# Non-parametric Bayesian Segmentation of Japanese Noun Phrases

Yugo Murawaki  Sadao Kurohashi
Graduate School of Informatics
Kyoto University

EMNLP2011, Edinburgh

# Outline

- Motivation
- Models and Inference
- Experiments

# Outline

- Motivation
- Models and Inference
- Experiments

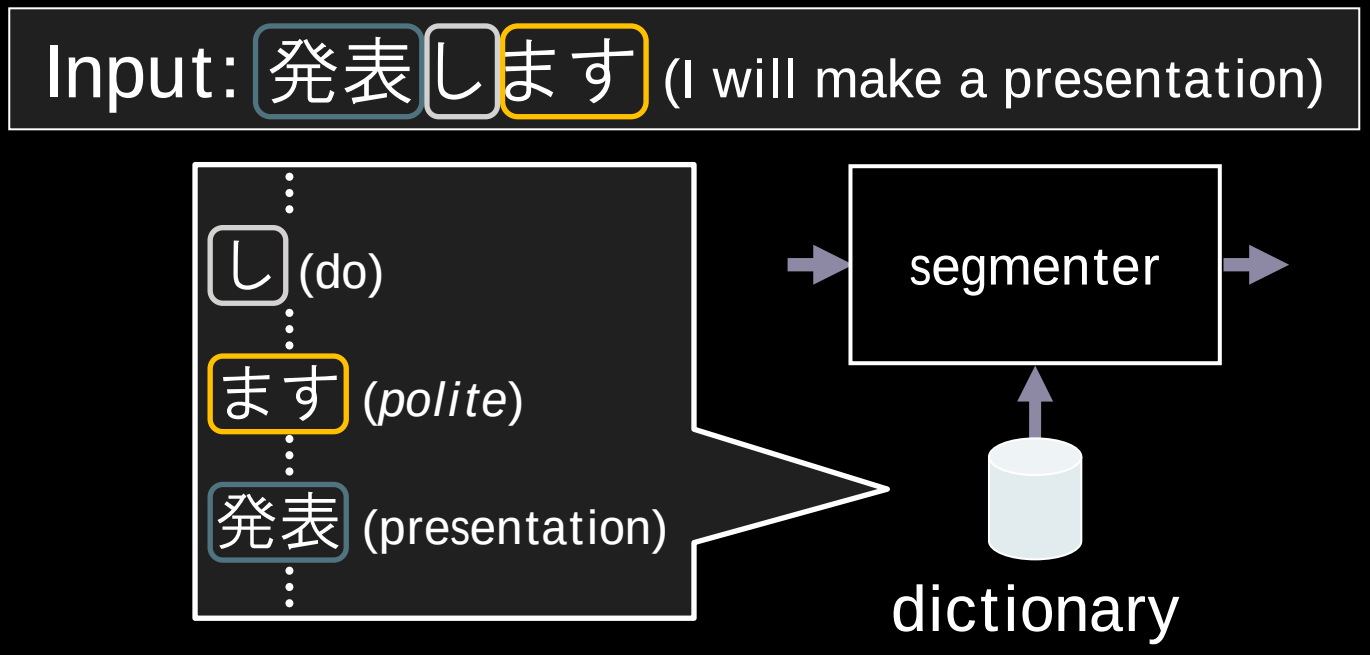
# Importance of a Dictionary in Japanese Word Segmentation

- No space between words

Input: 発表します (I will make a presentation)

# Importance of a Dictionary in Japanese Word Segmentation

- No space between words
- Words in the input are assumed to be covered by a pre-defined dictionary
  - To tell real words from non-words



# Unknown Word Problem

- Cannot register all words by hand

# Unknown Word Problem

- Cannot register all words by hand
- Solution: run a separate lexical acquisition process to find unknown words in text
  - Use broader context for disambiguation
  - Exploit rich morphology of Japanese (Murawaki+, 2008)

# Morphology does not help segment a noun phrase

フェルミ -φ エネルギー

Fermi energy

常山 -φ 城

Tsuneyama castle

(a place)

No *glue* is attached to the constituents of a noun phrase

# Morphology does not help segment a noun phrase

Every position within a noun phrase is potentially a boundary

常山 -φ 城

Tsuneyama castle

(a place)

# Morphology does not help segment a noun phrase

Every position within a noun phrase is potentially a boundary

| 常山 | -φ | 城 | ? | 常 | -φ | 山城 |
|---|---|---|---|---|---|---|
| Tsuneyama (a place) | | castle | | always | | Yamashiro (a place) |

# Statistical Approach

- Assumption: if the noun phrase in question consists of more than one word, its constituents should appear frequently in text, in isolation and as part of other noun phrases
- Simple concatenative model
  - Cannot handle complex morphology
  - But OK with noun phrases

# Statistical Approach

常山 - 城 ? 常 - 山城

Tsuneyama castle always Yamashiro

(a place) (a place)

Noun phrase: 常山城

Related text:

常山城は日本の城。
・・・にまたがる常山にあった
常山山頂からは児島湾・・・
最寄駅　　ＪＲ宇野線常山駅

# Statistical Approach

常山 - 城 ? 常 - 山城

Tsuneyama castle always Yamashiro

(a place) (a place)

Noun phrase: 常山城

Related text:

常山城は日本の城。
・・・にまたがる常山にあった
常山山頂からは児島湾・・・
最寄駅　ＪＲ宇野線常山駅

# Statistical Approach

常山 - 城 ? 常 - 山城

Tsuneyama (a place) castle always Yamashiro (a place)

Noun phrase: 常山城

Related text:

常山城は日本の城。
・・・にまたがる常山にあった
常山山頂からは児島湾・・・
最寄駅　ＪＲ宇野線常山駅

# Statistical Approach

常山 - 城 ? 常 - 山城

Tsuneyama castle always Yamashiro

(a place) (a place)

Noun phrase

常山城

Related text

常山城は日本の城。

・・・にまたがる常山にあった

常山山頂からは児島湾・・・

最寄駅 ＪＲ宇野線常山駅

# Outline

- Motivation
- Models and Inference
- Experiments

# Statistical Language Models

1. Unigram model
   - P(*A*-*B*) = P(*A*) P(*B*)
   - With a Dirichlet process prior, the model can treat any substring as a word candidate
   - Tends to misidentify common collocations as single words
     - The noun phrase in question is the case!

(Goldwater+, 2009)

# Statistical Language Models

2. Bigram model
   - P(*A*-*B*) = P(*A*|#) P(*B*|*A*) P(#|*B*)
   - Hierarchical Dirichlet process prior with a unigram base measure
   - Better handling of collocations

(Goldwater+, 2009)

# Inference

- Gibbs sampling as randomized search
  - Initialize segmentation
  - Repeat stochastic alternation of local segmentation
- Type-based block sampling (Liang+, 2010)
  - Much faster convergence than token-based sampling
  - Allows non-randomized, consistent initialization

# Token-based Sampling (Unigram)

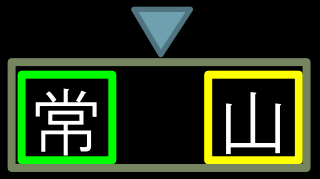
# – 常 – 山 – 山　頂 – か　ら – は –

# – 常 – 山 – に – あ　っ　た –

# – Ｊ　Ｒ – 宇　部 – 線 – 常 – 山 – 駅

# – 常 – 山　城 – は –

# Token-based Sampling (Unigram)



# – 常 山 – 山 頂 – か ら – は –

# – 常 – 山 – に – あ っ た –

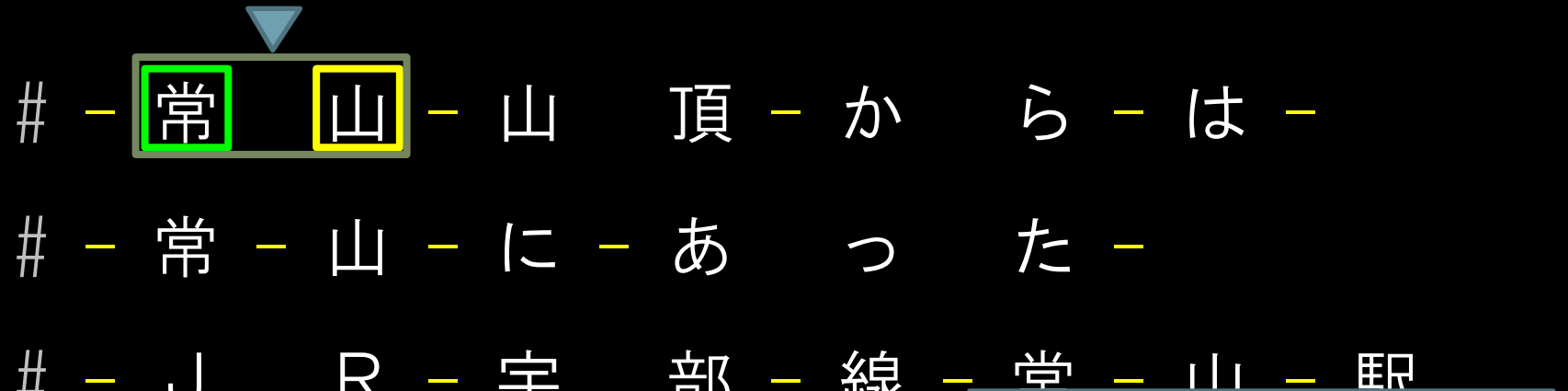
# – J R – 宇 部 – 線 – 常 – 山 – 駅

# – 常 – 山 城 – は –

Boundary: P(常|z-) P(山|z-,常)

Non-boundary: P(常山|z-)

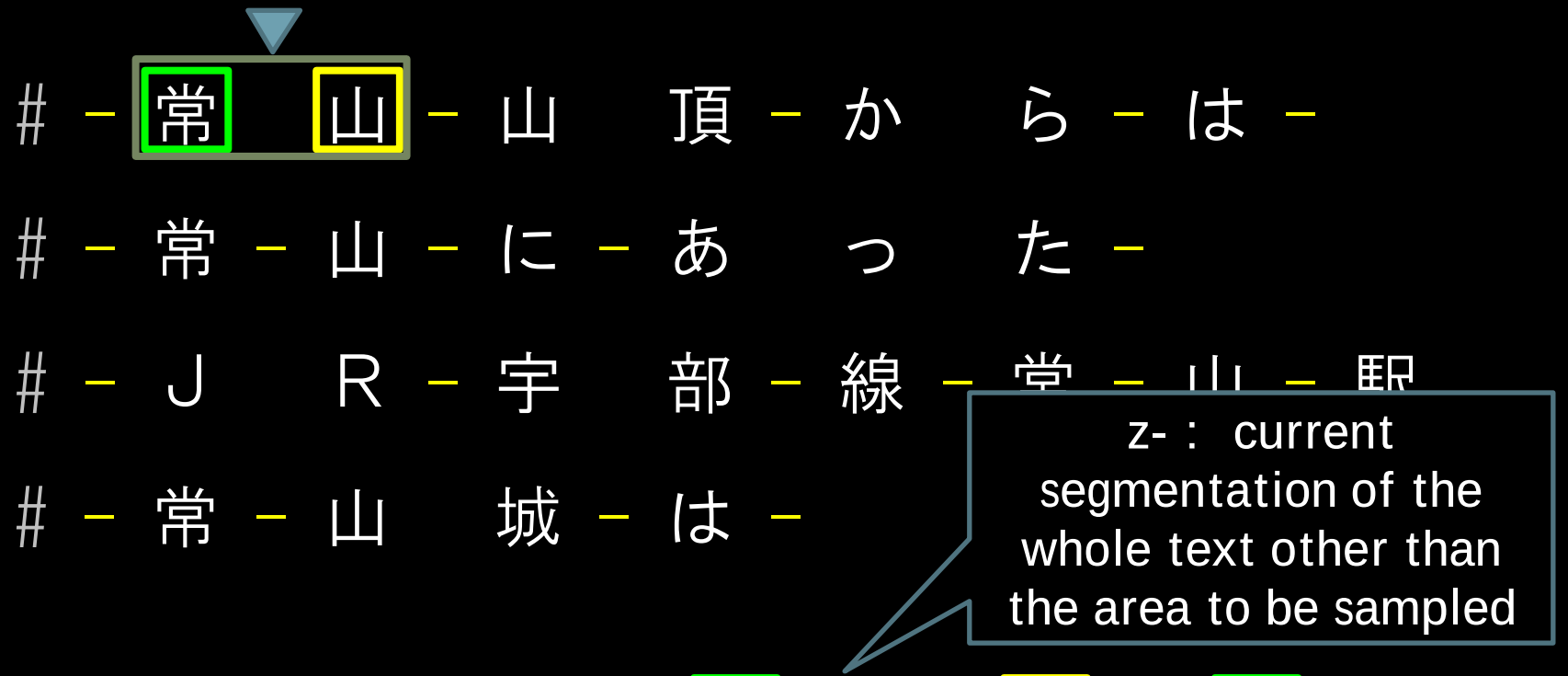
# Token-based Sampling (Unigram)




Boundary: P(常|z-) P(山|z-,常)

Non-boundary: P(常山|z-)

# Token-based Sampling (Unigram)



# Token-based Sampling (Unigram)

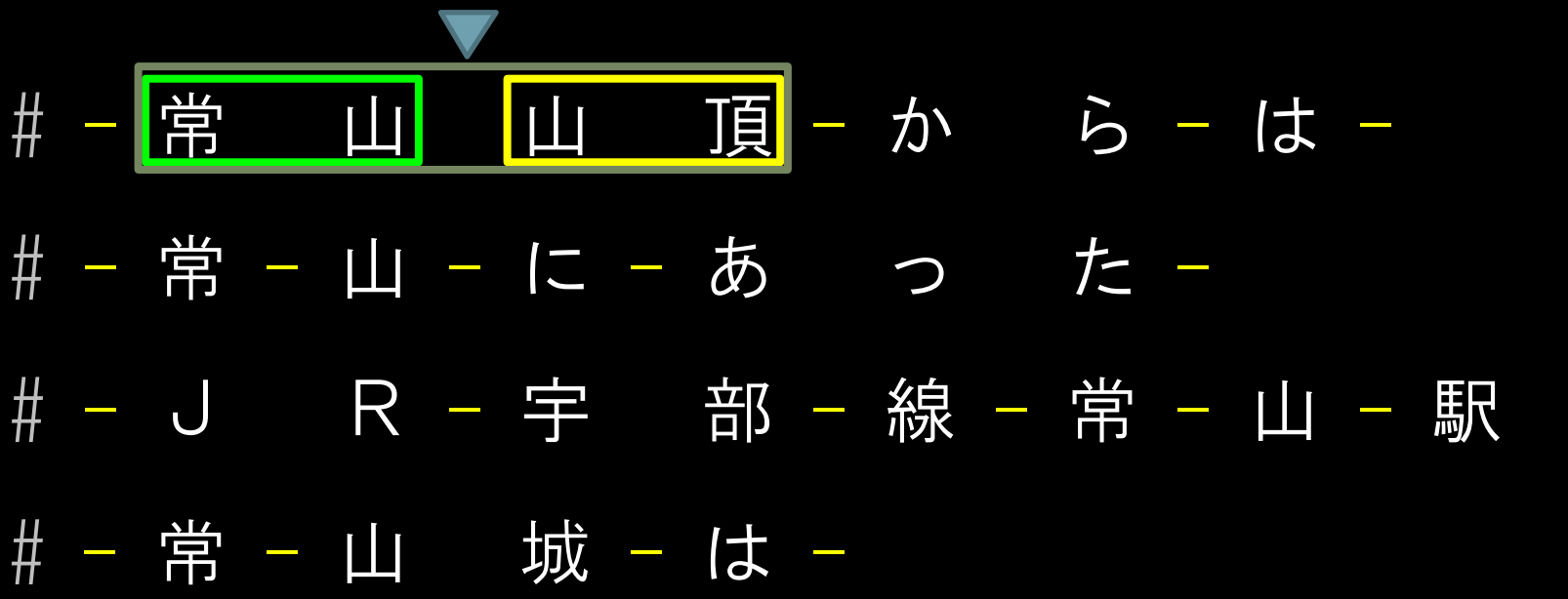
# – 常　山 – 山　頂 – か　ら – は –

# – 常 – 山 – に – あ　っ　た –

# – Ｊ　Ｒ – 宇　部 – 線 – 常 – 山 – 駅

# – 常 – 山　城 – は –

# Token-based Sampling (Unigram)




Boundary: P(常山|z-) P(山頂|z-,常山)

Non-boundary: P(常山山頂|z-)

# Token-based Sampling (Unigram)

# – 常　山 – 山　頂 – か　ら – は –

# – 常 – 山 – に – あ　っ　た –

# – J　R – 宇　部 – 線 – 常 – 山 – 駅

# – 常 – 山　城 – は –

# Type-based Sampling (Unigram)

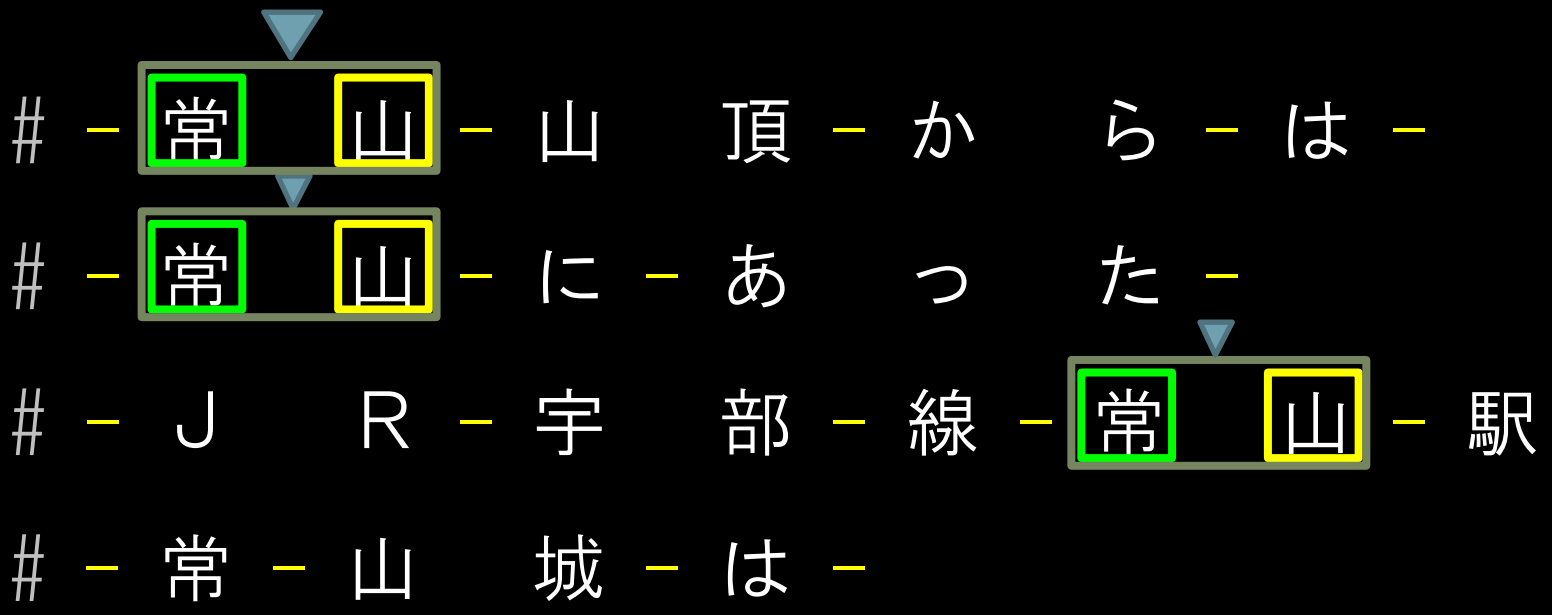
# – 常 – 山 – 山　頂 – か　ら – は –

# – 常 – 山 – に – あ　っ　た –

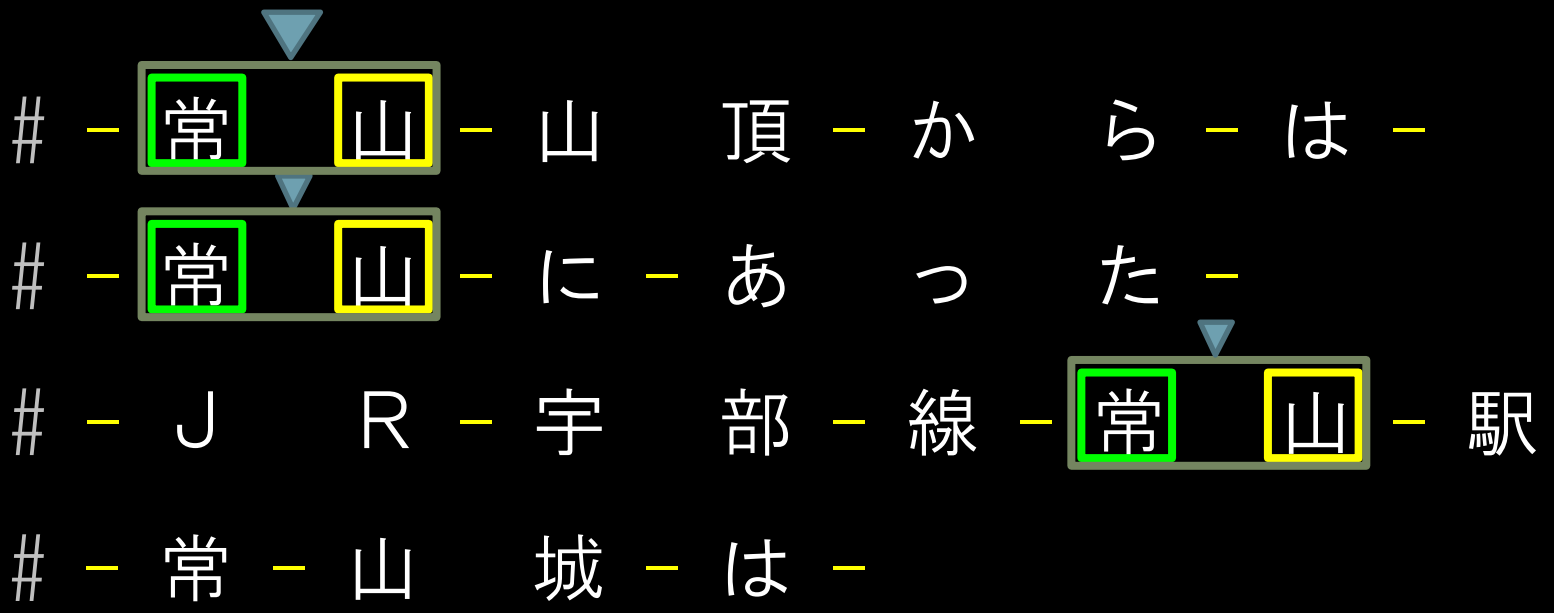
# – Ｊ　Ｒ – 宇　部 – 線 – 常 – 山 – 駅

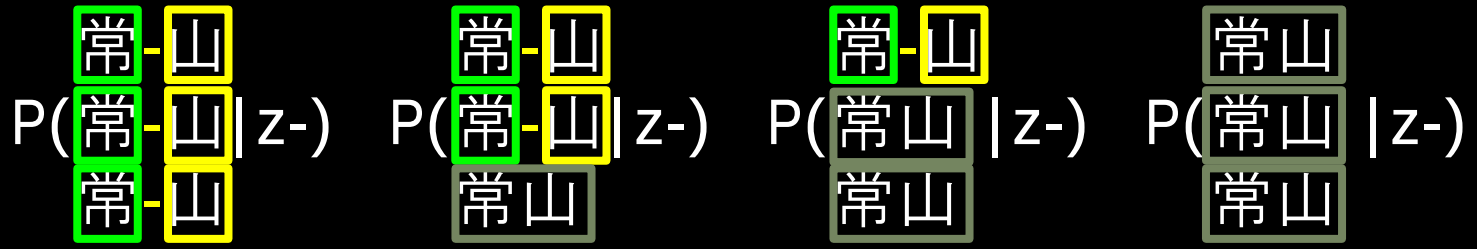
# – 常 – 山　城 – は –

# Type-based Sampling (Unigram)

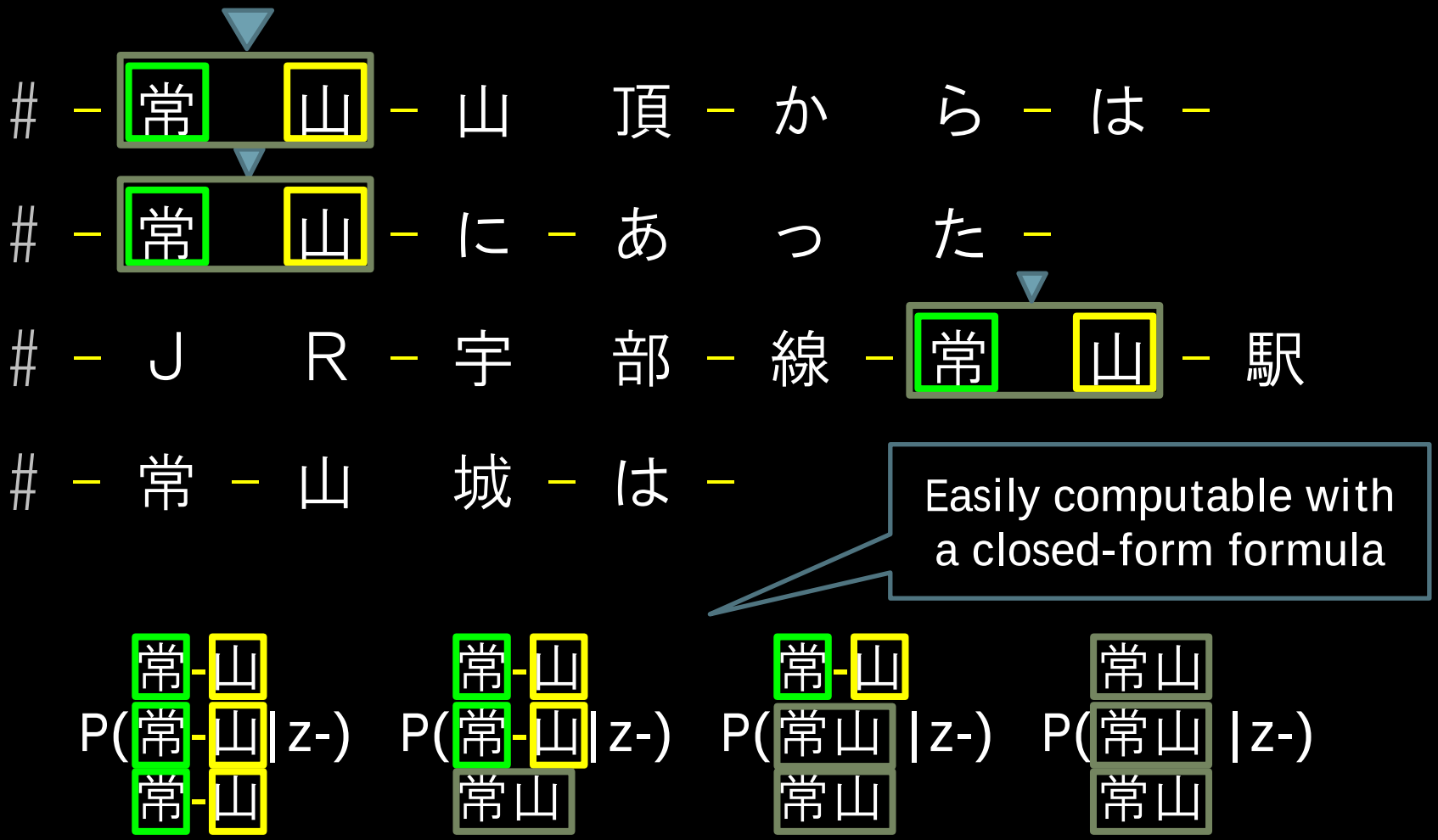


# Type-based Sampling (Unigram)

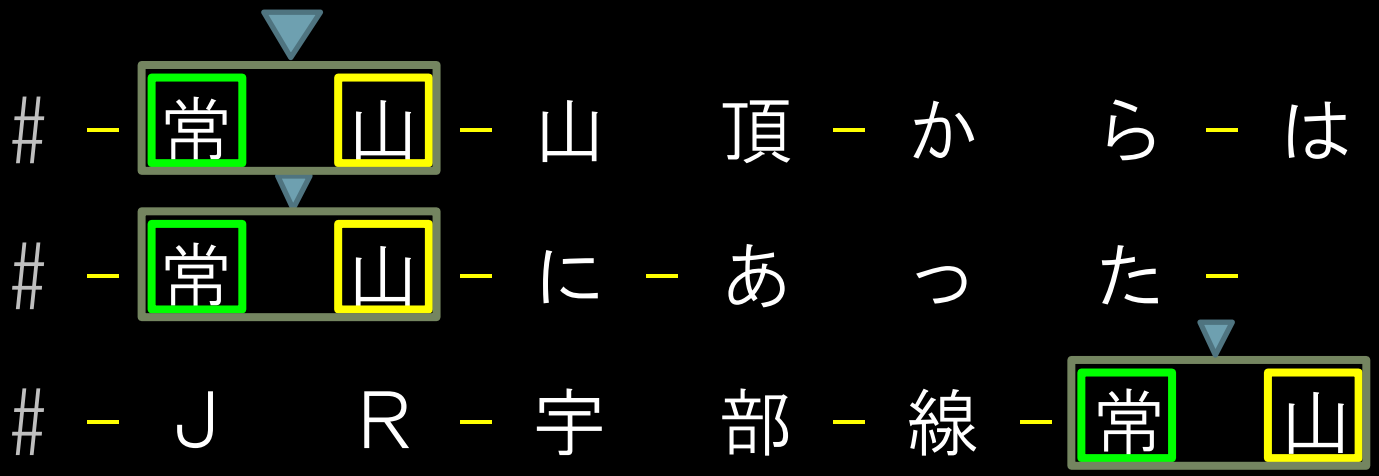


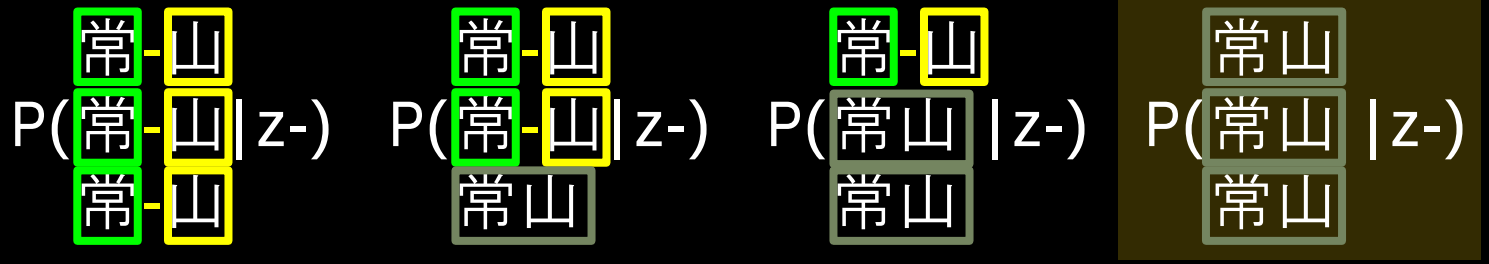
# Type-based Sampling (Unigram)



# Type-based Sampling (Unigram)




Easily computable with a closed-form formula



# Type-based Sampling (Unigram)

# – 常 山 – 山 頂 – か ら – は –

# – 常 山 – に – あ っ た –

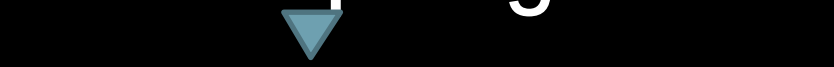
# – J R – 宇 部 – 線 – 常 山 – 駅

# – 常 – 山 城 – は –

# Problem when Applying Type-based Sampling to the Bigram Model

- The joint probability of a type block is no longer tractable due to the dependence on latent assignments
- Approximation by one specific latent assignment is possible, but too cumbersome to consider all the boundary assignment combinations

# Problem when Applying Type-based Sampling to the Bigram Model



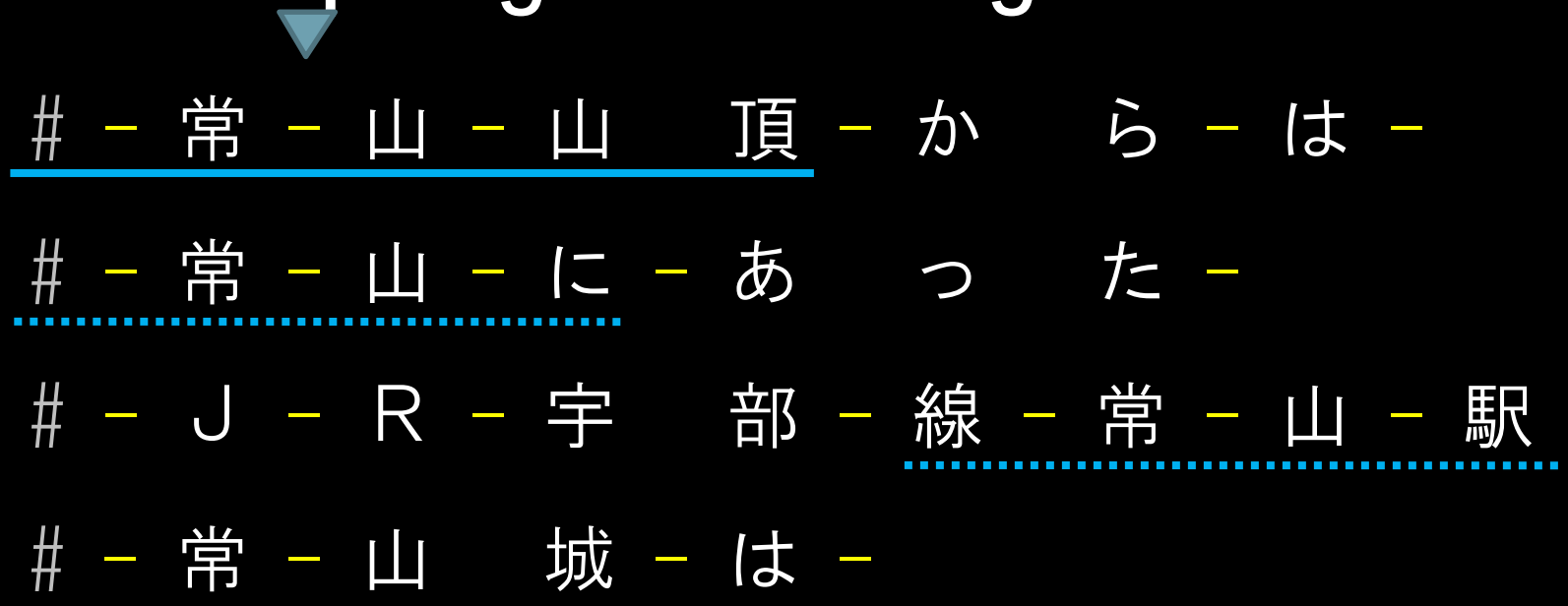
＃ － 常 － 山 － 山　頂 － か　ら － は －

＃ － 常 － 山 － に － あ　っ　た －

＃ － Ｊ － Ｒ － 宇　部 － 線 － 常 － 山 － 駅

＃ － 常 － 山　城 － は －

# Problem when Applying Type-based Sampling to the Bigram Model



# Problem when Applying Type-based Sampling to the Bigram Model

＃ － 常 － 山 － 山　頂 － か　ら － は －

＃ － 常 － 山 － に － あ　っ　た －

＃ － Ｊ － Ｒ － 宇　部 － 線 － 常 － 山 － 駅

＃ － 常 － 山　城 － は －

Closely-related, but a different type in the bigram model

# Solution: Metropolis-Hastings

1. Select a *unigram-level* type block
2. Draw a boundary assignment from a *proposal distribution*
3. Compute the joint probabilities of the current and proposal assignments
4. Accept the proposal according to the acceptance function

# Solution: Metropolis-Hastings

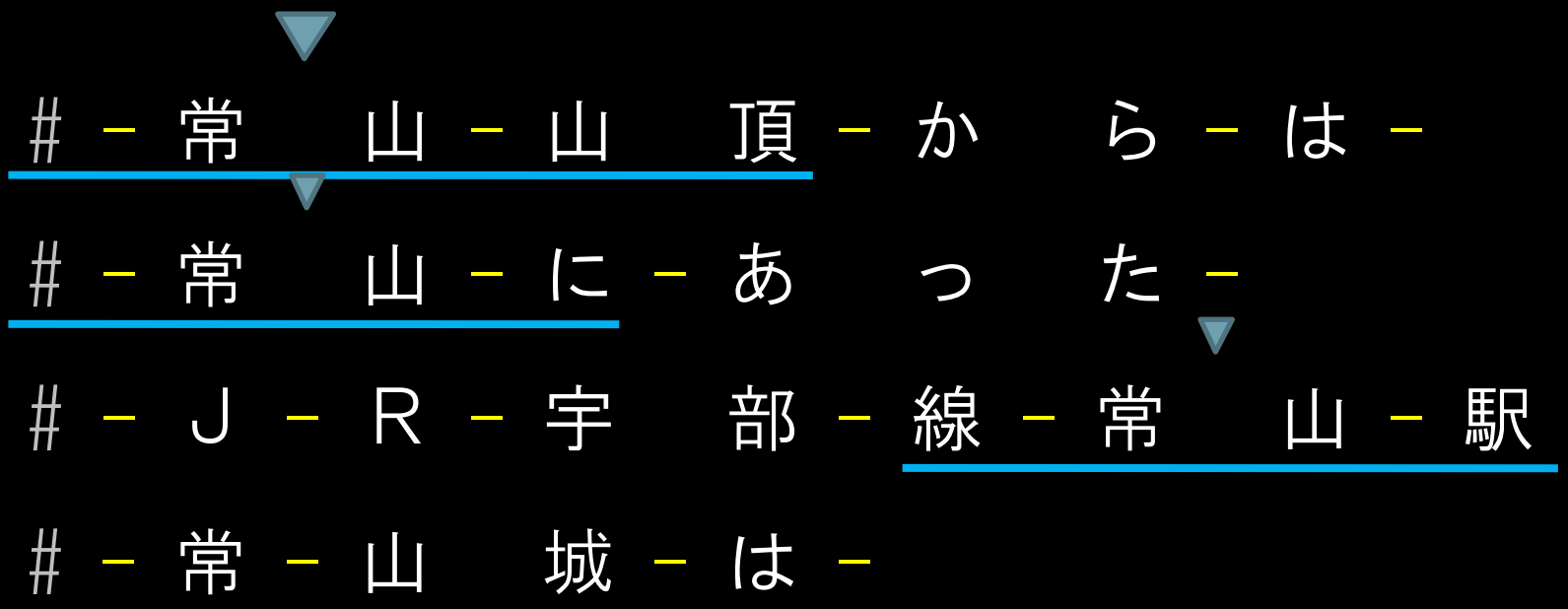
# – 常 – 山 – 山　頂 – か　ら – は –

# – 常 – 山 – に – あ　っ　た –

# – J – R – 宇　部 – 線 – 常 – 山 – 駅

# – 常 – 山　城 – は –

# Solution: Metropolis-Hastings



# Solution: Metropolis-Hastings

# – 常 – 山 – 山頂 – か – ら – は –

# – 常 – 山 – に – あ – っ – た –

# – J – R – 宇 – 部 – 線 – 常 – 山 – 駅

# – 常 – 山 – 城 – は –

current

#-常-山-山頂
P( #-常-山-に |z-)
線-常-山-駅

proposal

#-常山-山頂
P( #-常山-に |z-)
線-常山-駅

# Solution: Metropolis-Hastings

# – 常 – 山 – 山頂 – から – は –

# – 常 – 山 – に – あ – った –

# – J – R – 宇 – 部 – 線 – 常 – 山 – 駅

# – 常 – 山 – 城 – は –

current

#-常-山-山頂
P( #-常-山-に |z-)
線-常-山-駅

proposal

#-常山-山頂
P( #-常山-に |z-)
線-常山-駅

# Solution: Metropolis-Hastings

# – 常　山 – 山　頂 – か　ら – は –

# – 常　山 – に – あ　っ　た –

# – J – R – 宇　部 – 線 – 常　山 – 駅

# – 常 – 山　城 – は –

# Initialization by Dictionary-based Segmenter

- Very close to an optimal
  - But unknown words are often misidentified
- Consistent segmentation
  - Too stable for the token-based sampler to escape
  - Type-based sampler can directly jump to another consistent segmentation

# Summary of Inference Algorithms

| Method | Pros | Cons |
|---|---|---|
| Token-based sampling | Simple | Notoriously slow convergence |
| Type-based sampling (Liang+, 2010) | • Fast convergence<br>• Allow consistent initialization | Only for the unigram model |
| Hybrid type-based sampling (proposed) | • Fast convergence<br>• Allow consistent initialization<br>• Applicable to the bigram model | |

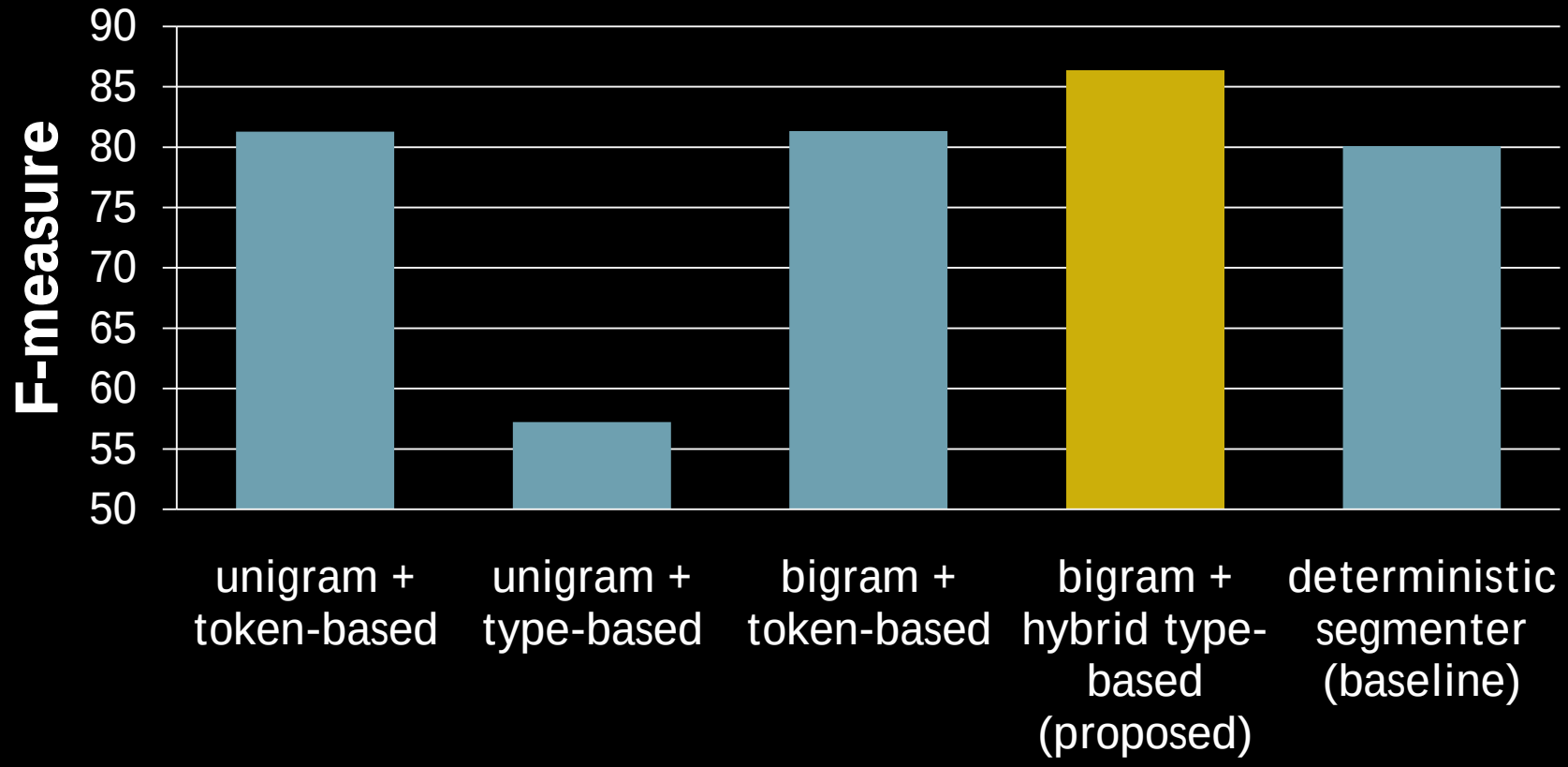
# Outline

- Motivation
- Models and Inference
- Experiments

# Experiments: Settings

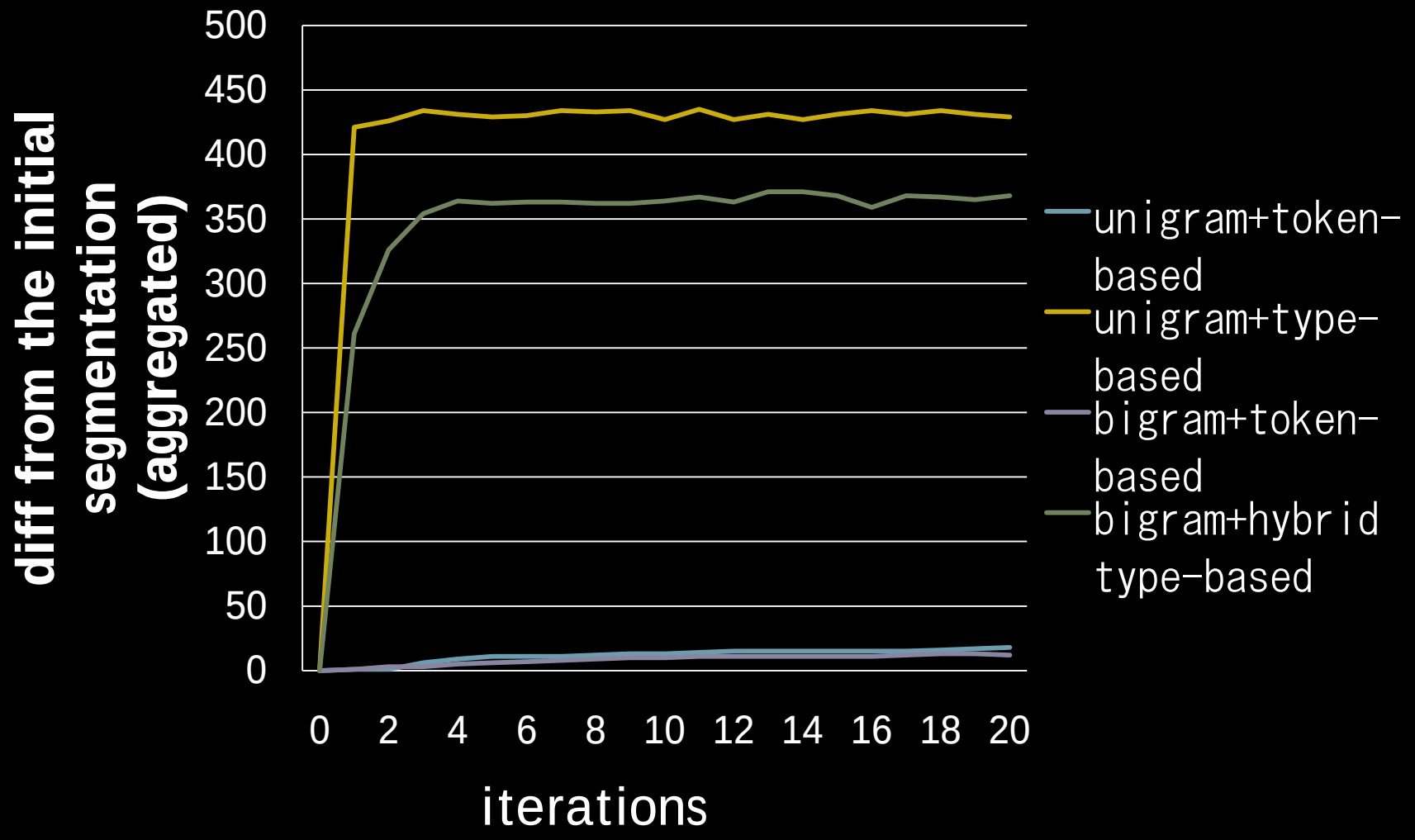
- Data
  - Noun phrases: entries of Japanese Wikipedia
    - Manually annotated 500 entries
  - Related text: article content
- Inference
  - Initialized by the dictionary-based segmenter JUMAN
  - Collect 10 samples after 10 burn-in iterations
  - Output the most frequent segmentation

# Experiments: Results

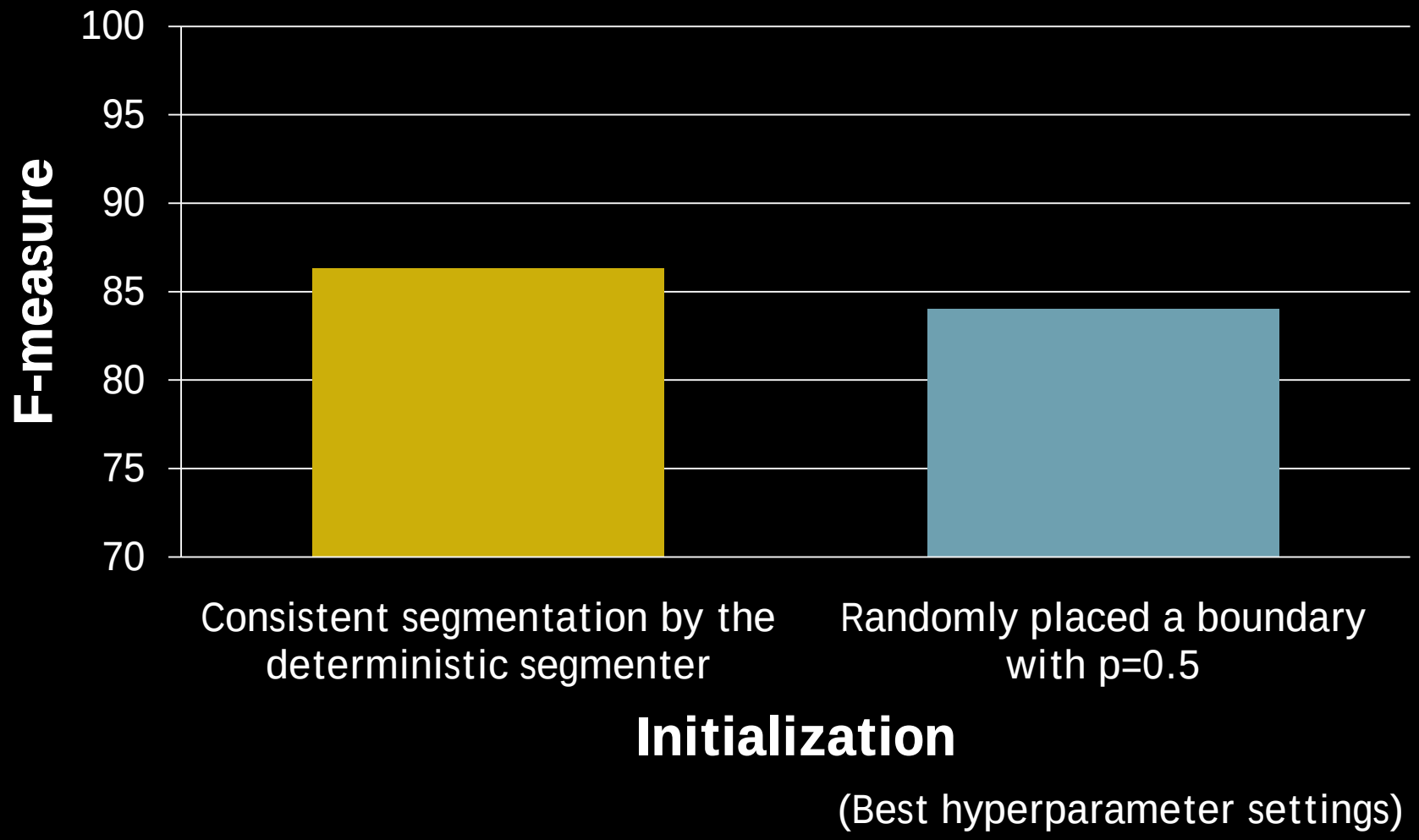



(Best hyperparameter settings)

# Convergence Speed

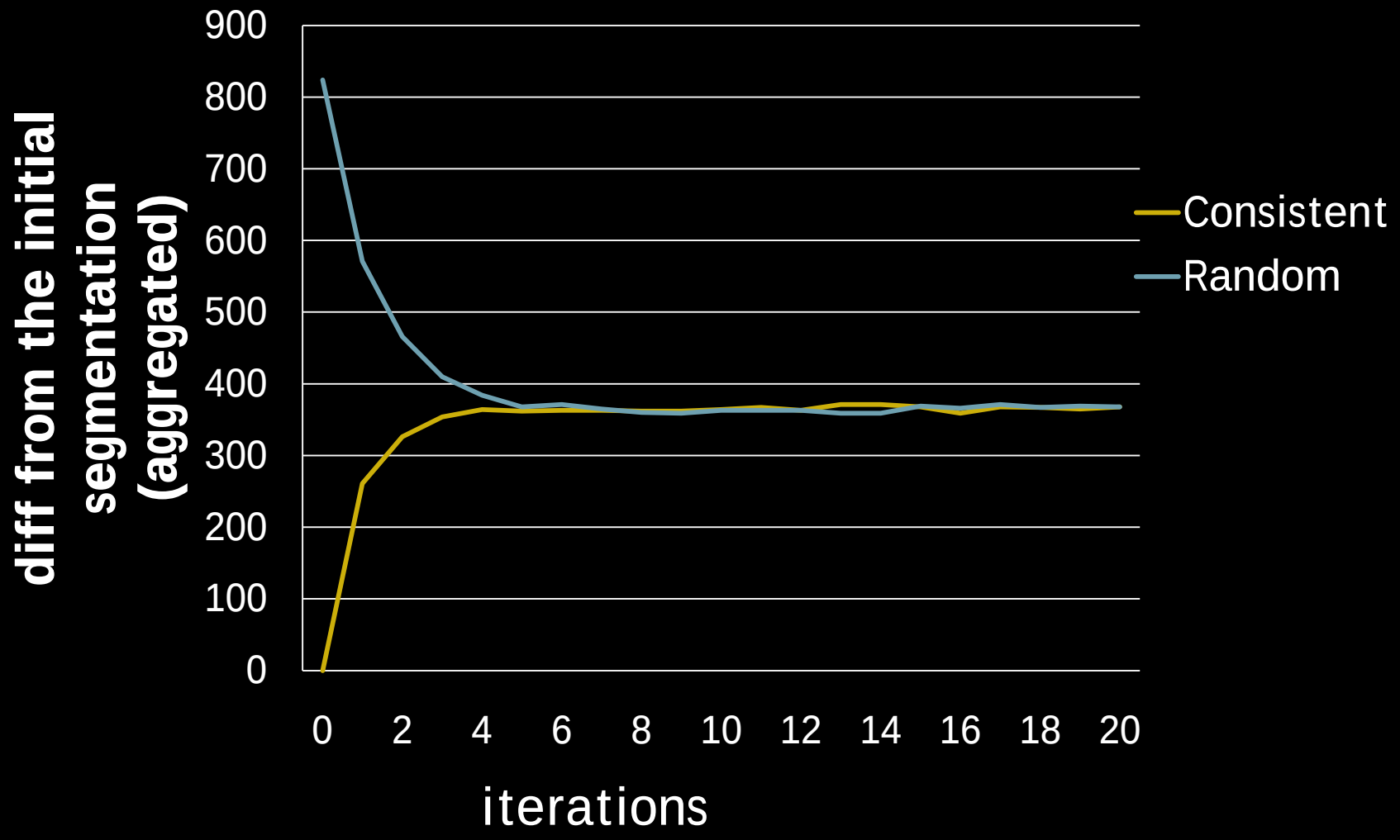


# Effect of Initialization




(Best hyperparameter settings)

# Effect of Initialization



# Conclusions

- Applied statistical language models to Japanese noun phrase segmentation
- Proposed an efficient inference procedure
- Future work
  - Integration into lexical acquisition from text