 1.525).

The red diamond indicates the 13,457 deleted underlying /d/ and 11,727 deleted underlying /t/ in the “gold” data.
The regularised log-likelihood as a function of the number of non-zero weights for different constraints on feature weights ($d = 1.525$).
The number of words posited by the model

- The number of underlying types proposed by the model as a function of the number of non-zero weights, for different constraints on feature weights ($d = 1.525$).
- There are 9,353 underlying types in the “gold” data.
F-score for deleted /d/ and /t/ recovery as a function of word length penalty $d$ and whether all surface/underlying pairs $\mathcal{X}$ are included in all contexts $\mathcal{C}$

OT + Lexical weight constraints
Outline

Background

A joint model of word segmentation and phonology

Computational details

Experimental results

Conclusion
Conclusion and future work

- Word segmentation and phonology can be integrated in a MaxEnt framework to produce state-of-the-art results
  - sensitivity to the *word length penalty* is a major drawback
  - can this be set in a principled way?

- Constraining the feature weights associated with Markedness and Faithfulness constraints improves the procedure’s performance considerably

- Can we generalise the approach to cover a wider range of phonological processes?

- Can we generalise the approach to cover morpho-phonological processes, where a single form has several hierarchical structures?