Do we still need NLP, or is machine learning enough?

Sharon Goldwater

MLSS, 15 January 2019

School of informatics

English Spanish French French - detected

Je ne sais pas!

English Spanish Arabic

I do not know!

Google

nautral language processing

natural language processing
natural language
natural language processing with python
natural language generation

About 8,210,000 results (0.42 seconds)

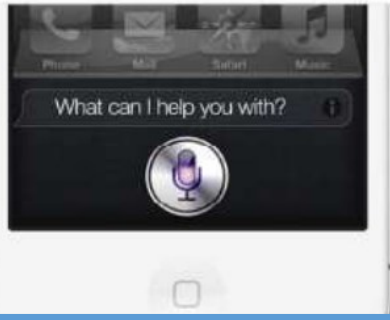
Introducing Dragon 13

Increased speed, accuracy and flexibility make it our best Dragon yet.

Learn more

Features Built-in Apps From the App Store iOS iCloud Tech Specs

Learn more about Siri.



Google who is the first indian president

Rajendra Prasad

The 1st President of India

List of **Presidents** of **India** - Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/List_of_**Presidents**_of_**India**
The **President** of **India** is the head of state and **first** citizen of **India**. The **President** is also the Commander-in-Chief of the **Indian** Armed Forces. Although the ...
Zakir Hussain - Rajendra Prasad - VV Giri - R. Venkataraman

Connect with us

cnet Search Reviews News Video How To Smart Home Cars Deals US

CNET › Software › The many faces of Cortana: How Microsoft's virtual assistant wants to woo the world

The many faces of Cortana: How Microsoft's virtual assistant wants to woo the world

Virtual assistants have become commonplace in modern technology, but Microsoft thinks it knows how to push its Cortana a step beyond the rest.

Sources: google.co.uk, nuance.co.uk, apple.com, www.amazon.co.uk, cnet.com

Deep learning revolution

- Many standard architectures now used across fields
- Much less need for feature engineering (= domain knowledge)
- Easy to use toolkits for building models

So, lower barrier to entry.

Deep learning revolution

- Many standard architectures now used across fields
- Much less need for feature engineering (= domain knowledge)
- Easy to use toolkits for building models

So, lower barrier to entry.

Is NLP/linguistic knowledge unnecessary?

Outline

1. Why did we need NLP in the first place?

2. What NNets do and do not bring to the table
   (i.e., why we still need NLP)

Outline

1. Why did we need NLP in the first place?

2. What NNets do and do not bring to the table
   (i.e., why we still need NLP)

Goals:

- teach you a little bit about language/linguistics

- briefly review a range of interesting work (very high level)

Why did we need NLP in the first place?

Language was/is "special".

- Discrete (some disconnect between NLP and speech!)
- Structured
- Uniquely human

Example: language modeling

(At the edge of NLP...in fact will motivate more NLP-ish models)

- Input: sequence of words (or chars) $\vec{w} = w_1 \ldots w_N$
- Output: $P(w_1 \ldots w_N)$ or $P(w_{N+1}|w_1 \ldots w_N)$

Example: language modeling

(At the edge of NLP...in fact will motivate more NLP-ish models)

- Input: sequence of words (or chars) $\vec{w} = w_1 \ldots w_N$
- Output: $P(w_1 \ldots w_N)$ or $P(w_{N+1}|w_1 \ldots w_N)$
- Uses:
  - Predictive text completion
  - (Traditionally) speech recognition, machine translation
  - (Recently) pretraining for other NLP tasks

Traditional n-gram language models

We want $P(w_{N+1}|w_1 \ldots w_N)$.

- For vocab V, that's $|V|^N$ conditioning contexts!
- So, make a Markov assumption. E.g., for a trigram model:

$$P(\vec{w}) \approx \prod_{i=1}^{N} P(w_i|w_{i-2}, w_{i-1})$$

- Trigram model example: $P(\text{the, cat, ate, here}) \approx$
  $P(\text{here} \mid \text{cat, ate}) \cdot P(\text{ate} \mid \text{the, cat}) \cdot P(\text{cat} \mid \text{the, } \langle\text{s}\rangle) \cdot P(\text{the} \mid \langle\text{s}\rangle, \langle\text{s}\rangle)$

Naive (MLE) estimation of trigram probs

- Collect bigram and trigram counts over a large text corpus:

$$P_{\text{MLE}}(w_3|w_1, w_2) = \frac{C(w_1, w_2, w_3)}{C(w_1, w_2)}$$

- However, even in a very large corpus, we still encounter many rare/unseen n-grams (esp. if n is larger than 3).

- Various ways to be clever...

Insight #1: interpolation

- When building a trigram model, suppose we never observe
  - Scottish beer drinkers
  - Scottish beer eaters
- We should still be able to assign the first higher probability by looking at just one context word instead of two.

Insight #1: interpolation

- Higher and lower order N-gram models have different strengths and weaknesses
  - high-order N-grams are sensitive to more context, but have sparse counts
  - low-order N-grams consider only very limited context, but have robust counts

- So, combine them:

$$P_{\text{INTRP}}(w_3|w_1,w_2) = \lambda_1\ P_1(w_3) \qquad \text{P}_1(\text{drinkers})$$
$$+\ \lambda_2\ P_2(w_3|w_2) \qquad \text{P}_2(\text{drinkers}|\text{beer})$$
$$+\ \lambda_3\ P_3(w_3|w_1,w_2) \qquad \text{P}_3(\text{drinkers}|\text{Scottish, beer})$$

Insight #2: diversity of histories

Example from MacKay and Bauman Peto (1994):

> Imagine, you see, that the language, you see, has, you see, a frequently occurring couplet, 'you see', you see, in which the second word of the couplet, 'see', follows the first word, 'you', with very high probability, you see. Then the marginal statistics, you see, are going to become hugely dominated, you see, by the words 'you' and 'see', with equal frequency, you see.

- $P(\text{see})$ and $P(\text{you})$ both high, but *see* nearly always follows *you*.
- So $P(\text{see}|novel)$ should be much lower than $P(\text{you}|novel)$.

Insight #2: diversity of histories

- A less contrived example: the word Angeles
  - Not that uncommon as an English unigram, but almost invariably follows Los.
  - So, if we use MLE unigram model in interpolation, will over-predict Angeles in novel contexts.

- **Kneser-Ney smoothing** bases the lower-order models on diversity of contexts rather than occurrence counts; state-of-the-art until recently.

Insight #2: diversity of histories

- A less contrived example: the word Angeles
  - Not that uncommon as an English unigram, but almost invariably follows Los.
  - So, if we use MLE unigram model in interpolation, will over-predict Angeles in novel contexts.

- **Kneser-Ney smoothing** bases the lower-order models on diversity of contexts rather than occurrence counts; state-of-the-art until recently.

- KN can also be interpreted as a hierarchical Bayesian nonparametric model (Goldwater et al., 2006; Teh, 2006).

Insight #3: sub-word information

Sub-word information can help.

- $P(\text{She is plecting}) > P(\text{She is plection})$
- $P(\text{Dr. Sklare ate}) > P(\text{Dr. sklare ate})$

Not easy to incorporate in generative models, but used in discriminative language models.

Problem #1: similarity

- Imagine two words, one much more frequent than the other.
  - salmon
  - swordfish

Problem #1: similarity

- Imagine two words, one much more frequent than the other.
  - salmon
  - swordfish
- $P(\text{salmon}|\text{caught two})$ tells us nothing about $P(\text{swordfish}|\text{caught two})$.
- More generally, n-gram models have no way to share statistical strength between **similar words**.
  - If the contexts where I did see them are similar, then I can expect them to behave similarly in new contexts as well.

Problem #2: fixed-size history

Linguistic dependencies can be arbitrarily long.

- E.g., number agreement in English:

  The **kids like** books vs The **kid likes** books

Problem #2: fixed-size history

Linguistic dependencies can be arbitrarily long.

- E.g., number agreement in English:

  The **kids like** books vs The **kid likes** books

- But also:

  The **kids** in the shop **like** books

  The **kids** in the shop across the street **like** books

  The **kids** in the big shop across the street where I bought my glasses really really **like** books

Quiz questions

Don't know/true/false:

1. Inability to capture long-distance dependencies (like the above) is a fundamental problem of traditional (non-NN) NLP techniques.

2. NN methods (e.g., LSTM) **are able to** capture such dependencies.

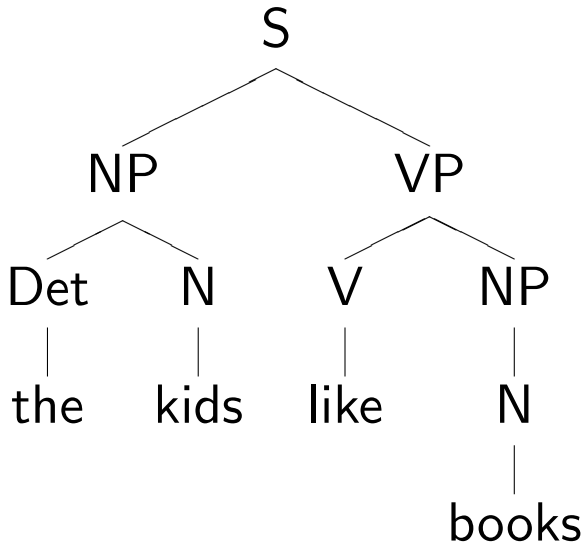
3. NN methods (e.g., LSTM) **do** capture such dependencies.

Beyond n-grams

Traditional NLP approaches **can and do** deal with long-distance dependencies.

- Dependencies are "long-distance" in n-gram (and other sequence) models.
- But not if we adopt the linguist's view and use hierarchical (tree-structured) models.
- Various specific models, but virtually everyone agrees on trees.

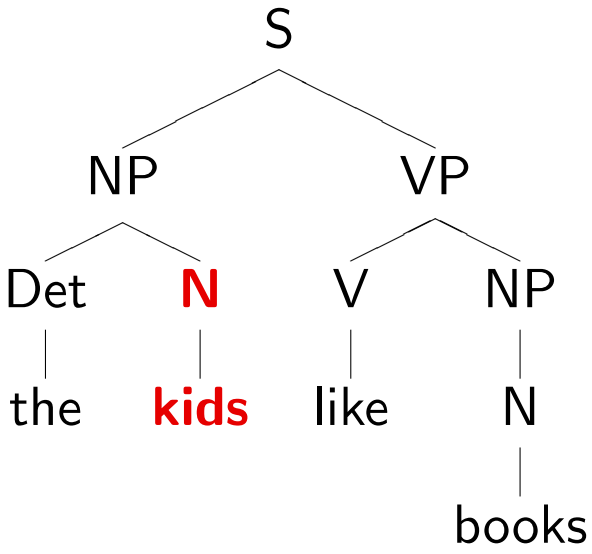
The latent trees

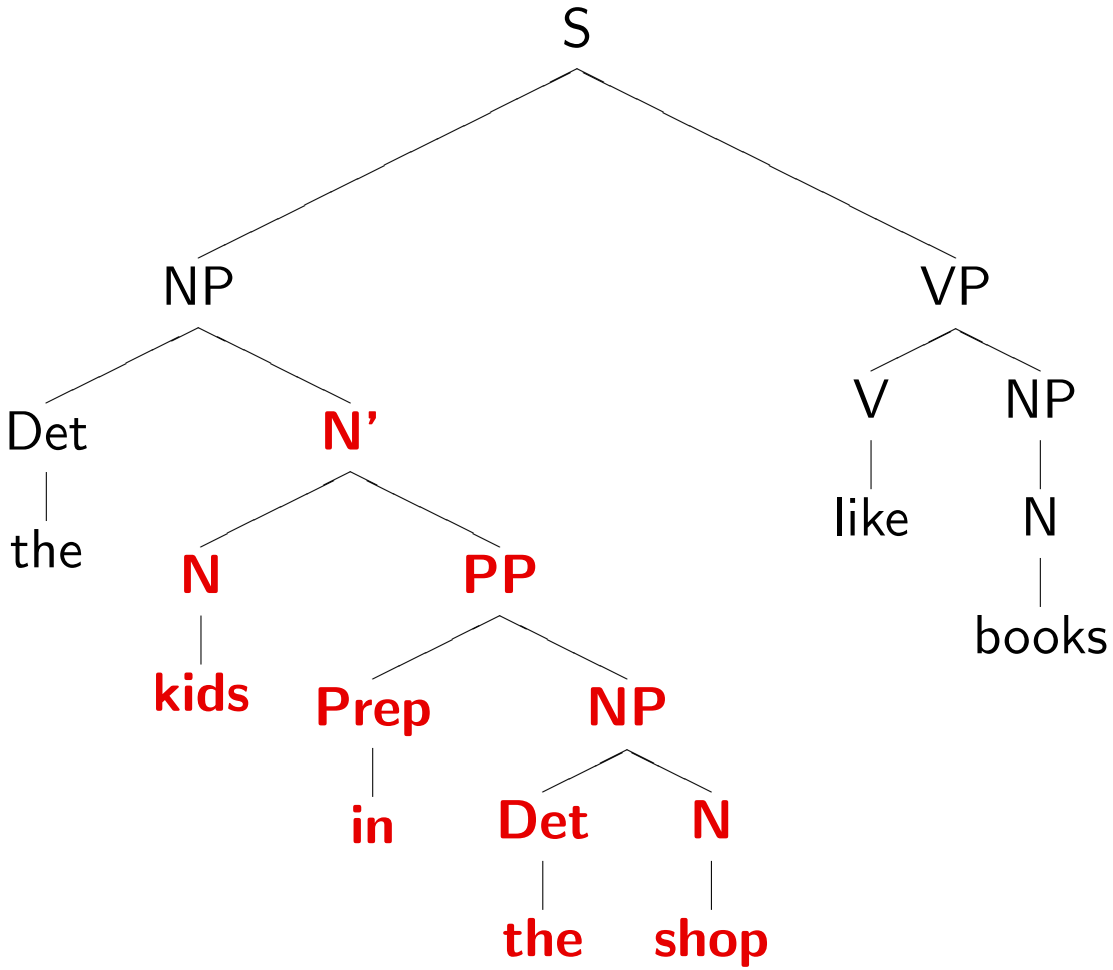


S → NP VP	Det → the
VP → V NP	N → kids
NP → Det N	V → like
...	N → books
	...

(Can add probs to rules)

The latent trees

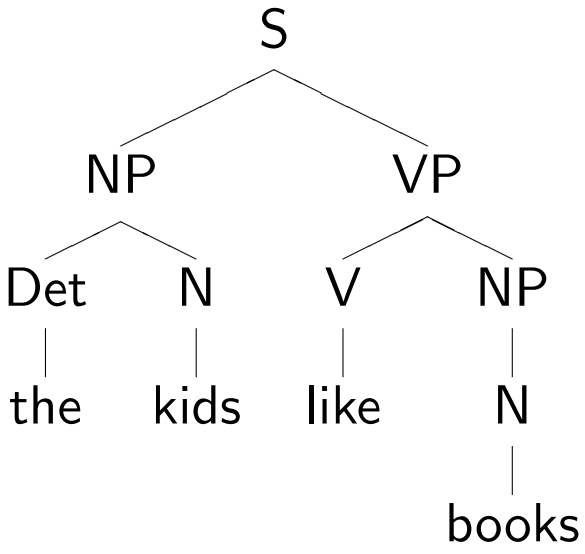


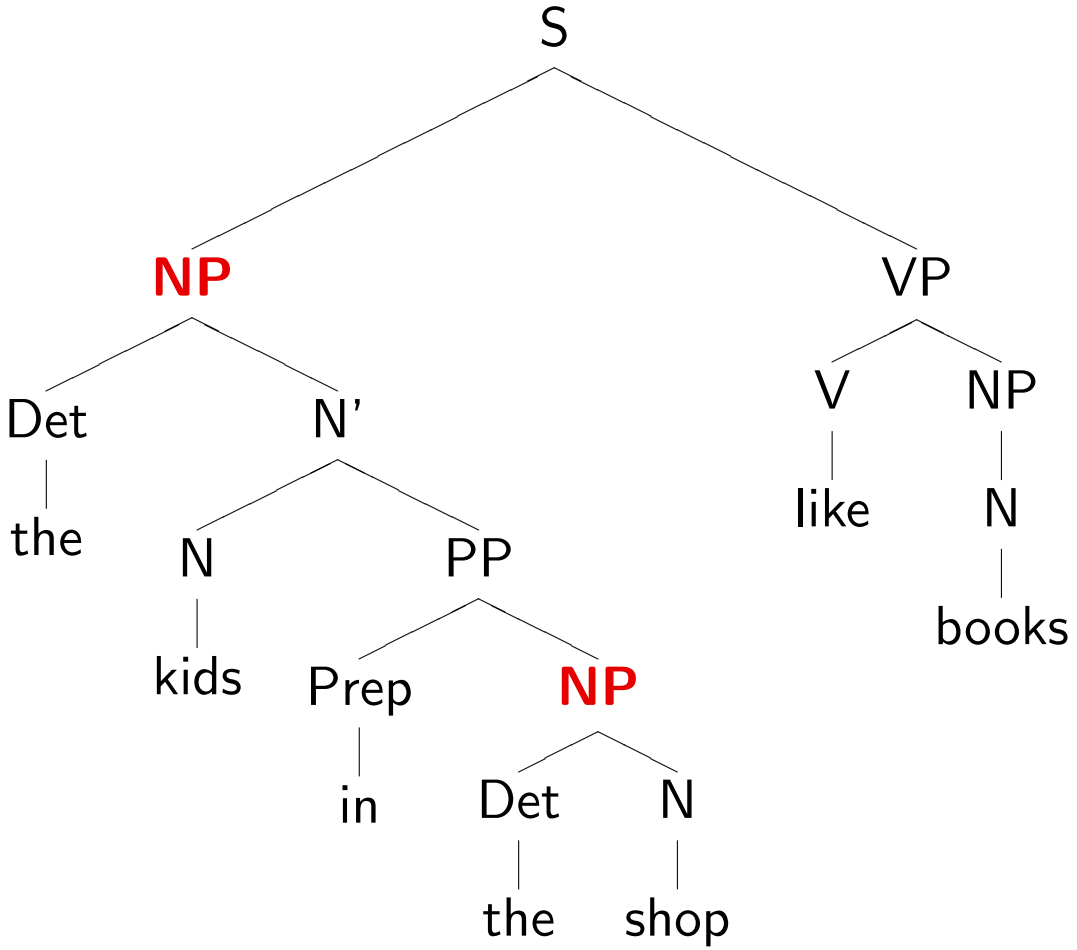


$S \rightarrow$ NP VP
VP $\rightarrow$ V NP
NP $\rightarrow$ Det N
NP $\rightarrow$ Det N'
N' $\rightarrow$ N PP
PP $\rightarrow$ Prep NP
...

Det $\rightarrow$ the
N $\rightarrow$ kids
V $\rightarrow$ like
N $\rightarrow$ books
...

The latent trees

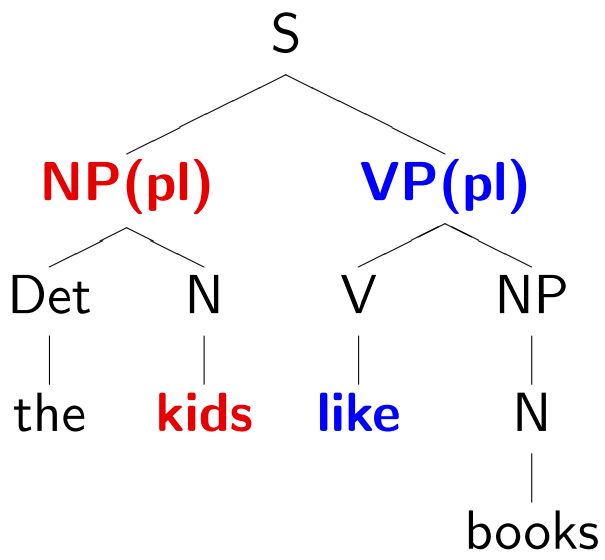




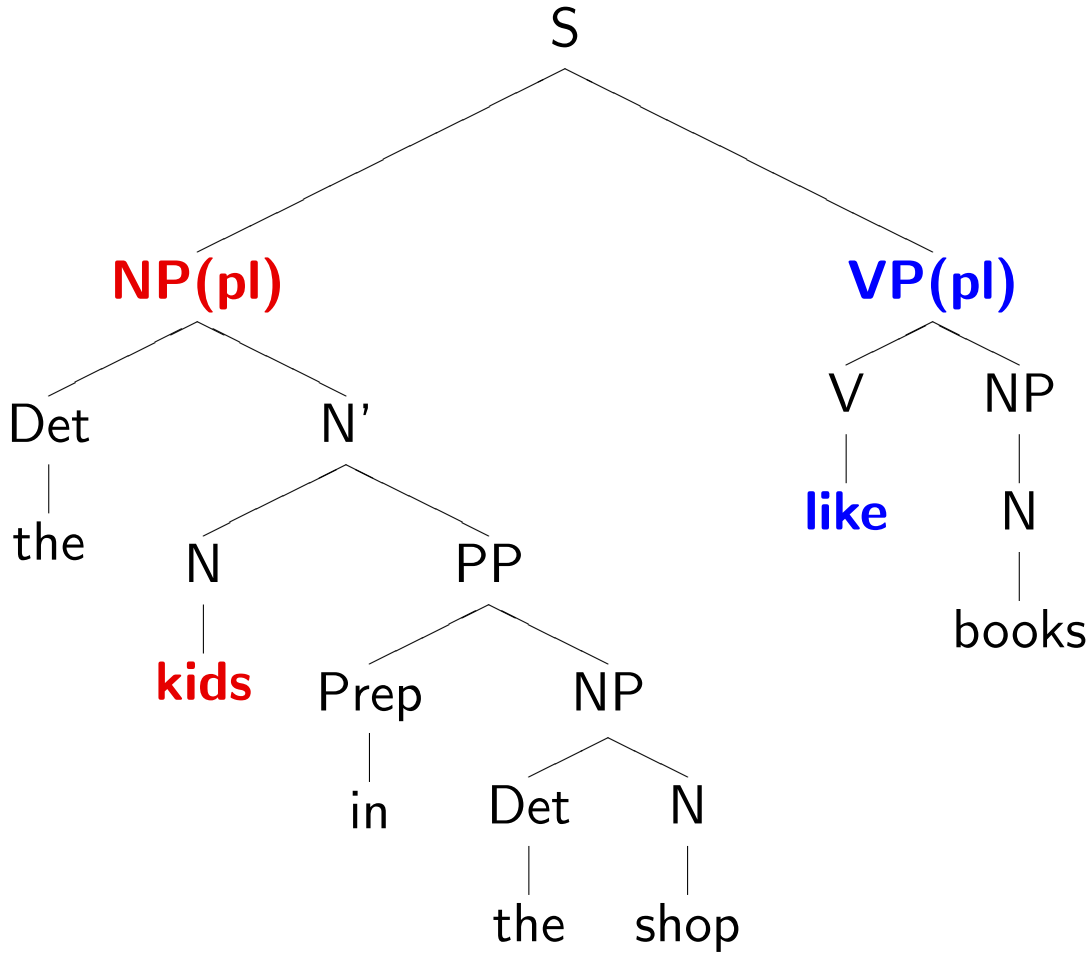
$S \rightarrow$ NP VP
VP $\rightarrow$ V NP
NP $\rightarrow$ Det N
NP $\rightarrow$ Det N'
N' $\rightarrow$ N PP
PP $\rightarrow$ Prep NP
...

Det $\rightarrow$ the
N $\rightarrow$ kids
V $\rightarrow$ like
N $\rightarrow$ books
...

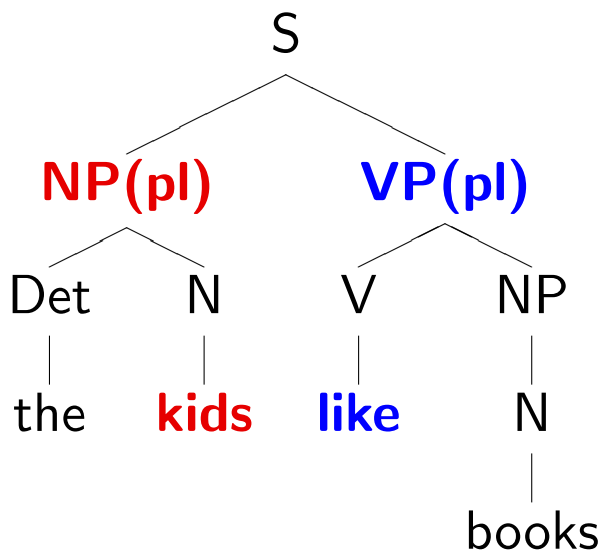
Adding number agreement

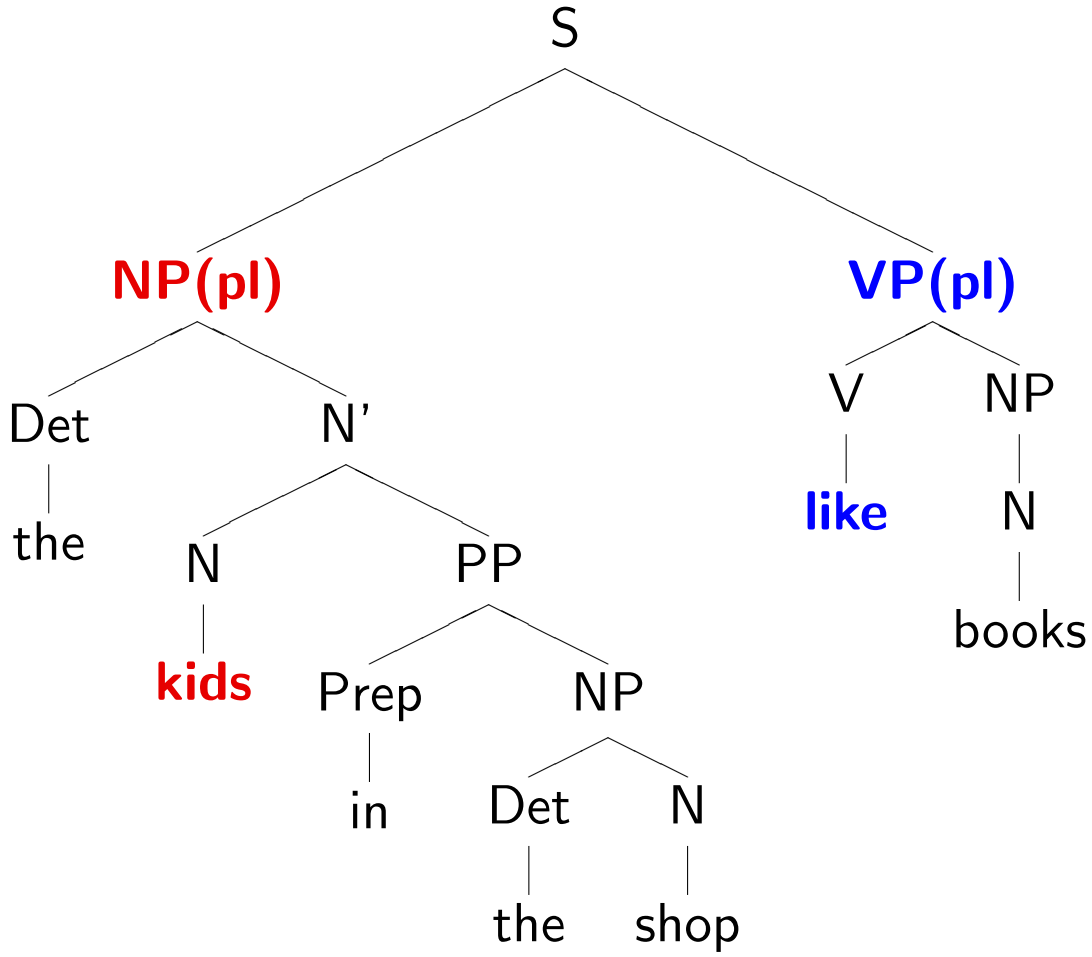


An XP gets its number from the topmost X under it.



Adding number agreement

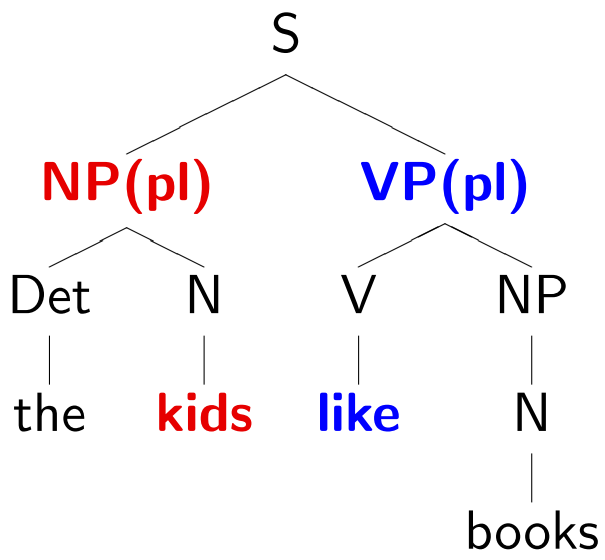


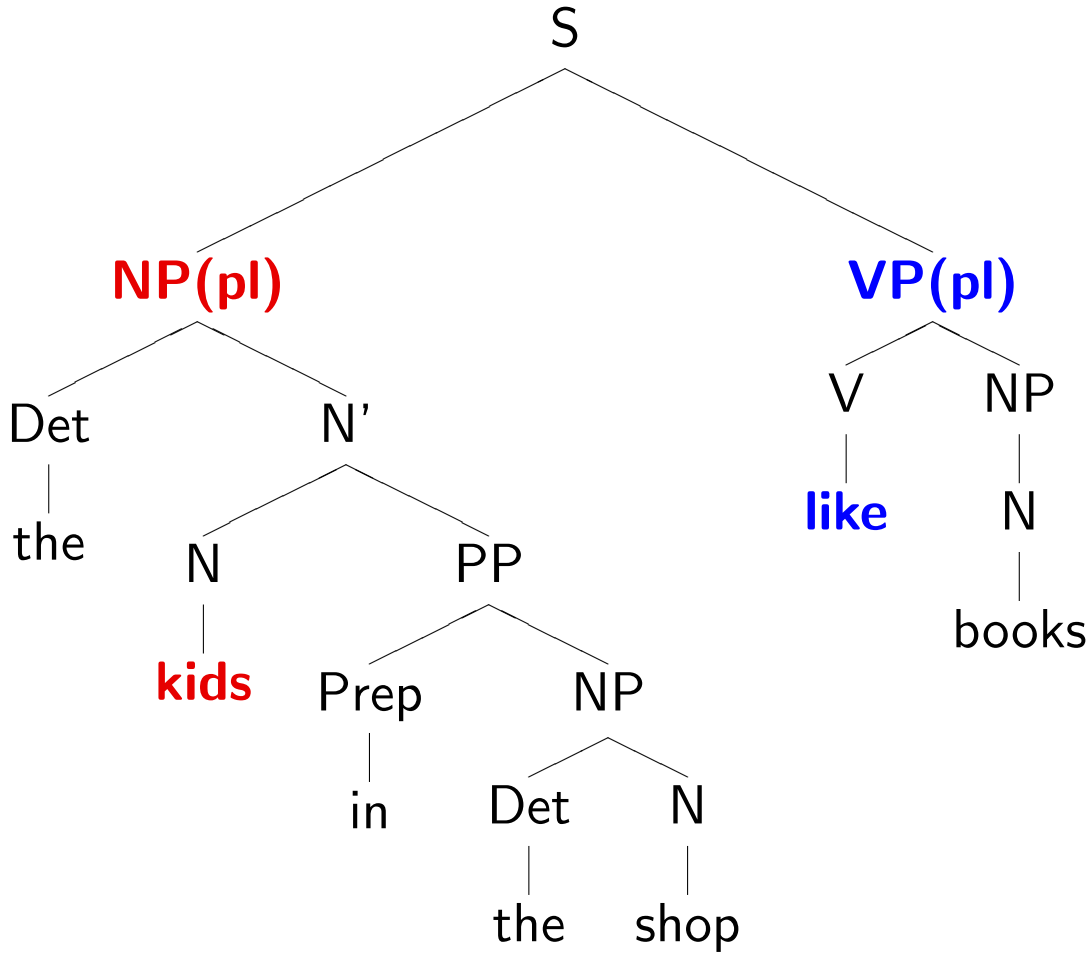


An XP gets its number from the topmost X under it.

An S must consist of NP and VP with matching number.

Adding number agreement





An XP gets its number from the topmost X under it.

An S must consist of NP and VP with matching number.

Agreement now in adjacent nodes, not long-distance.

Summary: pre-deep NLP

Models typically

- build in linguistically motivated inductive biases (structure)
- use probabilistic models over discrete/symbolic representations

As a result,

- new models/tasks often require hand-designed inference algorithms and feature engineering
- restricted ability to model similarity limits generalization

Outline

1. Why did we need NLP in the first place?

2. What NNets do and do not bring to the table: A tale of learning and inductive bias.
   - Example 1: syntactic structure
   - Example 2: subword structure
   - Current workarounds and why NLP is still not just ML.

Consider: RNN language model

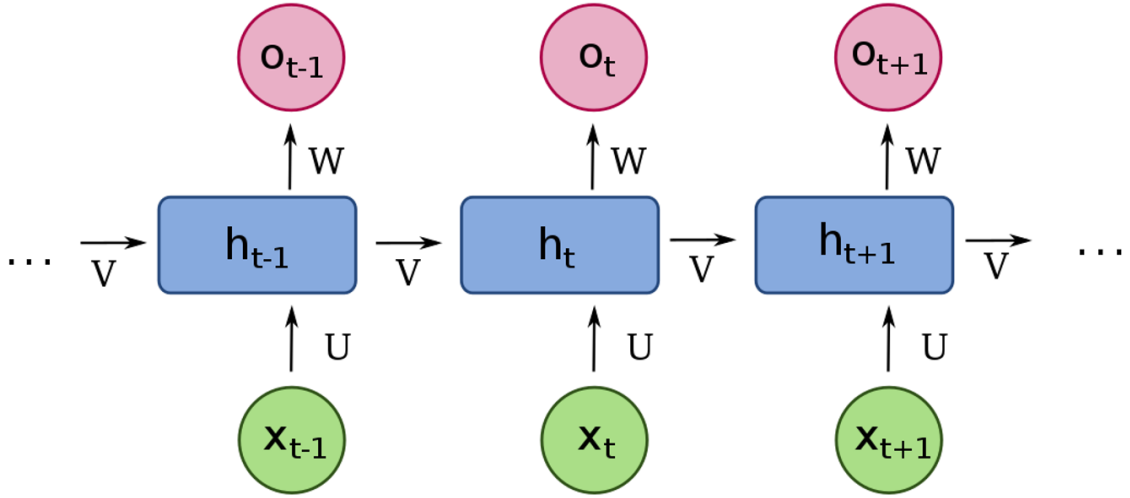
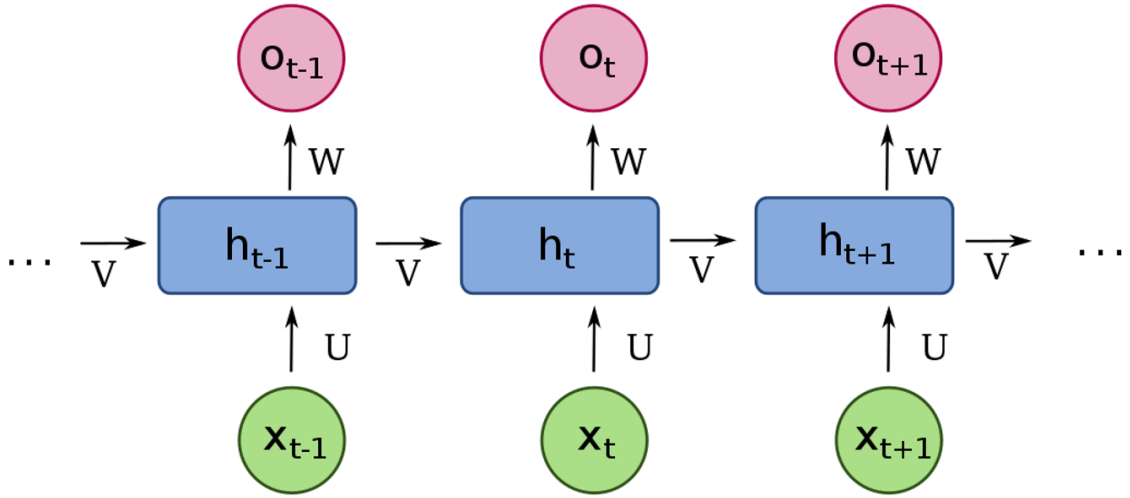


Figure: modified from one by Franois Deloche, CC-BY-SA-4.0

Units: LSTM, GRU, etc

Target $\mathbf{o_t}$ is word $\mathbf{x_{t+1}}$

Consider: RNN language model



Units: LSTM, GRU, etc

Target $\mathbf{o_t}$ is word $\mathbf{x_{t+1}}$

Figure: modified from one by Franois Deloche, CC-BY-SA-4.0

- Distributed representations capture similarity and generalize well.
- Practical limits on dependency length are much longer than n-gram model.
- In principle, can learn which aspects of history are important.

"In principle, can learn..."

- Do these models implicitly learn linguistically structured generalizations in practice?

- Even if so, at what cost?

These are questions of **inductive bias**.

Unless we can say "yes, and efficiently", we shouldn't throw out NLP/linguistics.

Example 1: syntactic structure

Test case: what do RNNs learn about number agreement?

- Examine sentences with target **noun** and **verb**, plus zero or more attractors.

the **kids** from the party last week **are**/*is ...

Example 1: syntactic structure

Test case: what do RNNs learn about number agreement?

- Examine sentences with target **noun** and **verb**, plus zero or more attractors.

  the **kids** from the party last week **are**/*is ...

  – Sentences can be extracted from a corpus (Linzen et al., 2016; Gulordava et al., 2018)...

  – ...or constructed with specific properties (Marvin and Linzen, 2018)

Properties of example sentences

- Corpus sentences tend to have few words and few attractors between target noun and verb.

- Constructed sentences are controlled for these and other factors:
  - the **officer** near the skaters **laughs**/*laugh (prep phrase)
  - the **officer** that loves the skaters **laughs**/*laugh (rel clause)

Properties of example sentences

- Corpus sentences tend to have few words and few attractors between target noun and verb.

- Constructed sentences are controlled for these and other factors:
  - the **officer** near the skaters **laughs**/*laugh (prep phrase)
  - the **officer** that loves the skaters **laughs**/*laugh (rel clause)
  - the **officer** smiles and **laughs**/*laugh (short conjunction)
  - the **officer** writes in a journal every day and **laughs**/*laugh (long conjunction)

Evaluation setup

From Gulordava et al. (2018); Marvin and Linzen (2018):

- Train RNN LM on 90m words of Wikipedia
- Compare probabilities of two sequences that differ only in verb agreement:
  - the officer near the skaters laughs
  - the officer near the skaters laugh
- Does the RNN assign higher probability to the correct option?
- Can also have humans do the same task.

Results

- RNN accuracy on corpus examples (92.1%) only slightly below humans (94.5%).
  - Both do (similarly) worse with more attractors.

Results

- RNN accuracy on corpus examples (92.1%) only slightly below humans (94.5%).
  - Both do (similarly) worse with more attractors.
- But RNN much worse than humans on constructed examples.
  - Several types are $< 60\%$, with humans $> 80\%$.

Conclusion: RNNs are good at common examples but don't seem to encode structure more generally.

Diff't architectures, diff't inductive bias

How do RNNs compare to other NNet architectures?

- Recurrent neural network grammar (RNNG; Dyer et al. (2016))
  - designed to explicitly encode hierarchy through structure-building actions (shift/reduce).
  - Trained on sentences with their trees (produced by parser)

Diff't architectures, diff't inductive bias

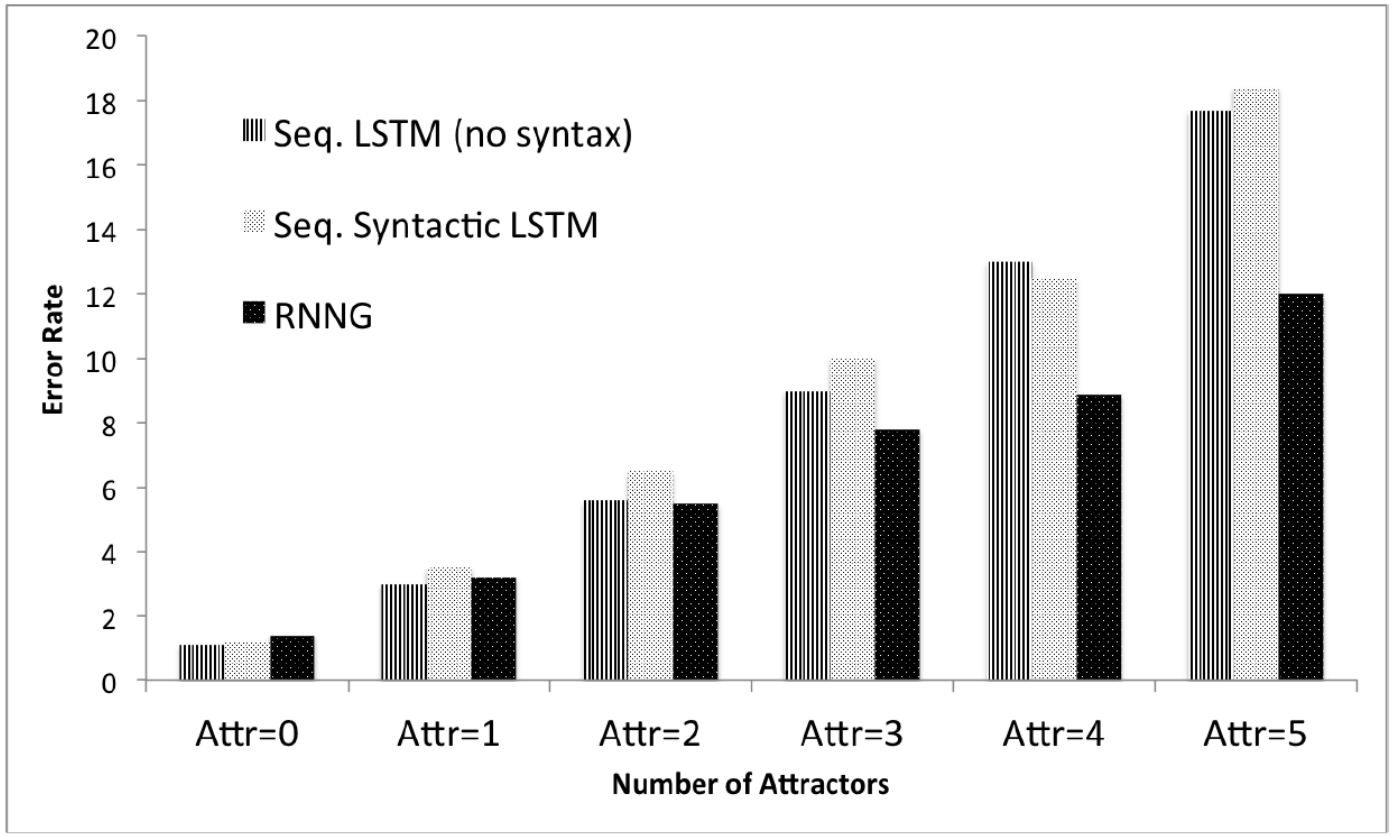
How do RNNs compare to other NNet architectures?

- Recurrent neural network grammar (RNNG; Dyer et al. (2016))
  - designed to explicitly encode hierarchy through structure-building actions (shift/reduce).
  - Trained on sentences with their trees (produced by parser)

- Fully-attentional network (FAN or Transformer; Vaswani et al. (2017))
  - Attention mechanism allows any word in the sentence to influence all others.
  - Demonstrated to work well for machine translation.

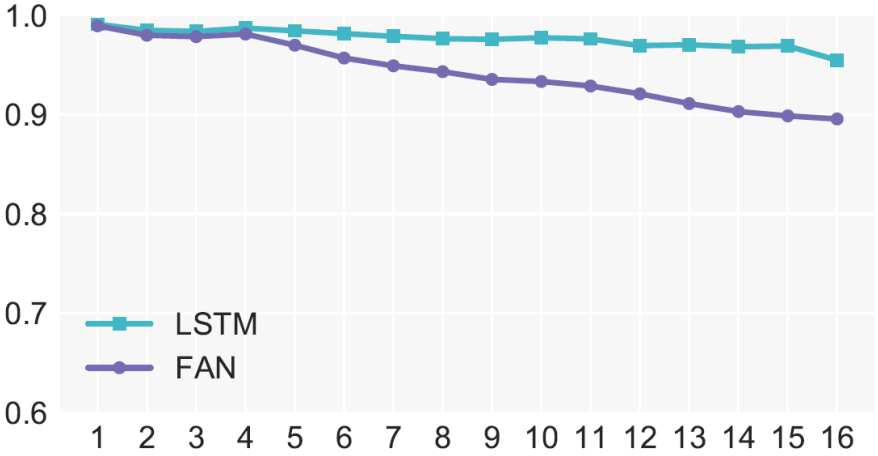
RNNG works better than RNN*

Shows slower rise in agreement error rate as the number of attractors increases (Kuncoro et al., 2018):

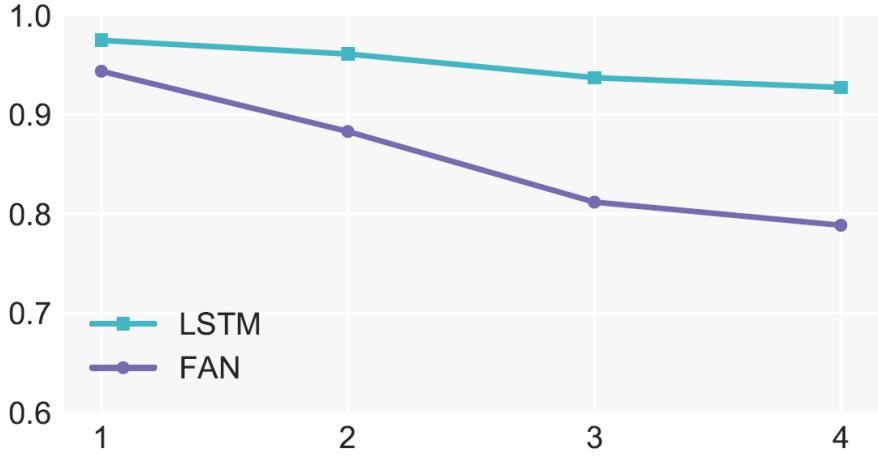


But RNN still better than FAN*

FAN is even worse than RNN at learning the necessary structure for long-distance agreement (Tran et al., 2018):



(a) Language model, breakdown by distance



(b) Language model, breakdown by # attractors

- Despite similar overall LM performance
- Despite access in principle to all context words

*But be wary of these claims

- Performance of any model is a complex function of architecture and training regimes.

- Improvements due to "better" architectures are often later found to come partly/largely from training tricks (Melis et al., 2017; Chen et al., 2018)

*But be wary of these claims

- Performance of any model is a complex function of architecture and training regimes.
- Improvements due to "better" architectures are often later found to come partly/largely from training tricks (Melis et al., 2017; Chen et al., 2018)
- A model with good inductive bias should be easy to train over a wide range of hyperparameters.
  - Yet "researcher tweaking effort" is hard to quantify/control for.

No clear winners yet. (Maybe the community is doing MCMC?)

Summary: syntactic structure

With similar amounts of data, different networks generalize differently.

- Small differences on corpus data can hide big differences in implicit knowledge/generalization.
- Even very well-performing networks don't easily learn hierarchical syntactic structure.
- Humanlike language processing requires humanlike generalization—not just getting the frequent stuff right.
  - Maybe the difference between doing well on similar data, vs across domains and genres (see Koehn and Knowles (2017))

Example 2: subword structure

We want $P(\text{She is plecting}) > P(\text{She is plection})$.

Example 2: subword structure

We want $P(\text{She is plecting}) > P(\text{She is plection})$.

- Note: RNN with words as input don't solve this.
  - Can learn relationships such as king:kings :: queen:queens.

Example 2: subword structure

We want $P(\text{She is plecting}) > P(\text{She is plection})$.

- Note: RNN with words as input don't solve this.
  - Can learn relationships such as king:kings :: queen:queens.
  - i.e., for frequent words can **infer** morphological relationships **using** context.
  - But for rare/unseen words, cannot **infer** likely contexts **using** subword info.

Example 2: subword structure

We want $P(\text{She is plecting}) > P(\text{She is plection})$.

- So, how to solve this?
  - Prof. NLP says: Use morphology!
  - Prof. ML says: Use characters!

Morphology

Regularities in sub-word structure:

- Inflection:
  - Aspect (simple/continuous): walk/walking, plect/plecting
  - Number (singular/plural): kid/kids, book/books
  - Case (nominative/genitive): kid/kid's, book/book's

Morphology

Regularities in sub-word structure:

- Inflection:
  - Aspect (simple/continuous): walk/walking, plect/plecting
  - Number (singular/plural): kid/kids, book/books
  - Case (nominative/genitive): kid/kid's, book/book's
- Derivation: organize/organization, plect/plection
- Compounds: bookseller, whitewash, playhouse

Morphology

Regularities in sub-word structure:

- Inflection:
  - Aspect (simple/continuous): walk/walking, plect/plecting
  - Number (singular/plural): kid/kids, book/books
  - Case (nominative/genitive): kid/kid's, book/book's
- Derivation: organize/organization, plect/plection
- Compounds: bookseller, whitewash, playhouse

Requires annotation: words can be **segmented** into morphemes, and/or **tagged** with inflection information.

Character-based models

Pros:

- No linguistic knowledge or annotation required
- Can model novel (out-of-vocabulary) words and (in principle) morphological relationships.
- Vocabulary size is small: no giant softmax.

Cons:

- Input sequences very long; slower/harder to train.
- Inefficient: model has to learn that words are meaningful units.
- In practice, not always better than word-level models.

Hybrid/in-between models

Two ways to improve on the words vs characters dilemma:

- Hybrid models: simultaneously model both words and characters (Hwang and Sung, 2016; Chung et al., 2016)

- Use in-between units: use sub-word chunks (Mikolov et al., 2012; Sennrich et al., 2016)

These combine the benefits of characters and words, and often work better than either.

But do these models learn morphology?

- Vania et al. (2017; 2018) compared models using
  - characters
  - words
  - unsupervised subword units
  - hand-annotated morphological information
- Vania and Lopez (2017): language modeling.
  - 12 languages, most training sets around 200k words.
- Vania et al. (2018): dependency parsing.
  - 10 languages, 0.6-1.4m words.

Morphology is not redundant

In their results,

- Character-based models beat word-based models.
- Subword-based models sometimes beat character-based models.
- Models using gold standard morphology work best.

But these papers use "small" data sets (< 2m words). Maybe we just need more training data?

Morphology still helps

Matthews et al. (2018): language model includes words, characters, and gold standard morphology.

- Trained on 4-40m words (Finnish, Russian, Turkish).
- Words+Chars+Morph model is better than Words, Chars, or Words+Chars.

Morphology still helps

Matthews et al. (2018): language model includes words, characters, and gold standard morphology.

- Trained on 4-40m words (Finnish, Russian, Turkish).
- Words+Chars+Morph model is better than Words, Chars, or Words+Chars.

Error analysis: in sentences where

- char-RNN without morphology is better: avg word freq is ∼3000.
- model with morphology is better: avg word freq is ∼300.

Even more data??

- Some NNet LMs now trained on billions of words.
- Perhaps they do/will eventually learn morphology.

But...

- 4yo children have heard $\leq$ 50m words.
- If models need 10x that to (maybe) learn morphology, probably the wrong inductive bias...and not always feasible.

Summary: subword structure

- Subword structure is important for rare/novel words.
- But as with syntax, there is evidence that networks do not easily/fully learn it.
- Very large data sets help, but better models (that learn well from less data) would be nicer...

Possible alternatives

If we don't know how to build in the right biases, we can still try to improve NLP by using big data to help with small data.

- Unsupervised pretraining to reduce amount of data needed for supervised tasks
- Multilingual training to improve performance on low-resource languages through transfer/multitask learning

Unsupervised pretraining

Pre-train an LM on $>$ 1b words, leverage it for supervised tasks (parsing, question answering, sentiment analysis, etc).

- Use a pretrained model to output context-sensitive embeddings and input these to a task-specific NNet.
  - ELMo: Peters et al. (2018)

Unsupervised pretraining

Pre-train an LM on $>$ 1b words, leverage it for supervised tasks (parsing, question answering, sentiment analysis, etc).

- Use a pretrained model to output context-sensitive embeddings and input these to a task-specific NNet.
  - ELMo: Peters et al. (2018)
- Fine-tune a pretrained model using task-specific objective.
  - ULMFit: Howard and Ruder (2018); OpenAI-GPT: Radford et al. (2018); BERT: Devlin et al. (2018)

Unsupervised pretraining

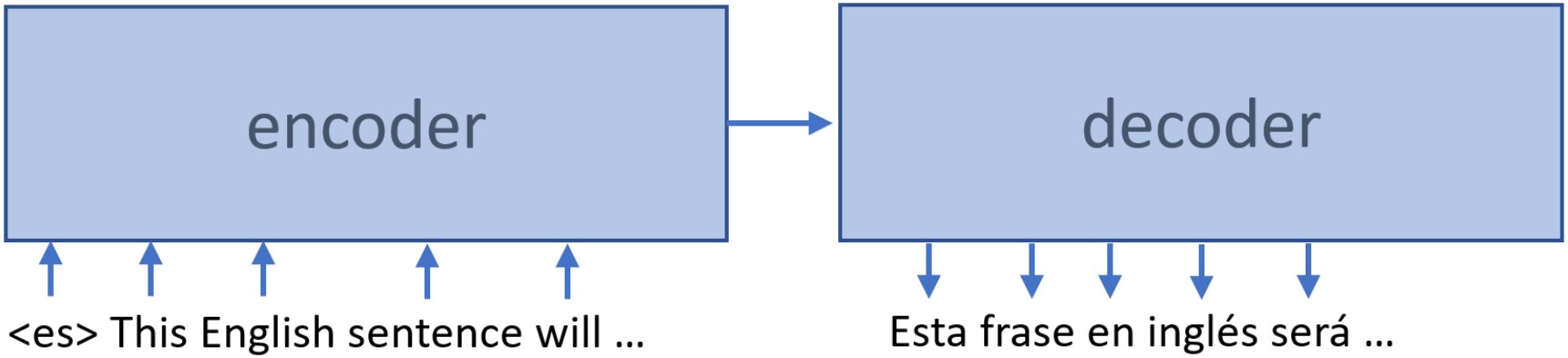
Pre-train an LM on $>$ 1b words, leverage it for supervised tasks (parsing, question answering, sentiment analysis, etc).

- Use a pretrained model to output context-sensitive embeddings and input these to a task-specific NNet.
  - ELMo: Peters et al. (2018)
- Fine-tune a pretrained model using task-specific objective.
  - ULMFit: Howard and Ruder (2018); OpenAI-GPT: Radford et al. (2018); BERT: Devlin et al. (2018)

Very successful in practice, no analysis yet re morphology or implicit syntactic structure learning. And (nearly?) all work is on English.

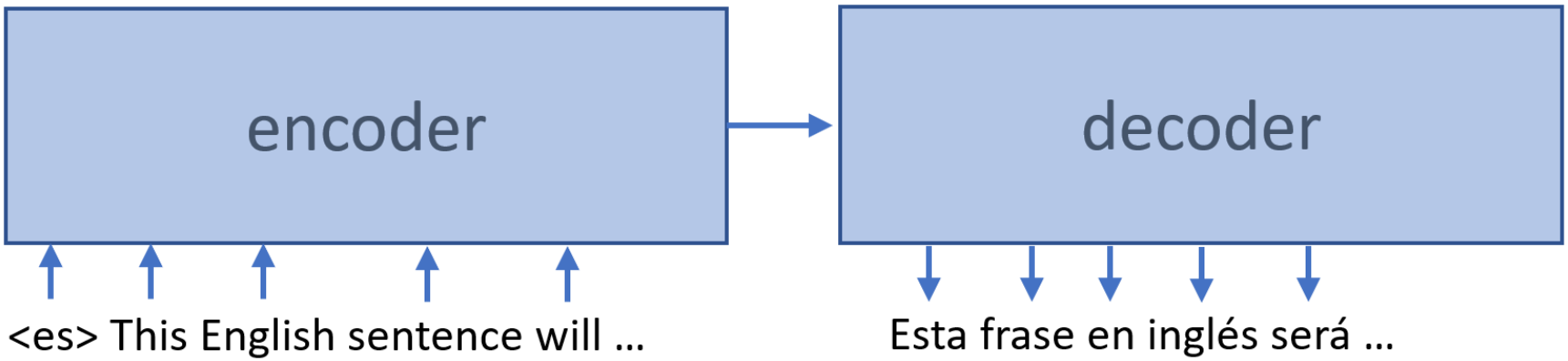
Multilingual training

Ex from machine translation: train a single model to translate between any two languages (Johnson et al., 2017).



Multilingual training

Ex from machine translation: train a single model to translate between any two languages (Johnson et al., 2017).



- Add symbol to input sequence to indicate output language:
  ⟨es⟩ This English sentence will be translated into Spanish
  ⟨fr⟩ And this one into French
  ⟨fr⟩ Wir können auch Deutsch ins Französische übersetzen

Multilingual training

Ex from machine translation: train a single model to translate between any two languages (Johnson et al., 2017).

- Single model trained with n input and m output languages is a bit worse than training separate pairwise models with same data size and model size.
  - But one model instead of $n \times m$.
  - And a big win for pairs with little parallel data.
- Increasing multilingual model size or training time could probably improve it.

Multilingual training

Examples from speech recognition: will present tomorrow!

Remaining issues

- NNets have rapidly advanced the state-of-the-art (good!)
- But test sets and evaluation standards haven't always kept up.
- Simply following previous practice may lead to over-claiming.

How to anger many NLP researchers

Title your paper "Achieving Human Parity on Automatic Chinese to English News Translation" (Hassan et al., 2018).

- Most researchers will know better, but the public won't!
- Headlines such as "Microsoft researchers match human levels in translating news from Chinese to English" (ZDNet, 14/03/18).
- This paper definitely not the only example of hype that is misleading. ML/NLP are now in public view...

A rebuttal

The problem: standard evaluation methods work on isolated sentences.

A rebuttal

The problem: standard evaluation methods work on isolated sentences.

- Hassan et al. (2018) released their data (good!), allowing further analysis.
- Läubli et al. (2018) showed that when sentences were presented in context, human evaluators **did** prefer human translations.
- Follow-up work proposes new test sets for targeted evaluation of discourse phenomena (Bawden et al., 2018)

Advancing evaluation and understanding

More generally, NLP continues to be important for developing new test sets and evaluation methods.

- Great example: "Build it, break it" adversarial shared task[1]
  - Builders: build the best/most robust systems they can.
  - Breakers: come up with linguistically motivated examples that break the systems.

[1] https://bibinlp.umiacs.umd.edu/

So, do we still need NLP?

- The power of NNets to model word similarity and capture frequent patterns is enough to outperform older probabilistic models.
- But those models were based on linguistic insights which NNets do not seem to fully learn.
- Can partially overcome this by massive pretraining or transfer, but unclear yet how far this will go.
- Until then, domain knowledge about language can help us to:
  - Build models with task-specific architectures or input encoding.
  - Design better evaluations to test hard cases and avoid over-claiming.

References

Bawden, R., Sennrich, R., Birch, A., and Haddow, B. (2018). Evaluating discourse phenomena in neural machine translation. In *16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Chen, M. X., Firat, O., Bapna, A., Johnson, M., Macherey, W., Foster, G., Jones, L., Parmar, N., Schuster, M., Chen, Z., et al. (2018). The best of both worlds: Combining recent advances in neural machine translation. *arXiv preprint arXiv:1804.09849*.

Chung, J., Ahn, S., and Bengio, Y. (2016). Hierarchical multiscale recurrent neural networks. *arXiv preprint arXiv:1609.01704*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dyer, C., Kuncoro, A., Ballesteros, M., and Smith, N. A. (2016). Recurrent neural network grammars. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 199–209.

Goldwater, S., Griffiths, T. L., and Johnson, M. (2006). Interpolating between types and tokens by estimating power-law generators. In *Advances in Neural Information Processing Systems 18*, pages 459–466, Cambridge, MA. MIT Press.

Gulordava, K., Bojanowski, P., Grave, E., Linzen, T., and Baroni, M. (2018). Colorless green recurrent networks dream hierarchically. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 1195–1205.

Hassan, H., Aue, A., Chen, C., Chowdhary, V., Clark, J., Federmann, C., Huang, X., Junczys-Dowmunt, M., Lewis, W., Li, M., et al. (2018). Achieving human parity on automatic chinese to english news translation. *arXiv preprint arXiv:1803.05567*.

Howard, J. and Ruder, S. (2018). Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 328–339.

Hwang, K. and Sung, W. (2016). Character-level incremental speech recognition with recurrent neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pages 5335–5339. IEEE.

Johnson, M., Schuster, M., Le, Q. V., Krikun, M., Wu, Y., Chen, Z., Thorat, N., Viégas, F., Wattenberg, M., Corrado, G., et al. (2017). Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Koehn, P. and Knowles, R. (2017). Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39.

Kuncoro, A., Dyer, C., Hale, J., Yogatama, D., Clark, S., and Blunsom, P. (2018). Lstms can learn syntax-sensitive dependencies well, but modeling structure makes them better. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1426–1436.

Läubli, S., Sennrich, R., and Volk, M. (2018). Has machine translation achieved human parity? a case for document-level evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4791–4796.

Linzen, T., Dupoux, E., and Goldberg, Y. (2016). Assessing the ability of lstms to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.

MacKay, D. and Bauman Peto, L. (1994). A hierarchical Dirichlet language model. *Natural Language Engineering*, 1(1).

Marvin, R. and Linzen, T. (2018). Targeted syntactic evaluation of language models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202.

Matthews, A., Neubig, G., and Dyer, C. (2018). Using morphological knowledge in open-vocabulary neural language models. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 1435–1445.

Melis, G., Dyer, C., and Blunsom, P. (2017). On the state of the art of evaluation in neural language models. *arXiv preprint arXiv:1707.05589*.

Mikolov, T., Sutskever, I., Deoras, A., Le, H.-S., and Kombrink, S. (2012). Subword language modeling with neural networks.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 2227–2237.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). Improving language understanding by generative pre-training. Technical report, OpenAI.

Sennrich, R., Haddow, B., and Birch, A. (2016). Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1715–1725.

Teh, Y. W. (2006). A hierarchical Bayesian language model based on Pitman-Yor processes. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 985–992, Syndney, Australia.

Tran, K., Bisazza, A., and Monz, C. (2018). The importance of being recurrent for modeling hierarchical structure. *arXiv preprint arXiv:1803.03585*.

Vania, C., Grivas, A., and Lopez, A. (2018). What do character-level models learn about morphology? the case of dependency parsing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2573–2583.

Vania, C. and Lopez, A. (2017). From characters to words to in between: Do we capture morphology? In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 2016–2027.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Towards more universal language technology: unsupervised learning from speech

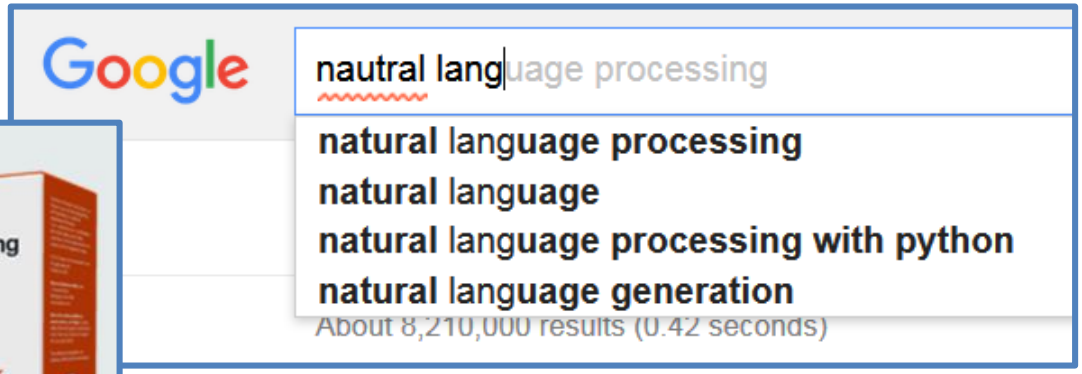
Sharon Goldwater

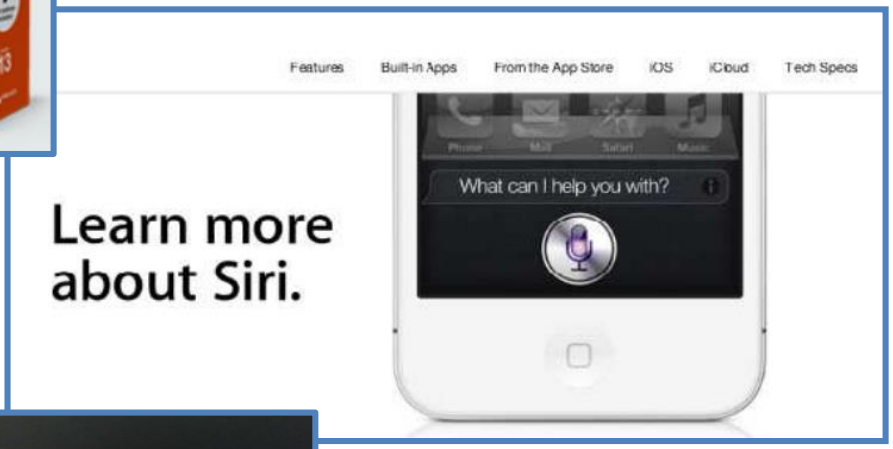
ilcc | Institute for Language, Cognition and Computation

Edinburgh University of Edinburgh Natural Language Processing NLP

THE UNIVERSITY of EDINBURGH informatics

Joint work with Sameer Bansal, Micha Elsner, Enno Hermann, Herman Kamper, Karen Livescu, Adam Lopez, Aren Jansen, Daniel Renshaw



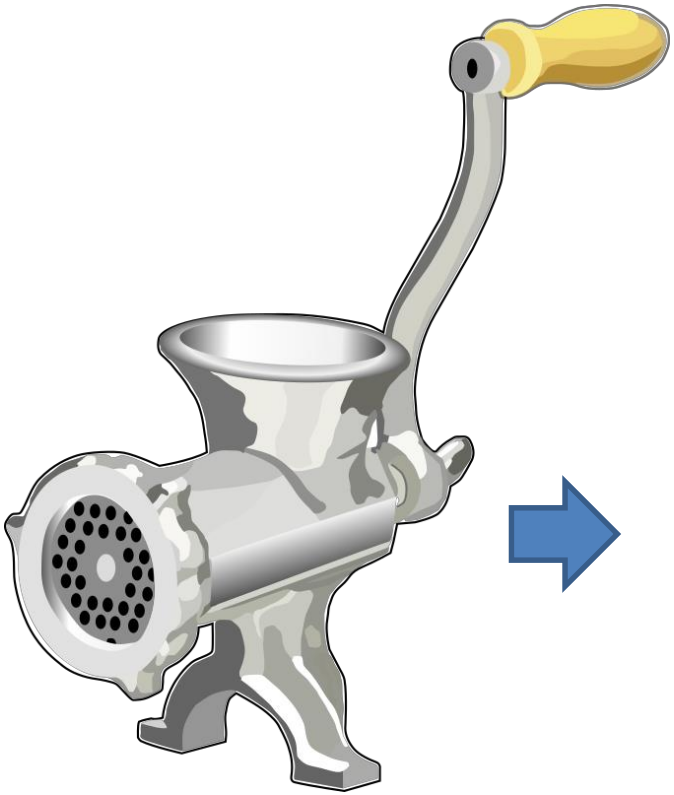


Sources: google.co.uk, nuance.co.uk, apple.com, www.amazon.co.uk, www.phonearena.com, cnet.com

Supervised language technology

Training:

Labelled examples

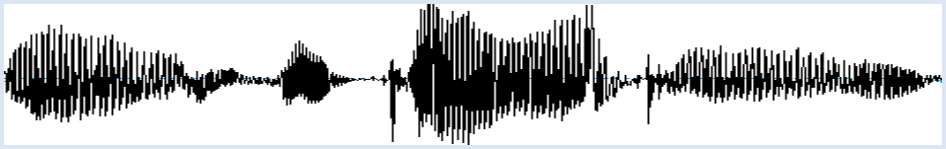


(Machine learning algorithm)

Prediction function

Labelled data is hard to obtain

- Humans must be paid:



We need more cycle paths on campus.

- Often need special training:

I met the woman in the cafe

- For many languages, resources are unavailable.

Even unlabelled data may be scarce

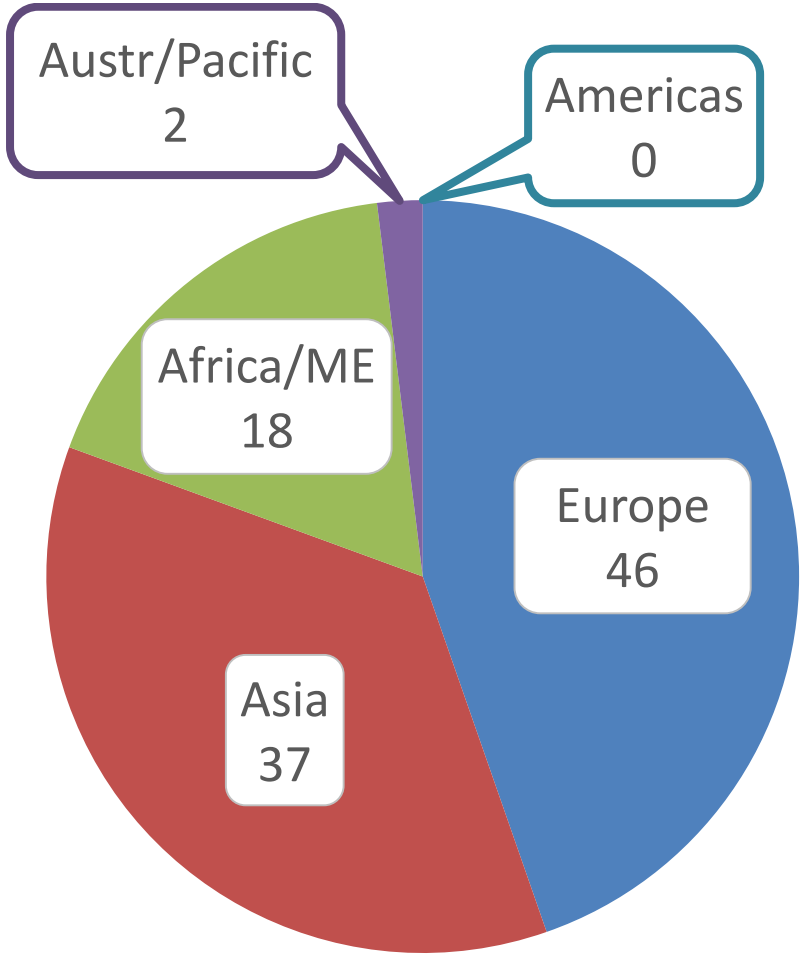
- Of the estimated 7000 language in the world,
  - 141 have > 10k Wikipedia articles (English: 5.8m)
  - Many lack a (standard or any) written form

Result: unequal access

- Google translate includes 103 languages.

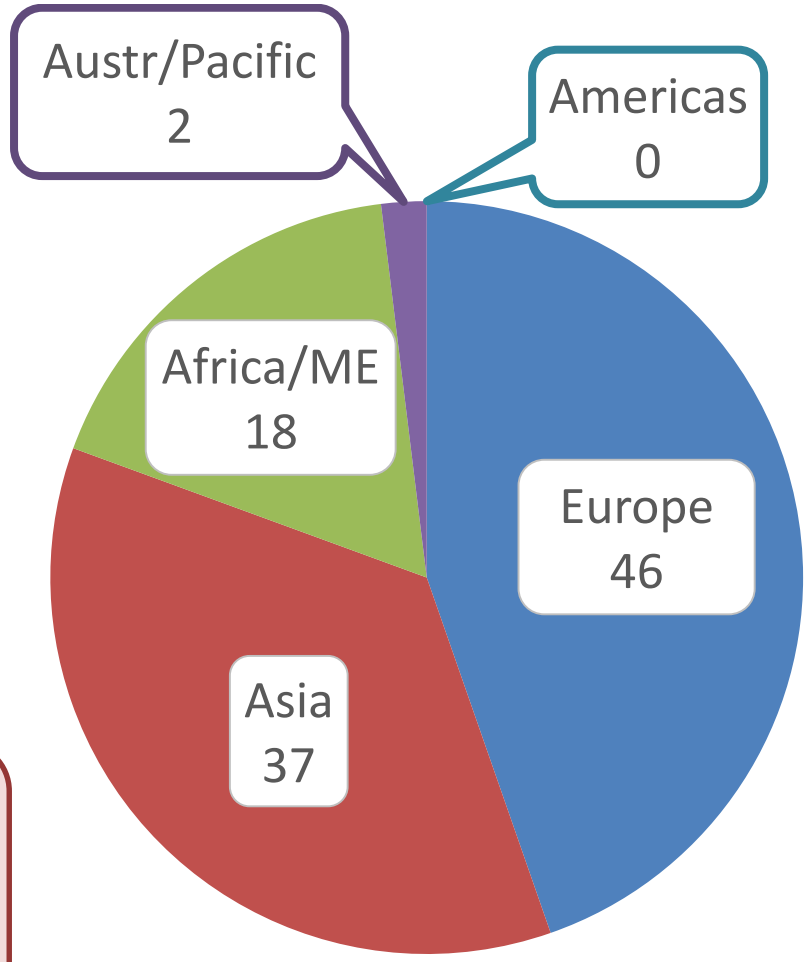
Result: unequal access

- Google translate includes 103 languages.



Result: unequal access

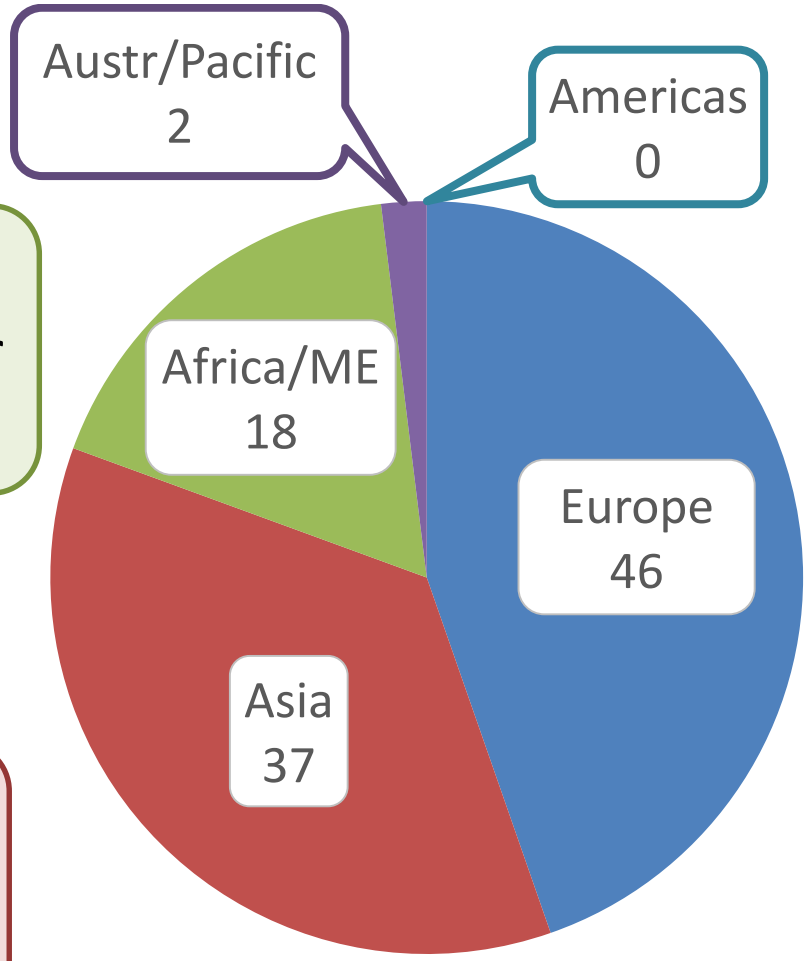
- Google translate includes 103 languages.



Includes only 9 of India's 22 official languages

Result: unequal access

- Google translate includes 103 languages.

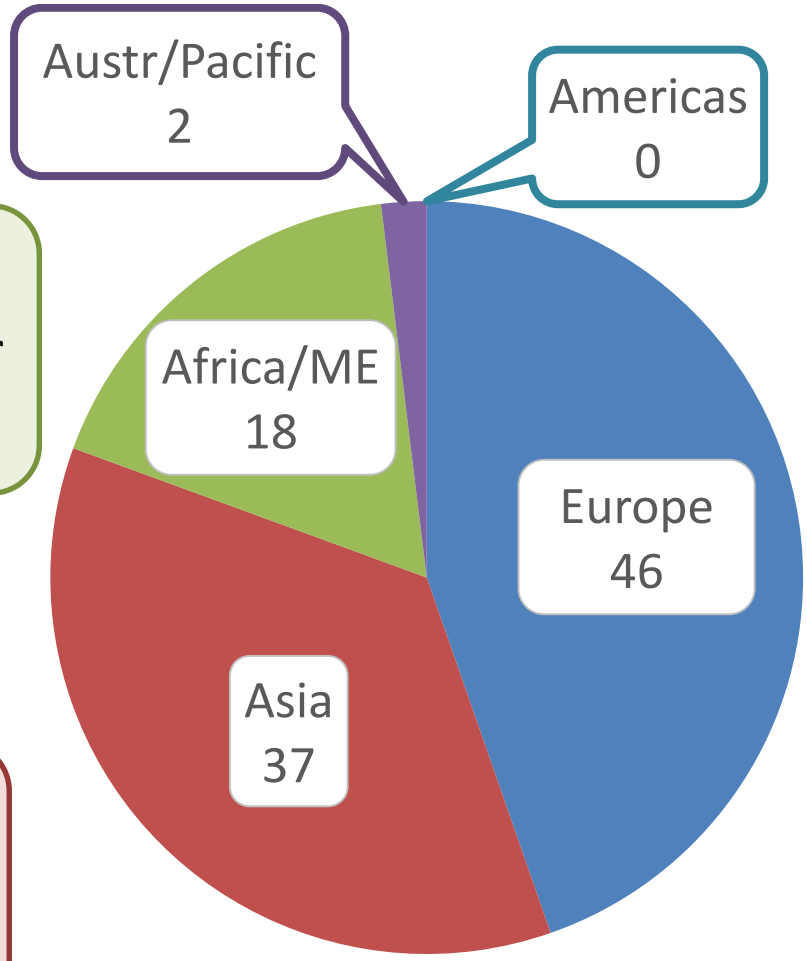


Missing 5 African languages with over 10m speakers each

Includes only 9 of India's 22 official languages

Result: unequal access

- Google translate includes 103 languages.



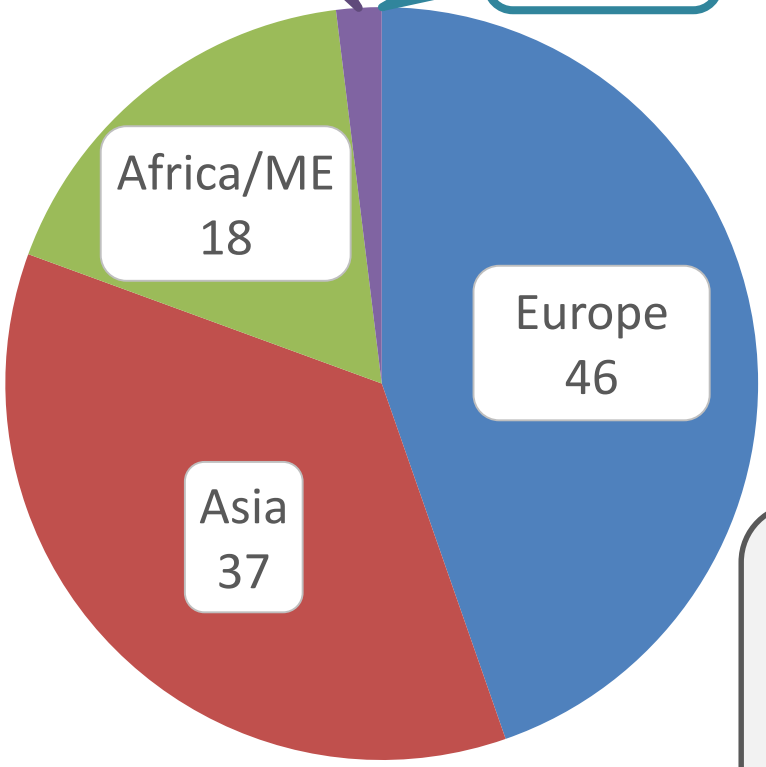
Missing 5 African languages with over 10m speakers each

From Quechua (6m speakers) to endangered languages of NA.

Includes only 9 of India's 22 official languages

Result: unequal access

- Google translate includes 103 languages.



Missing 5 African languages with over 10m speakers each

From Quechua (6m speakers) to endangered languages of NA.

Includes only 9 of India's 22 official languages

Altogether, est. 7000 languages, many minority or endangered.

Two possible directions

- Minimally supervised:
  - Small amount of labelled data, maybe also unlabelled data
- Unsupervised/distantly supervised
  - No labels for task of interest
  - May have other information/labels (translations, images, other context)
- Need a variety of approaches

Outline

1. Background
2. Unsupervised monolingual models
3. Using information from other languages

What *is* unsupervised ASR?

What *is* unsupervised ASR?

- Mostly: unsupervised term discovery (UTD)
  - High precision, low recall. Heuristic.

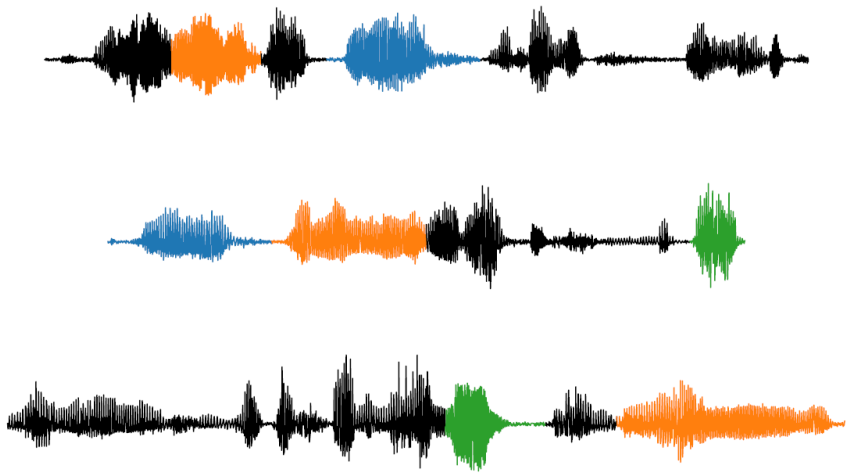


Figure: Herman Kamper

Park & Glass (2008), Jansen and van Durme (2011), Zhang et al. (2012)

What *is* unsupervised ASR?

- Mostly: unsupervised term discovery (UTD)
  - High precision, low recall. Heuristic.
  - Query-by-example, doc clustering/navigation.

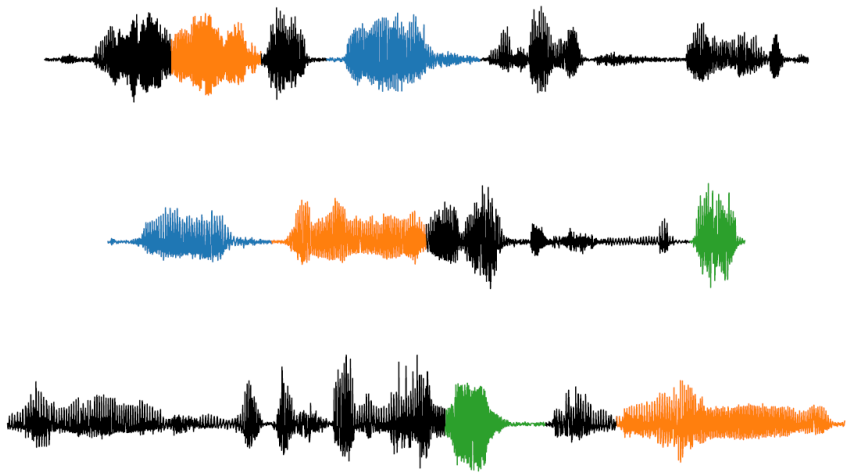


Figure: Herman Kamper

Park & Glass (2008), Jansen and van Durme (2011), Zhang et al. (2012), Siu et al. (2013), Dredze et al. (2010), Levin et al. (2015)

What *is* unsupervised ASR?

- Mostly: unsupervised term discovery (UTD)
- Here: full-coverage segmentation/clustering
  - Higher recall, lower precision. Probabilistic.

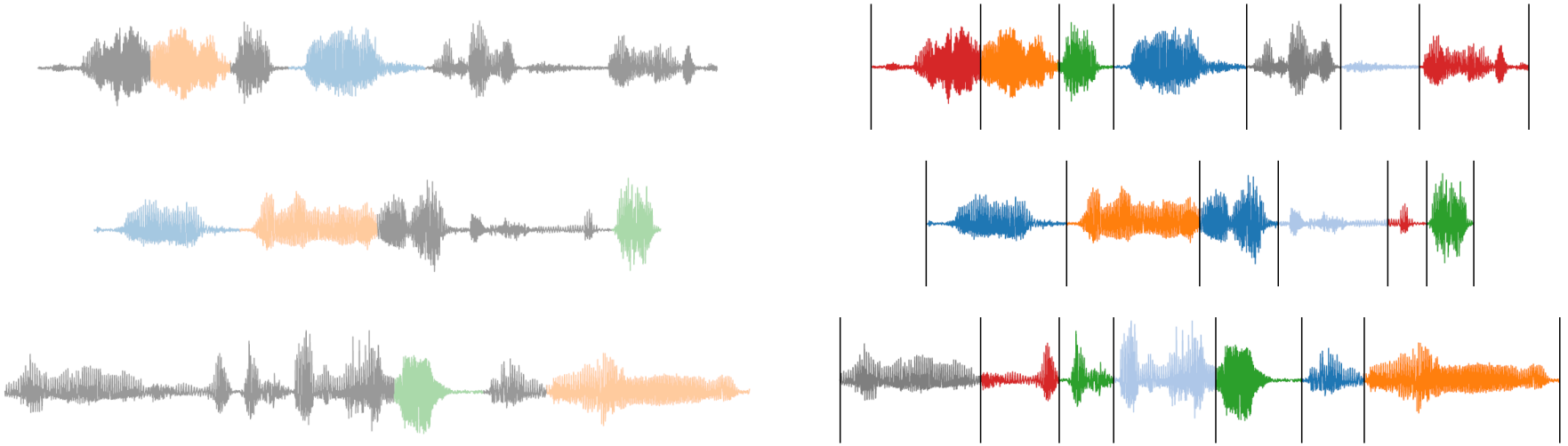
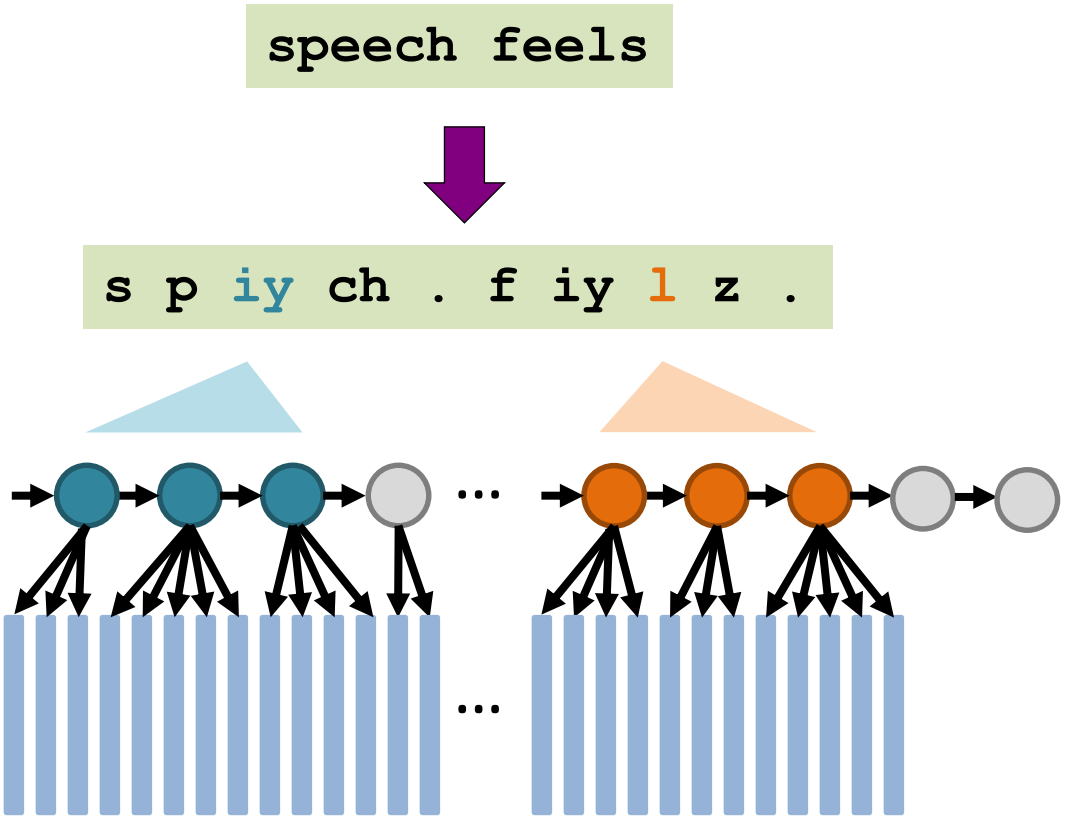


Figure: Herman Kamper

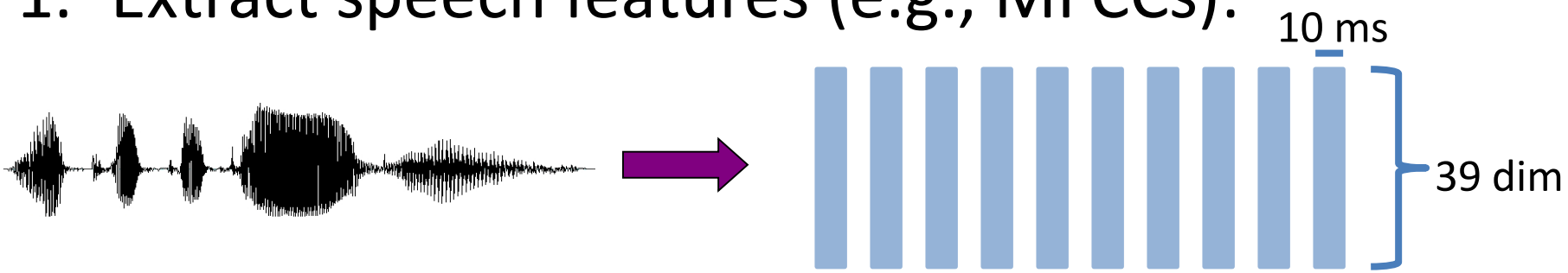
Straw man approach

- Use a model like traditional supervised ASR:

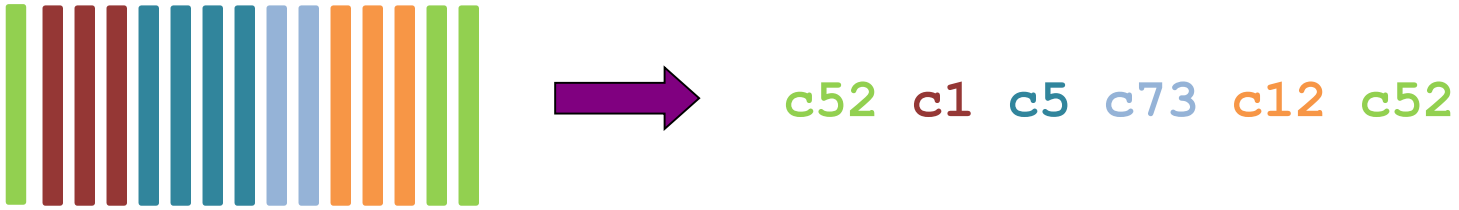


Straw man approach

1. Extract speech features (e.g., MFCCs):



2. Cluster feature vectors:



3. Group the symbol sequence into “words”*:

c52c1c5 c73 c12c52

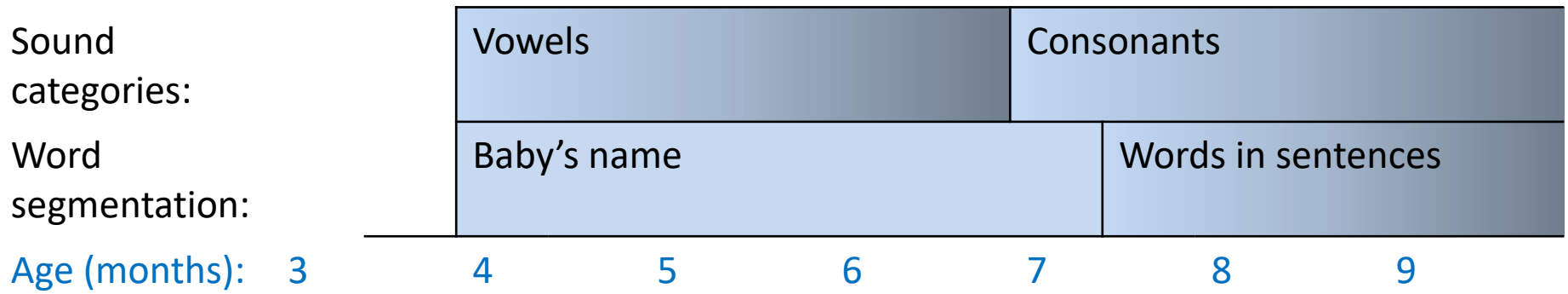
*Using methods as in Goldwater, Griffiths, and Johnson (2009), or many followups.

Why not bottom-up clustering?

- Speech is just too variable!
  - Speaker
  - Gender
  - Channel noise
  - Phonetic context

Infant speech perception

- Phonetic and word learning occur in parallel:



- *Joint learning* of words and subwords?

Figure adapted with permission from: Dupoux, E. (2016). "Cognitive Science in the era of Artificial Intelligence: A roadmap for reverse-engineering the infant language-learner". arXiv:1607.08723

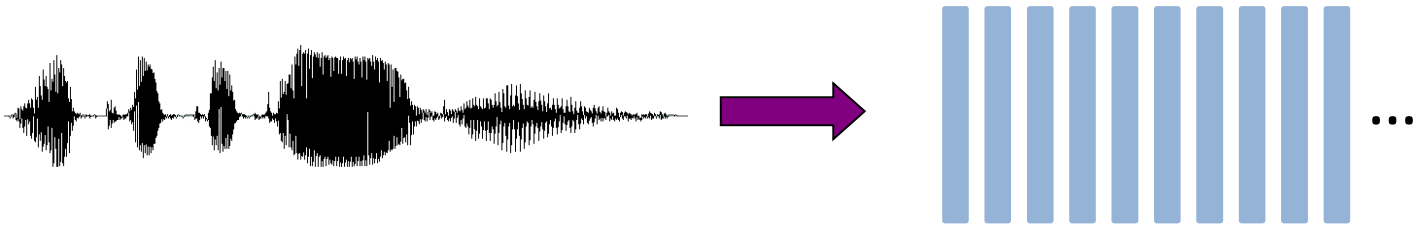
Outline

1. Background
2. Unsupervised monolingual models
   - Using top-down information to improve subword representations.
   - Combining top-down and bottom-up information for segmentation and clustering.
3. Using information from other languages

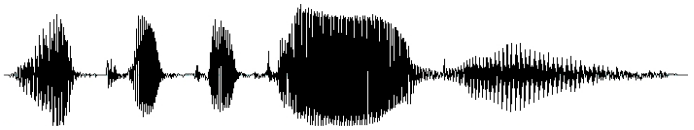
H. Kamper, M. Elsner, A. Jansen, and S. Goldwater. 2015. Unsupervised neural network based feature extraction using weak top-down constraints. *Proceedings of ASRU*.

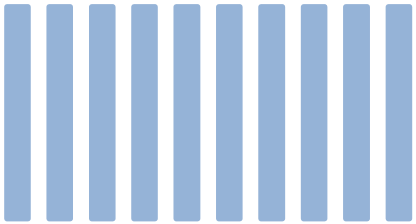
D. Renshaw, H. Kamper, A. Jansen, and S. Goldwater. 2015. A Comparison of Neural Network Methods for Unsupervised Representation Learning on the Zero Resource Speech Challenge. *Proceedings of Interspeech*.

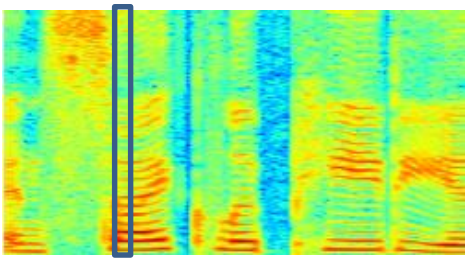
Speech features



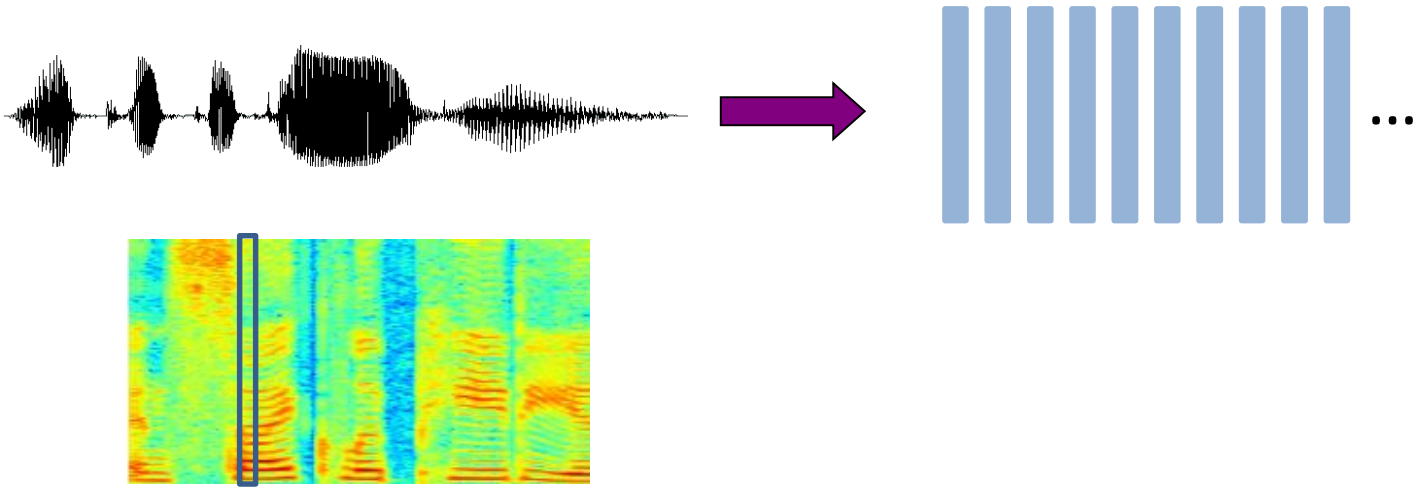
Speech features





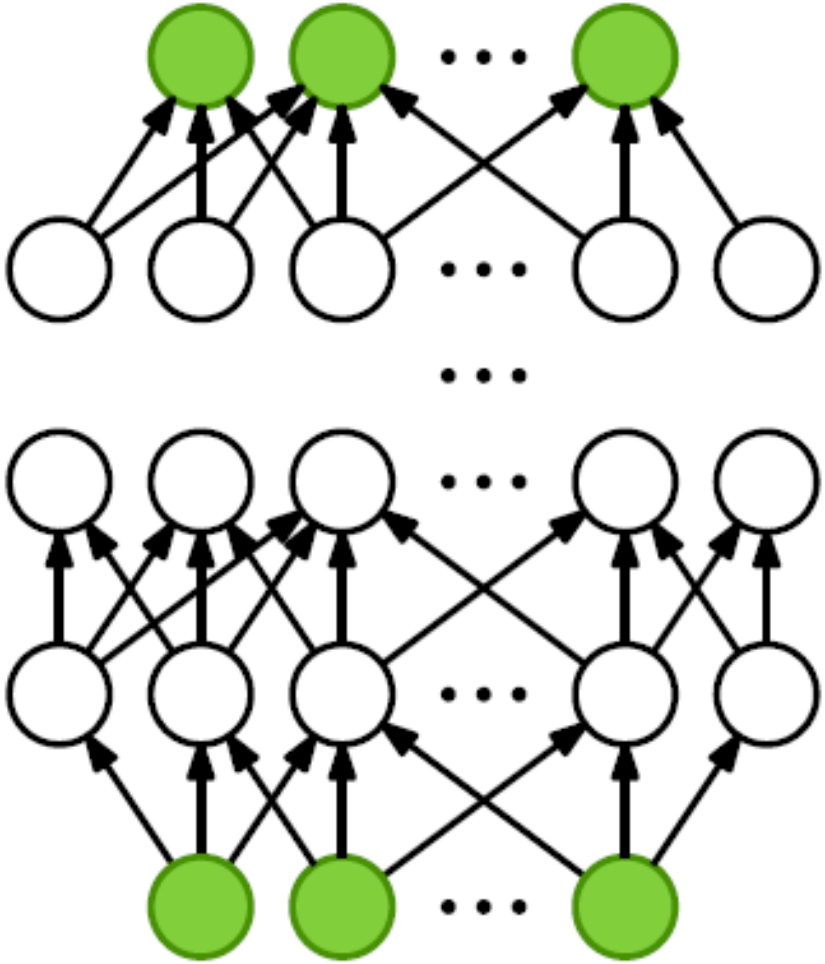


Speech features



- Traditionally, use signal processing.
- Can we use machine learning to improve?
  - Abstract away from “unimportant” variation
  - Capture content-critical variation

Autoencoder

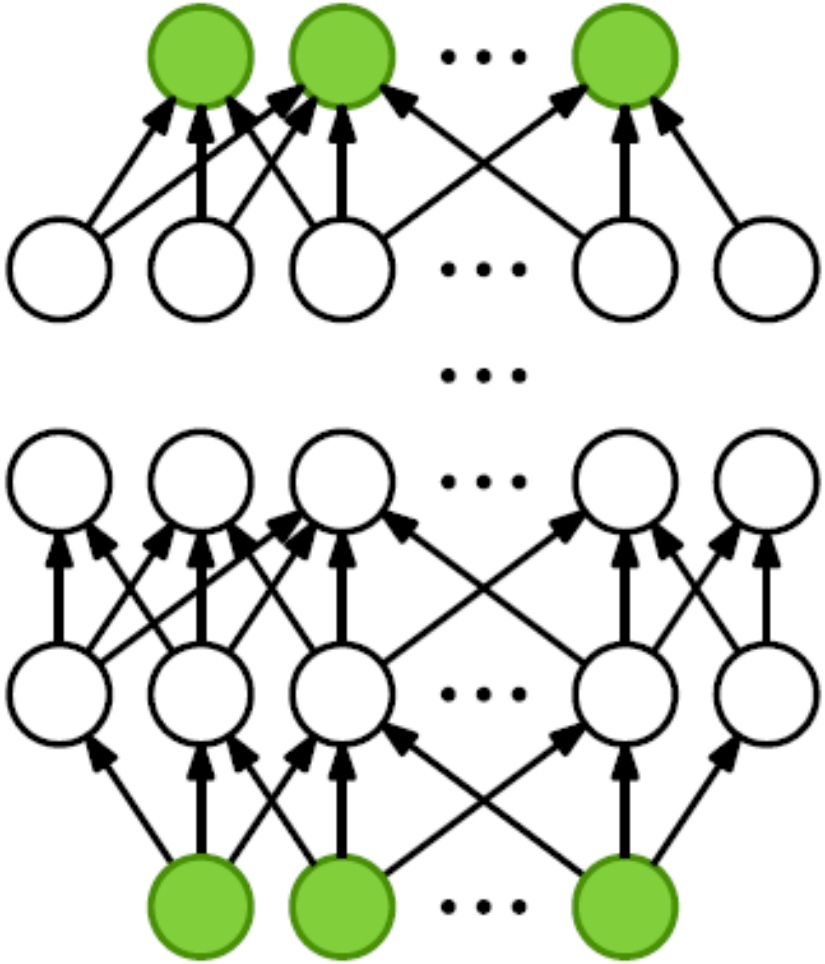


Output vector

Hidden layers

Input vector

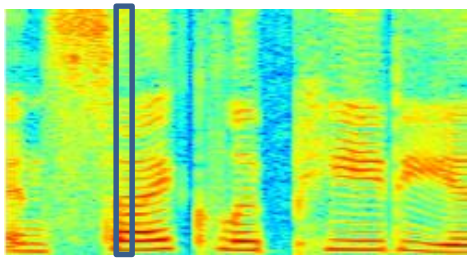
Autoencoder



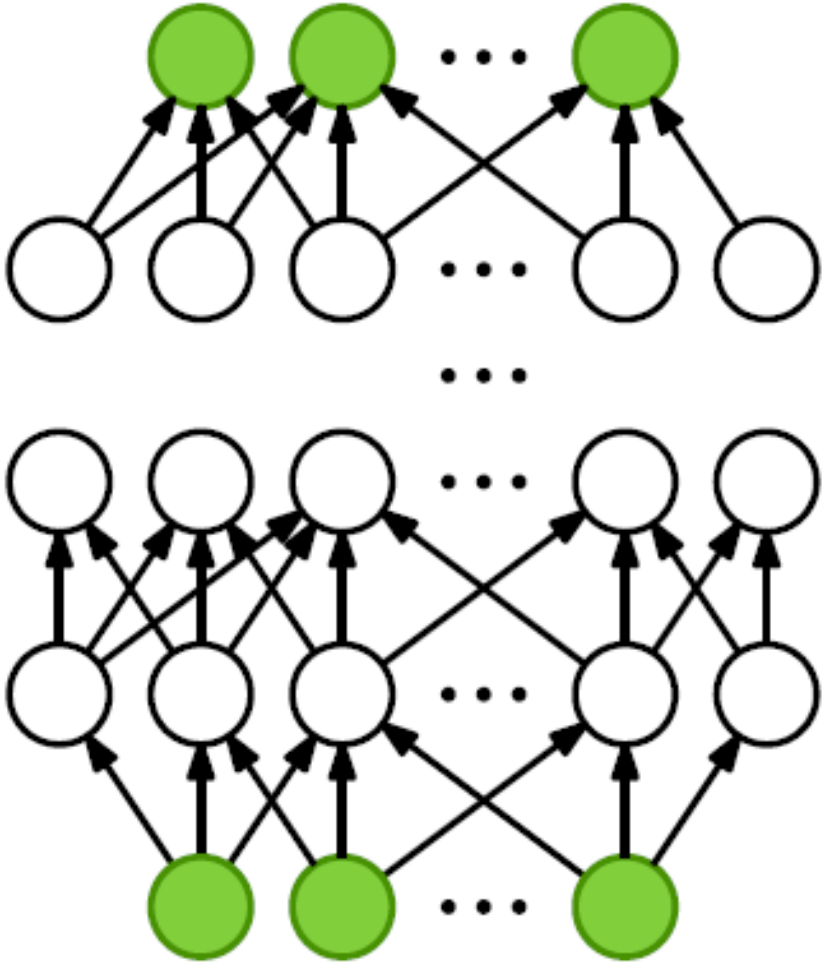
Output vector

Hidden layers

Input vector



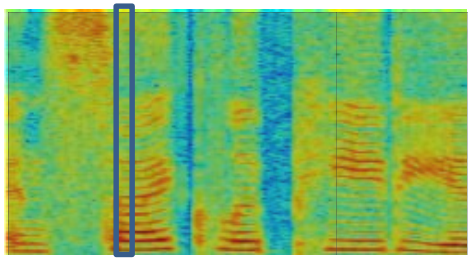
Denoising autoencoder



Output vector

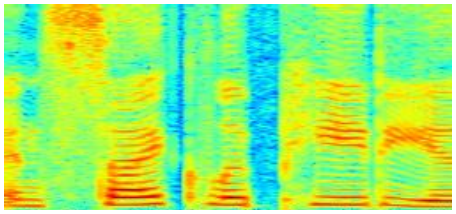
Hidden layers

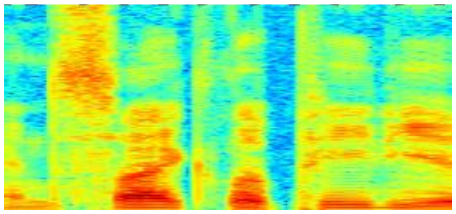
Input vector +
random noise

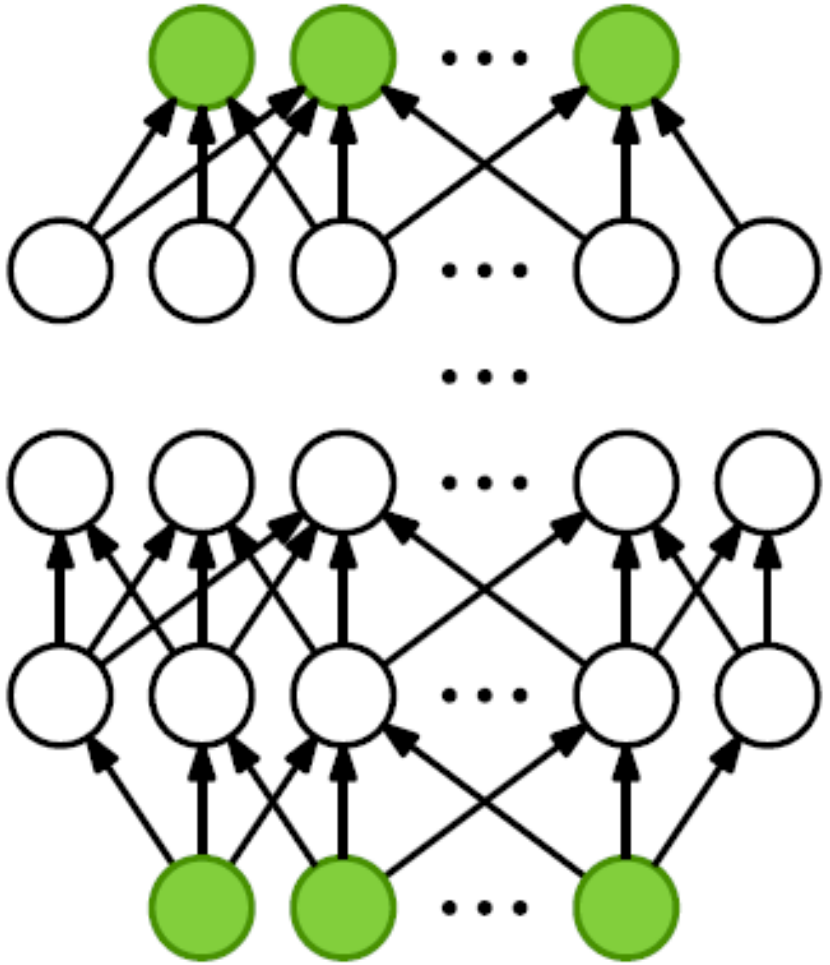


Correspondence autoencoder

Two examples of a word

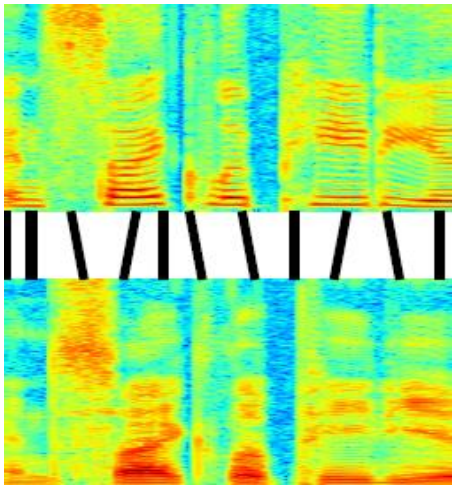






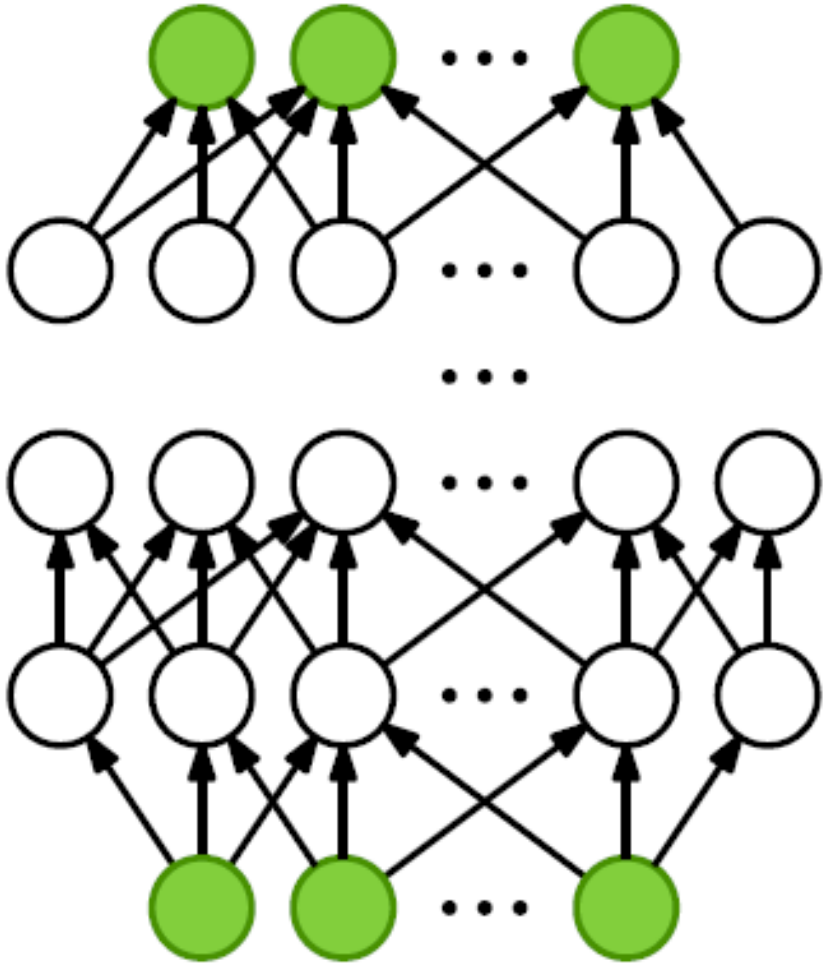
Correspondence autoencoder

Two examples of a word



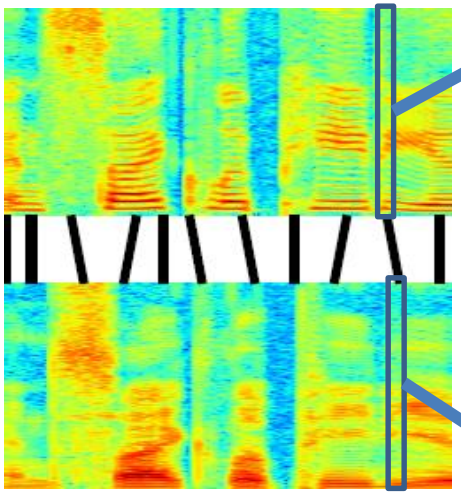
Align frames

(use Dynamic Time Warping)

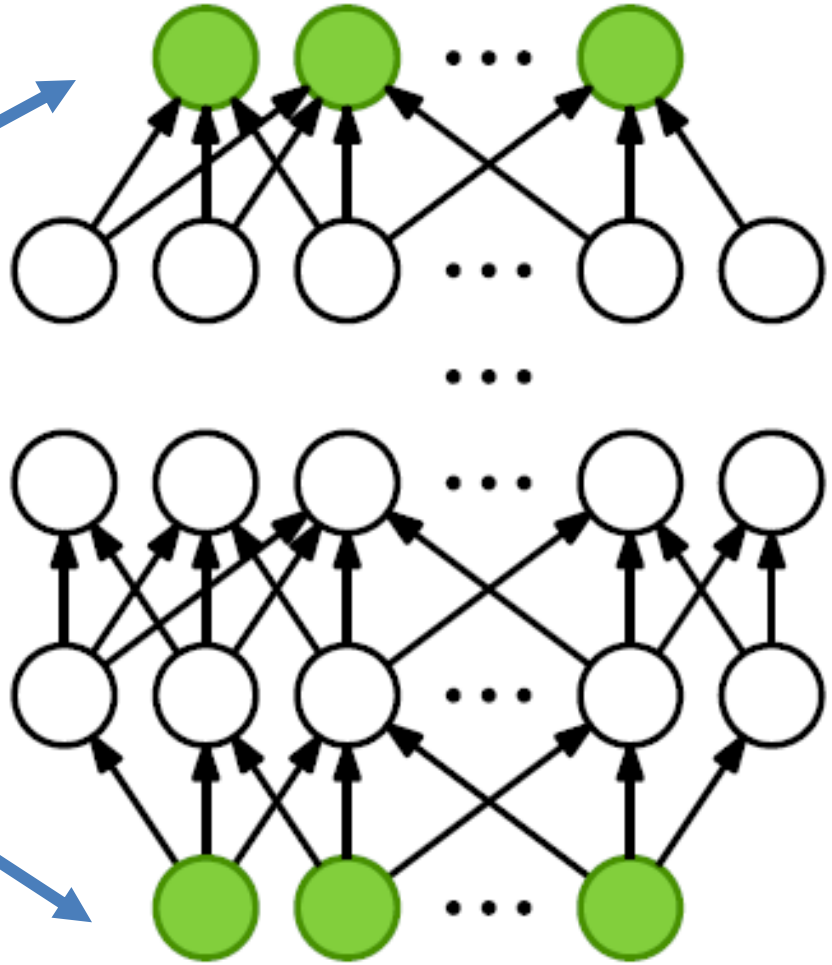


Correspondence autoencoder

Two examples of a word

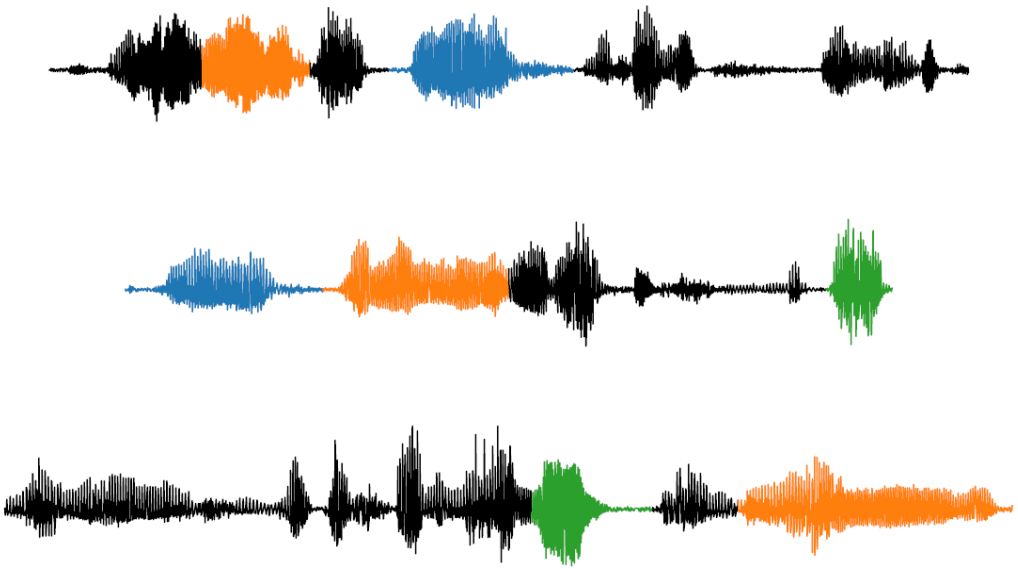


Align frames (use DTW)



Where do word pairs come from?

- Use unsupervised term detection!



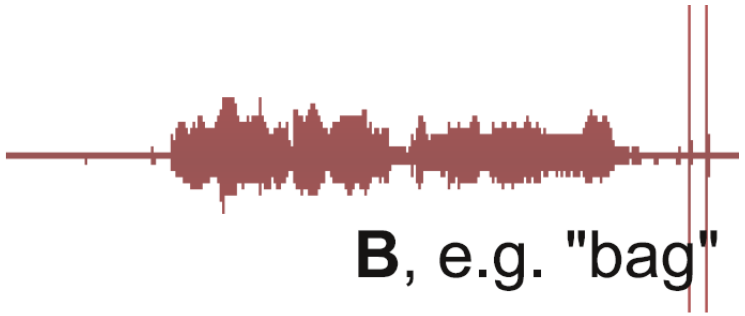
- Noisy top-down signal: many incorrect pairs.

Evaluation: triphone ABX task

- Triphone examples differ by a single phone:

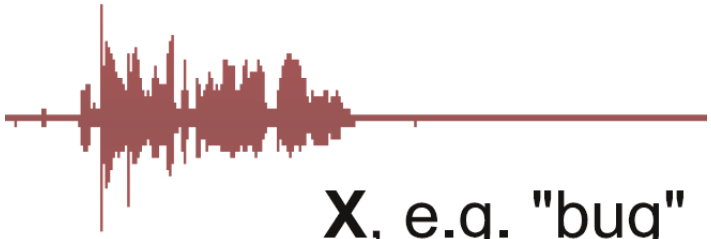
Speaker 1:

A, e.g. "bug"



B, e.g. "bag"

Speaker 2:



X, e.g. "bug"

Is X more similar to A or B?

Schatz et al. (2013), Versteegh et al. (2015).

Results

Model	ABX error rate	
	English (tuning)	Xitsonga (no tuning)
Baseline (MFCCs)	28.1	33.8
Autoencoder (AE)	28.6	29.5
Denoising AE (DAE)	25.3	25.9
Correspondence AE (CAE)	**21.1**	**19.3**

- *Cross-speaker* results improve more than *within-speaker*.

Summary

- Noisy top-down information (from UTD word pairs) can help learn better subword features.
- Results on intrinsic measure: discriminability
- Can these features also improve word segmentation and clustering?

Outline

1. Background
2. Unsupervised monolingual models
   - Using top-down information to improve subword representations.
   - Combining top-down and bottom-up information for segmentation and clustering.
3. Using information from other languages

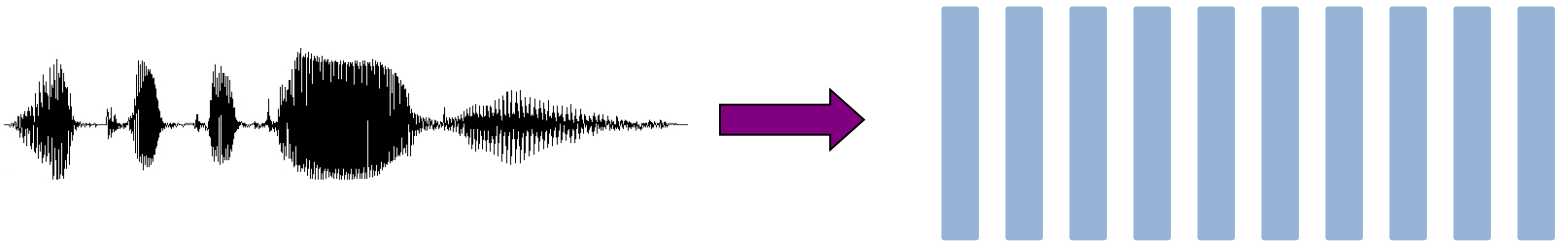
H. Kamper, A. Jansen and S. Goldwater. Unsupervised small-vocabulary speech recognition using a segmental Bayesian model. *Proceedings of Interspeech*, 2015.

H. Kamper, A. Jansen and S. Goldwater. Unsupervised Word Segmentation and Lexicon Discovery using Acoustic Word Embeddings. *IEEE TASLP* 24 (4), pp. 669–679. 2016.

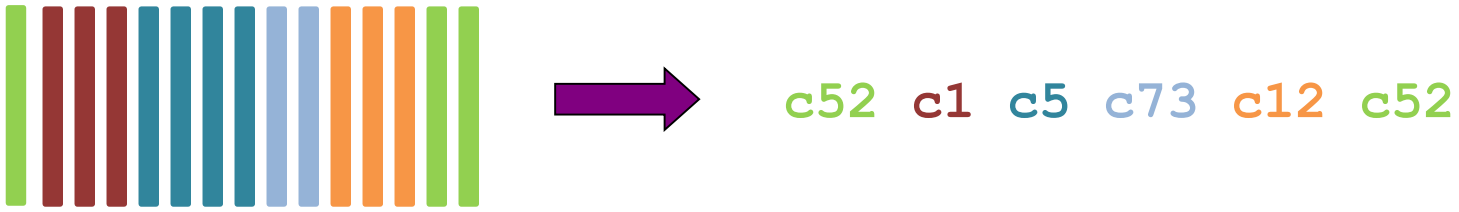
H. Kamper, A. Jansen and S. Goldwater. A segmental framework for fully-unsupervised large-vocabulary speech recognition. *Computer Speech & Language* 46, pp. 154–174. 2017.

Reminder: straw man approach

1. Extract speech features:



2. Cluster feature vectors:



3. Group the symbol sequence into “words”:

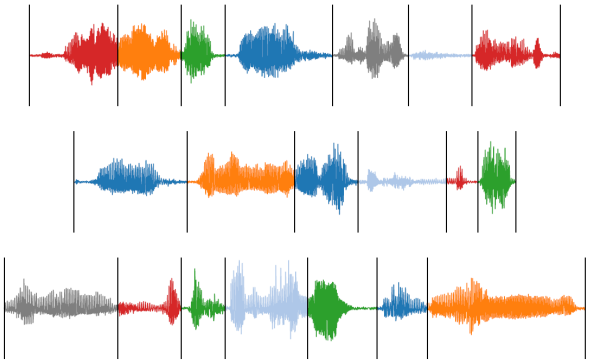
c52c1c5 c73 c12c52

Modelling contextual variability

- More complex phone-based models use noisy channel or relax independence assumptions.
  - Elsner, Goldwater, Feldman, and Wood (2013)
  - Lee, O'Donnell, and Glass (2015)
- But inference very complex, still not demonstrated on multi-speaker audio.

Our approach

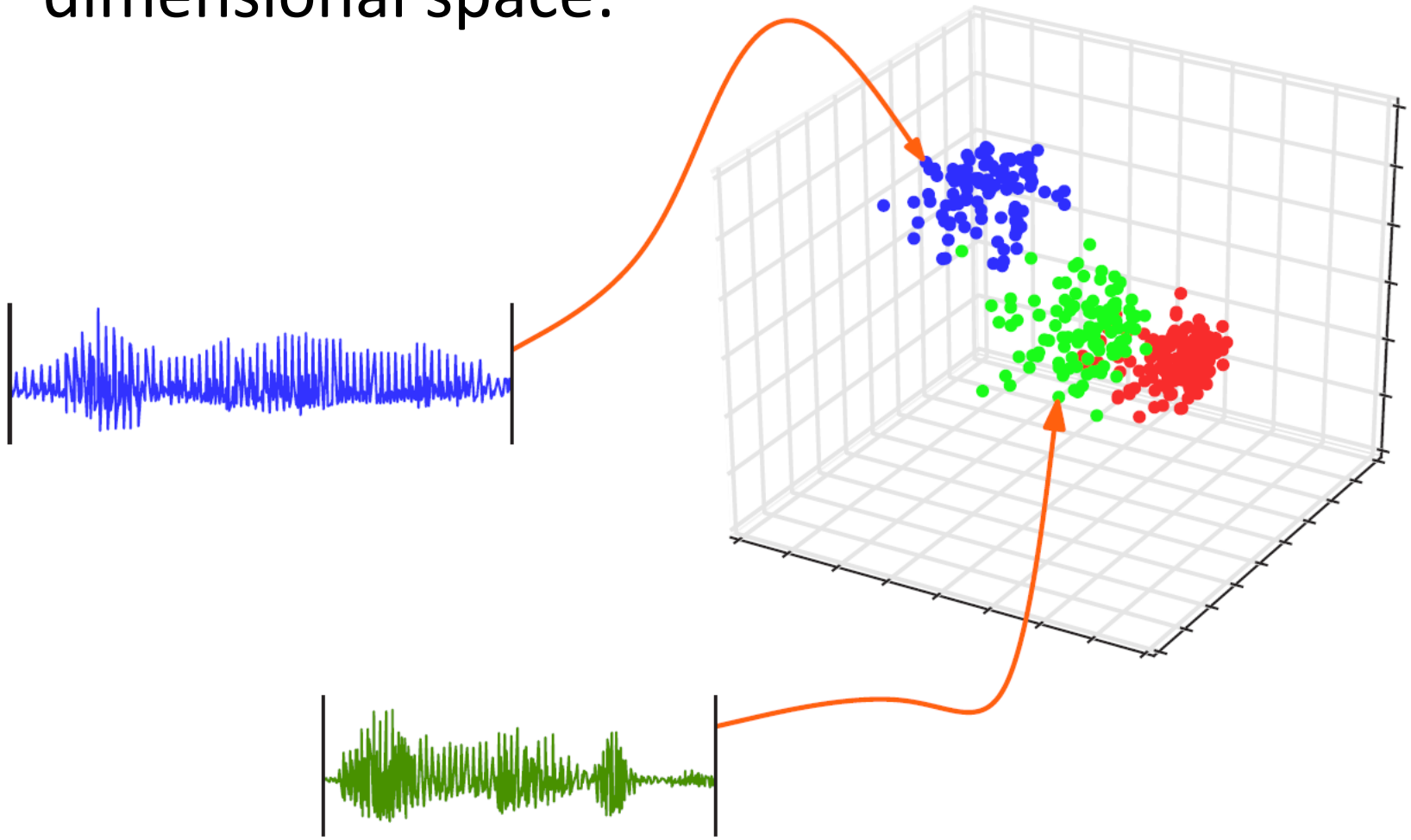
- Model entire words holistically (no phones)
  - Probabilistic model and standard algorithms
- Jointly segment and cluster:



- Main results:
  - Good multi-speaker clusters on small vocab
  - Outperforms competing systems on large vocab single-speaker
  - First system tested on large vocab multi-speaker

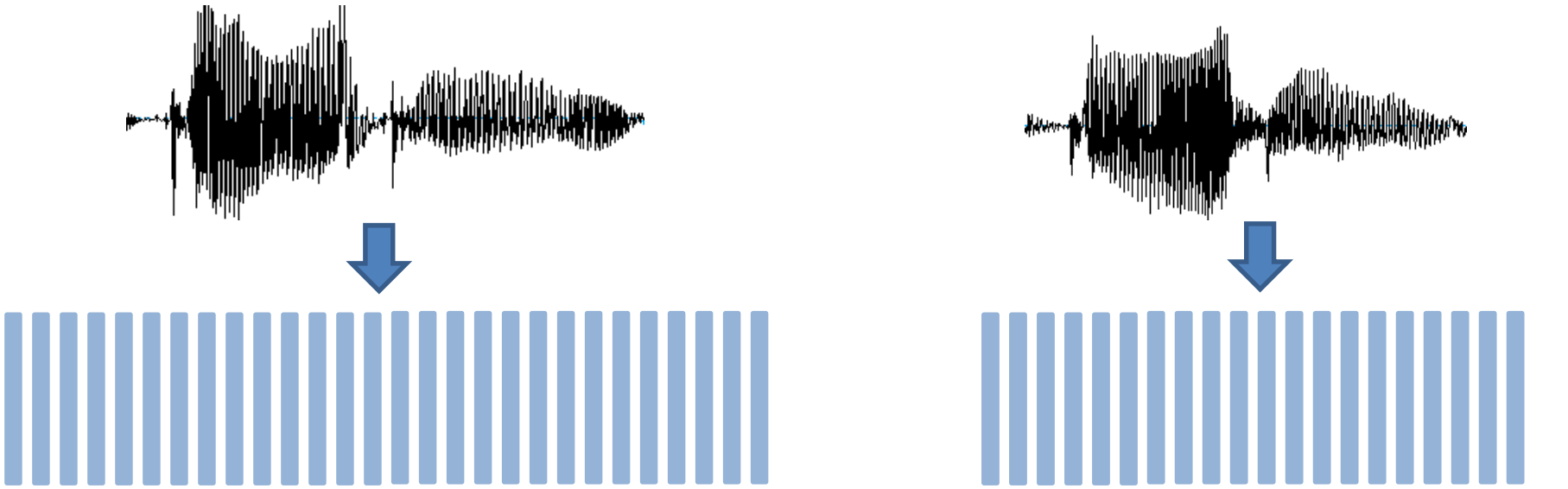
Acoustic embeddings

- Represent variable length segments in fixed-dimensional space:



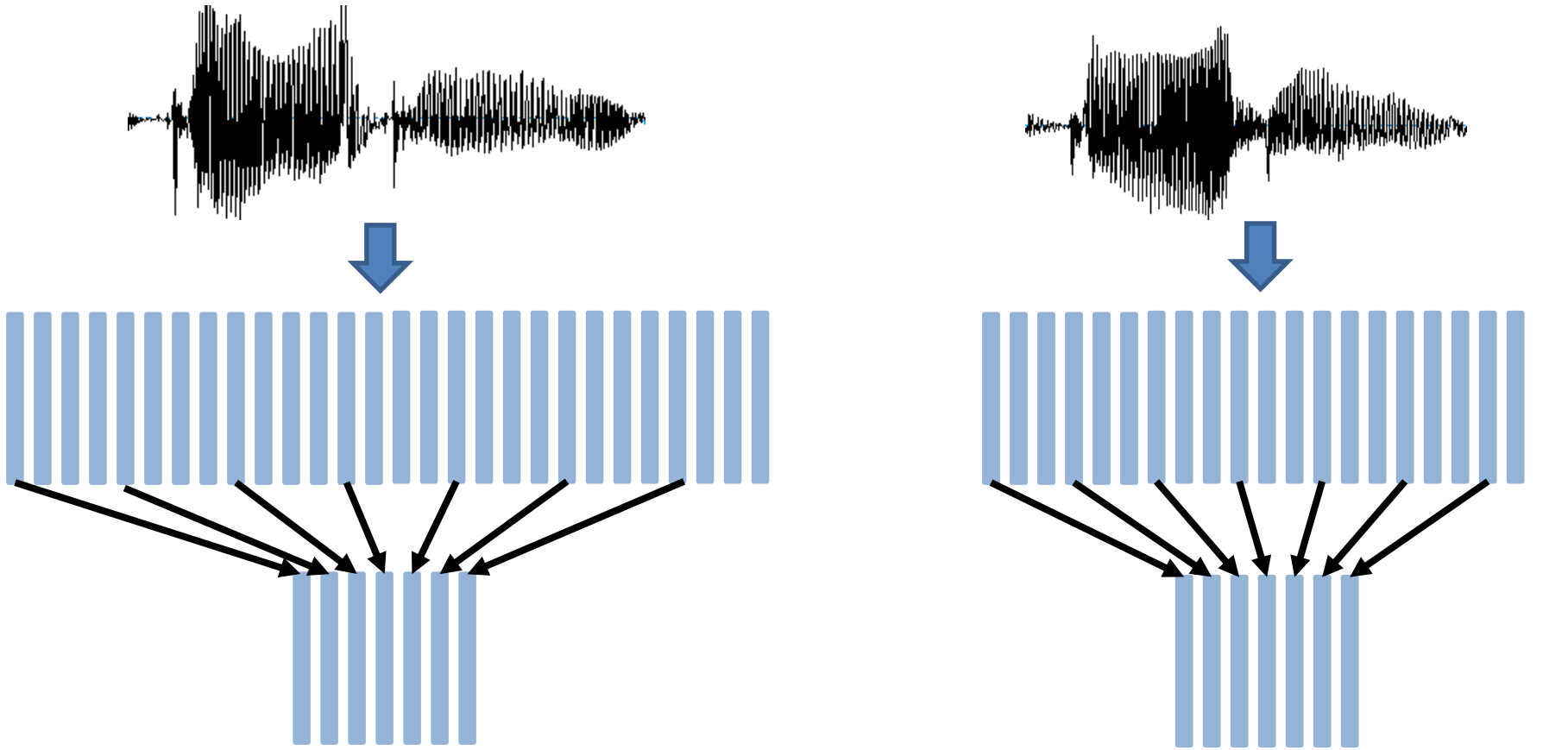
Acoustic embeddings

- Tried different methods, used the simplest: downsampling.



Acoustic embeddings

- Tried different methods, used the simplest: downsampling.



Acoustic model: GMM

- Data points $\mathbf{x}_i$ are acoustic embeddings.
- Each lexical item k is a multivariate Gaussian.
  - Given lexical item:

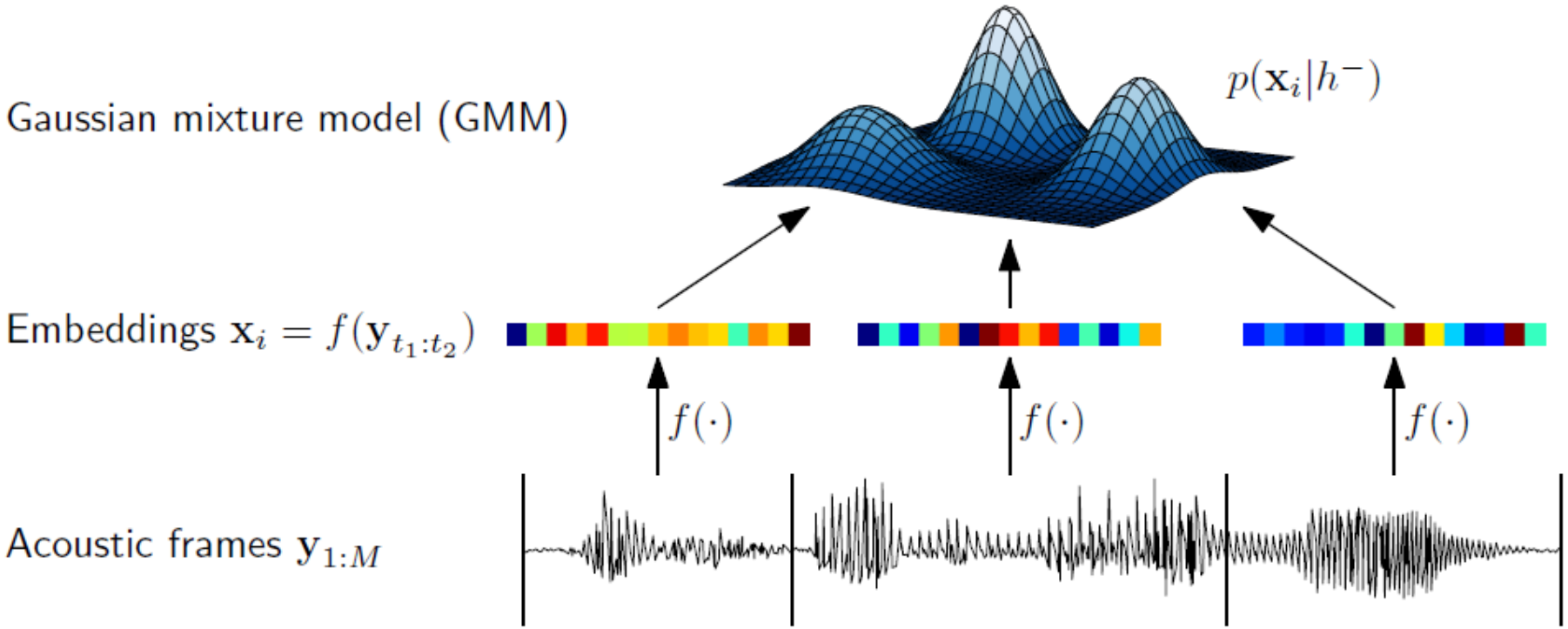
$$P(\mathbf{x}_i|\mu_k, \Sigma_k) = \mathcal{N}(\mu_k, \Sigma_k)$$

  - To decide which lexical item $\mathbf{x}_i$ should belong to:

$$P(\mathbf{x}_i \text{ in } k|\ \mu, \Sigma) \propto P(\mathbf{x}_i|\ \mu_k, \Sigma_k) P(\mathbf{x}_i \text{ in } k)$$

Clustering the embeddings

- Suppose we know the word boundaries and parameters of acoustic model (GMM)
  - Just assign each embedding to a lexical item (GMM component)



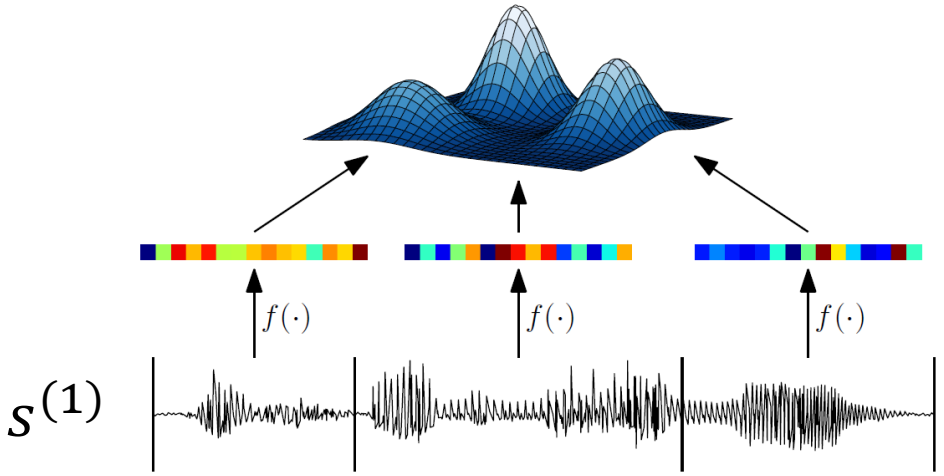
Learning the acoustic model

Suppose we *only* know word boundaries.

- Go Bayesian: add priors and integrate over cluster weights and means.
- Use Gibbs sampling to cluster embeddings while learning acoustic model (unsupervised).
  - Initialize clusters (lexical items) randomly
  - Iteratively re-sample a cluster for one embedding at a time, conditioned on current clusters.

Segmenting an utterance

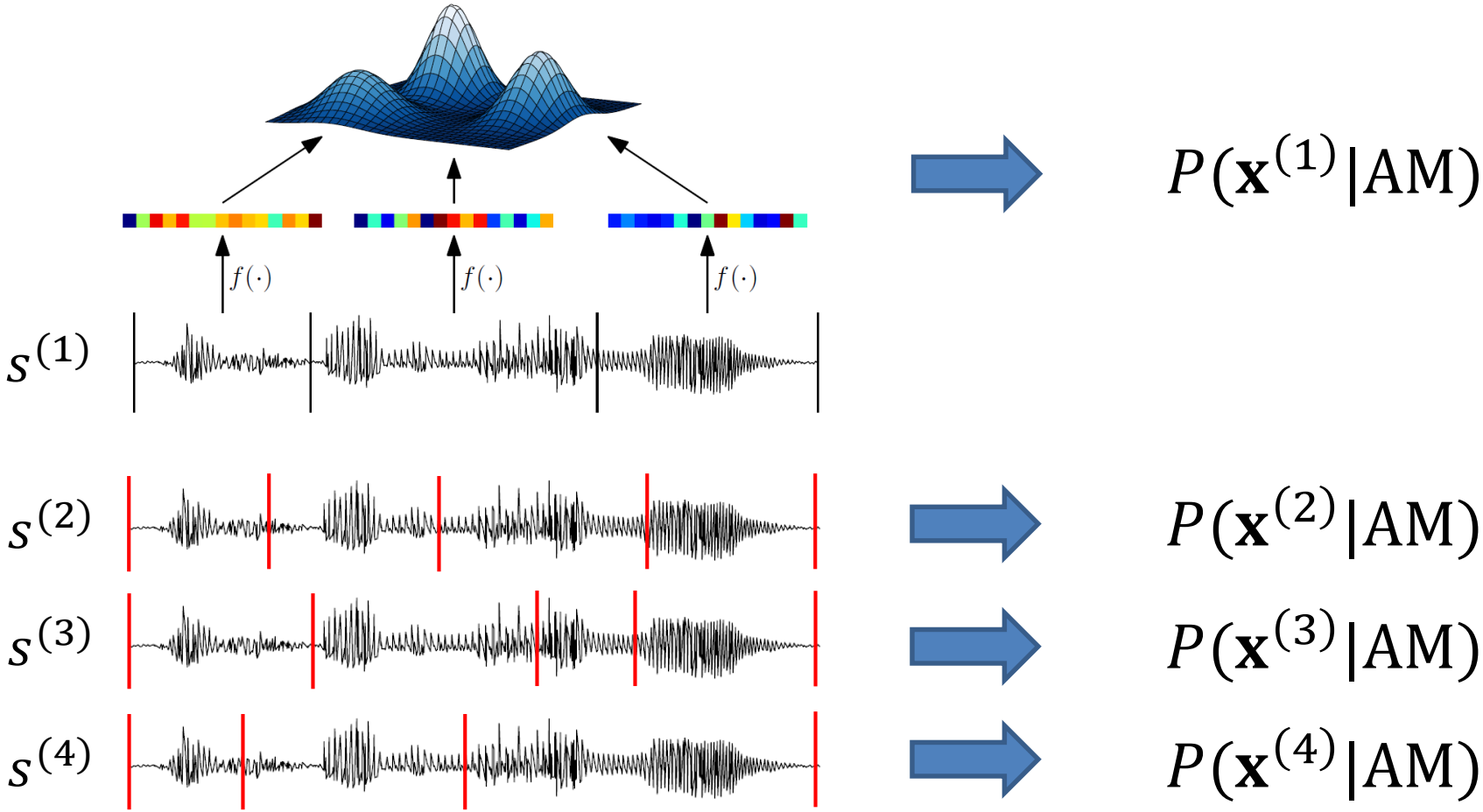
- Suppose we know the acoustic model.
  - Can compute best segmentation of each utterance.



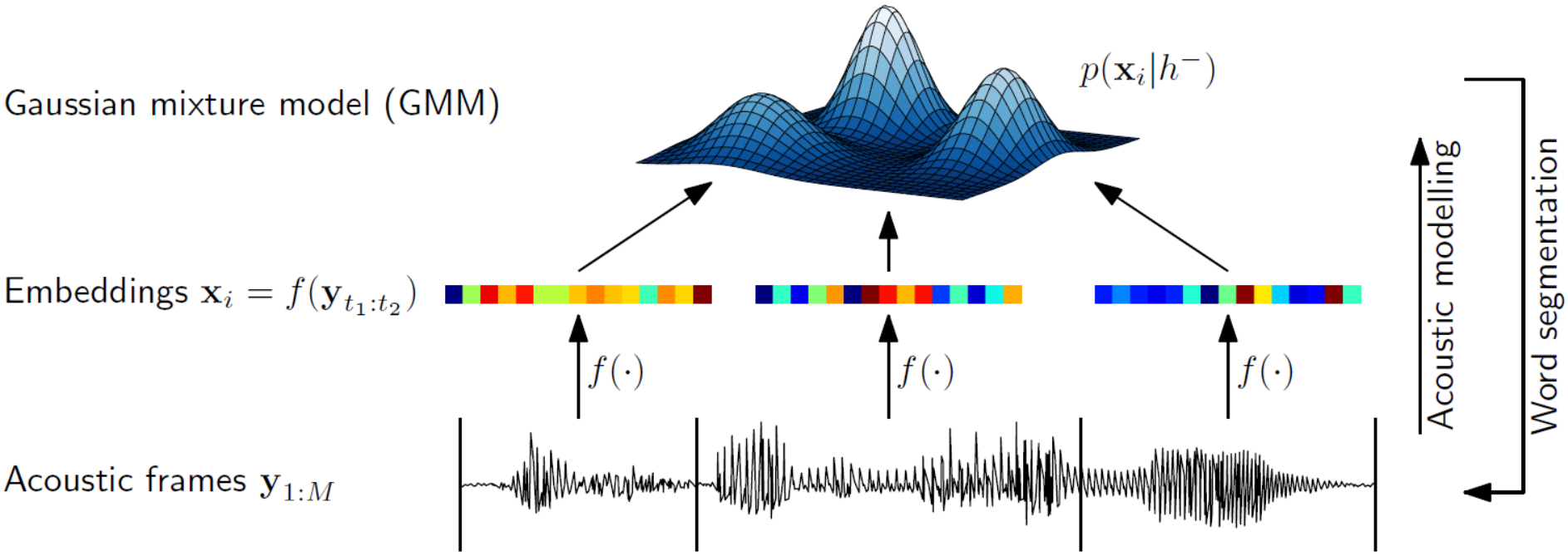
$$P(\mathbf{x}^{(1)}|\mathrm{AM})$$

Segmenting an utterance

- If we know the acoustic model, can compute best segmentation of each utterance.



Recap: circular problem



Gibbs sampling again

Choose initial segmentation (eg, random)

Cluster initial segments to create initial AM

For $i = 1$ to NumIters

For each segmented utterance s

Remove embeddings of s from AM

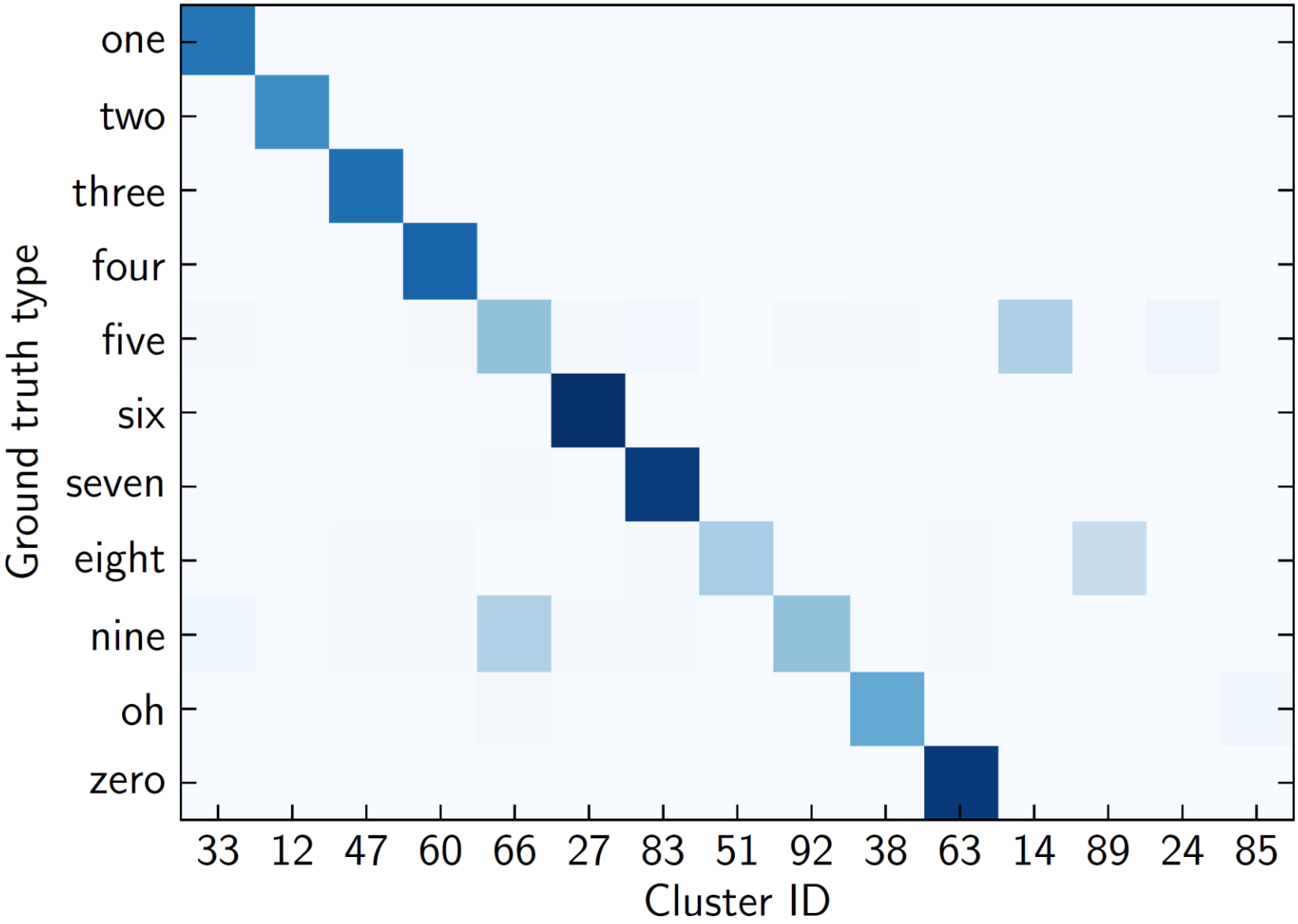
Resample segmentation for s based on current AM

Add new embeddings back into AM

First try: small vocabulary

- TIDigits dataset of English digit sequences
  - 11 words: ‘zero’-’nine’ and ‘oh’
  - ~110 speakers, 80 sequences each.

Confusion matrix of results



Example clusters

- Zero 
- Nine(1),  Nine(2) 
- Five(1),  Five(2) 

Scaling up: large vocabulary

- Use bottom-up cues to identify syllable boundaries, use as potential word boundaries (Räsänen et al., 2016)
- Impose minimum word duration
- Compare MFCC features to CAE features

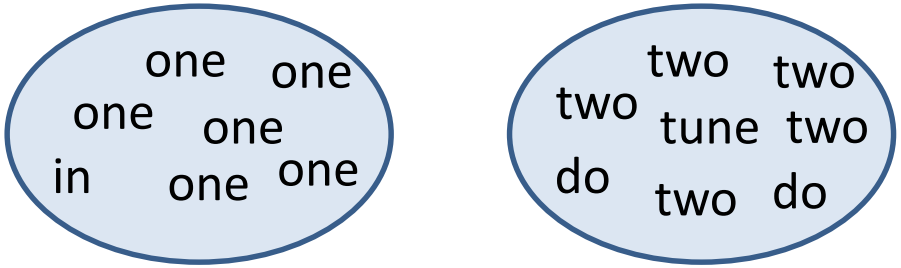
Scaling up: large vocabulary

- Tune: English1. Test: English2, Xitsonga

	English2	Xitsonga
Hours	5.0	2.5
# speakers	12	24
Word tokens	70k	20k
Word types	4538	966
Token/type	15.3	8.7

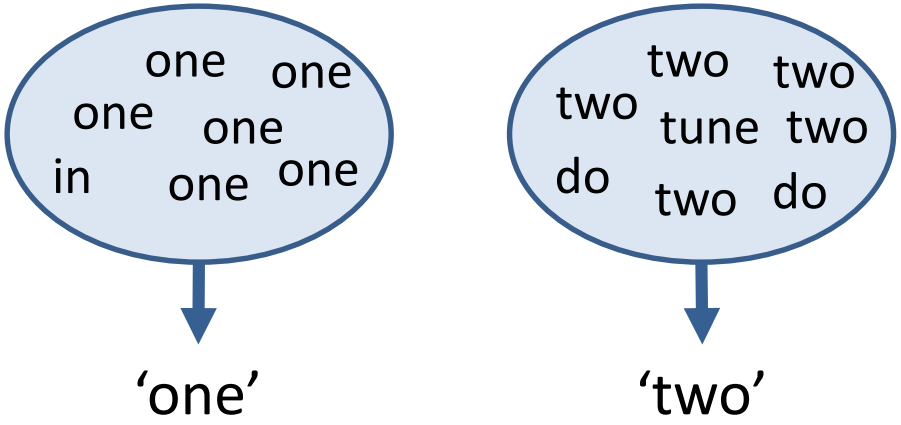
Evaluation measures

- Word Error Rate (WER)
  - Map one cluster to each true word



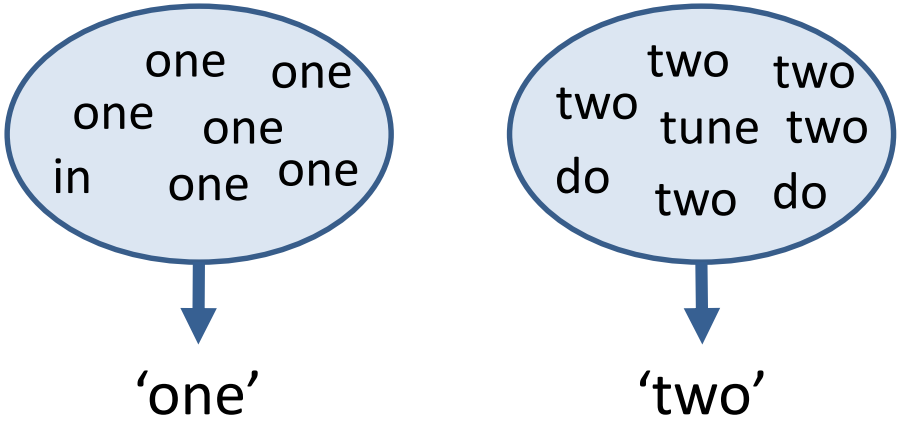
Evaluation measures

- Word Error Rate (WER)
  - Map one cluster to each true word



Evaluation measures

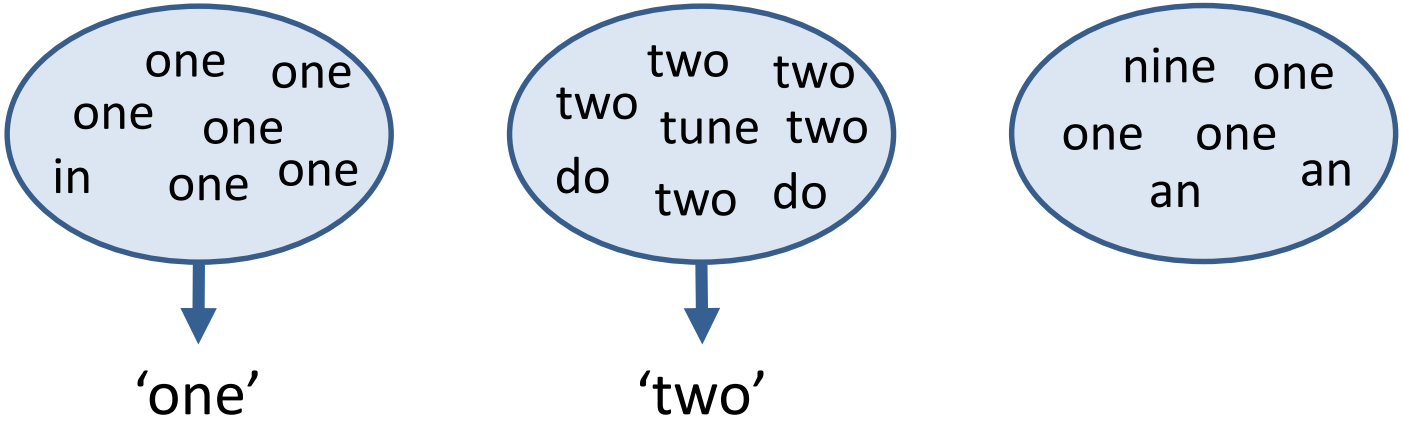
- Word Error Rate (WER)
  - Map one cluster to each true word



  - Align discovered sequences to true sequences to compute standard WER:
    - Count inserts, deletes, substitutes: can be > 100%!

Evaluation measures

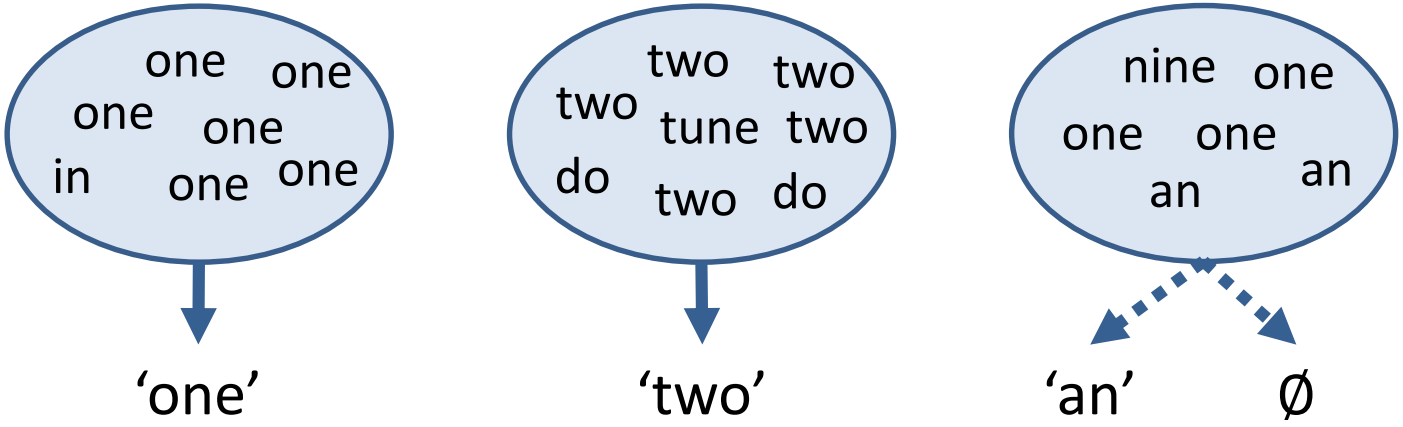
- Word Error Rate (WER)
  - Map one cluster to each true word



  - Align discovered sequences to true sequences to compute standard WER:
    - Count inserts, deletes, substitutes: can be > 100%!

Evaluation measures

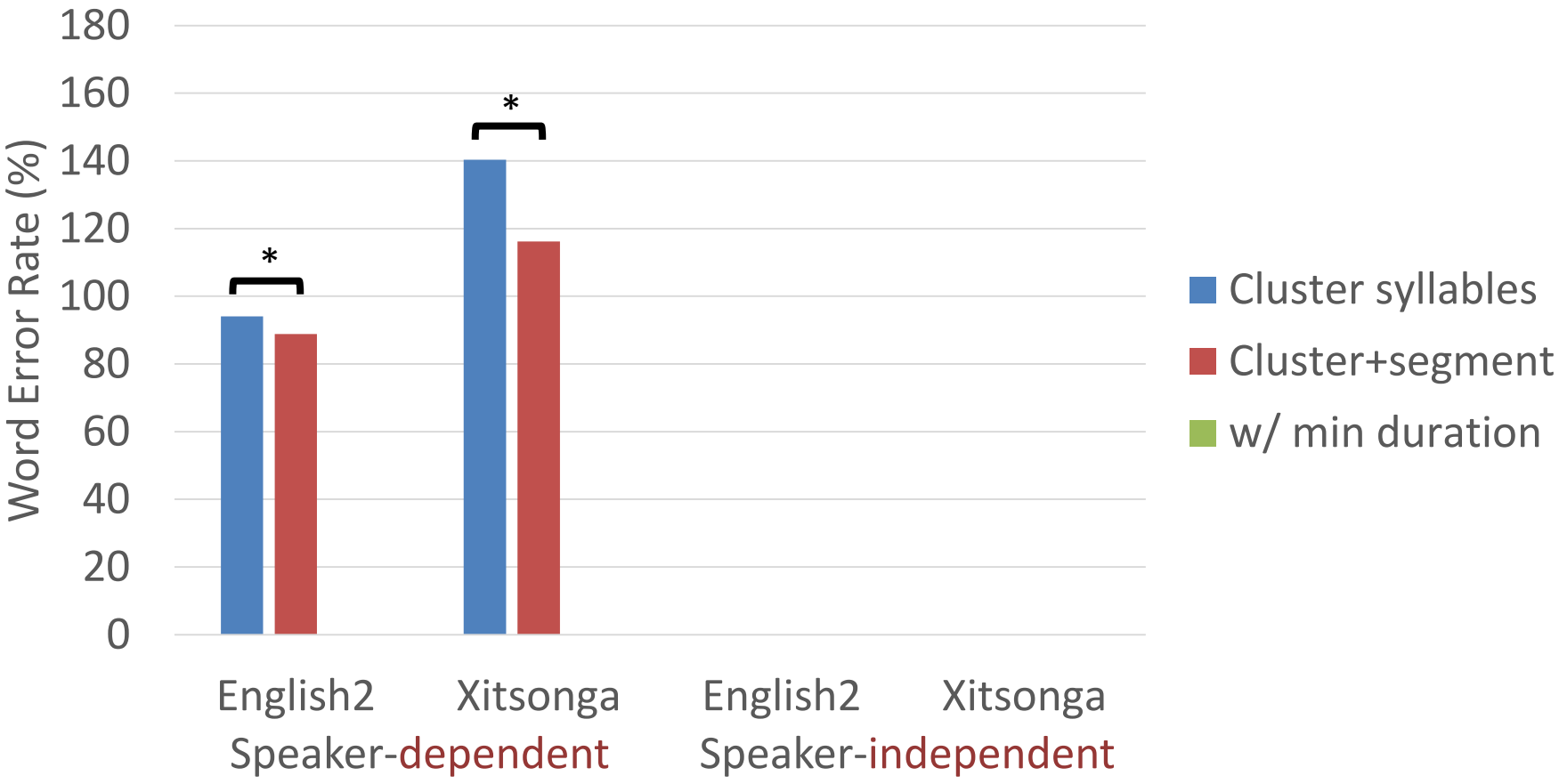
- Word Error Rate (WER)
  - Map one cluster to each true word



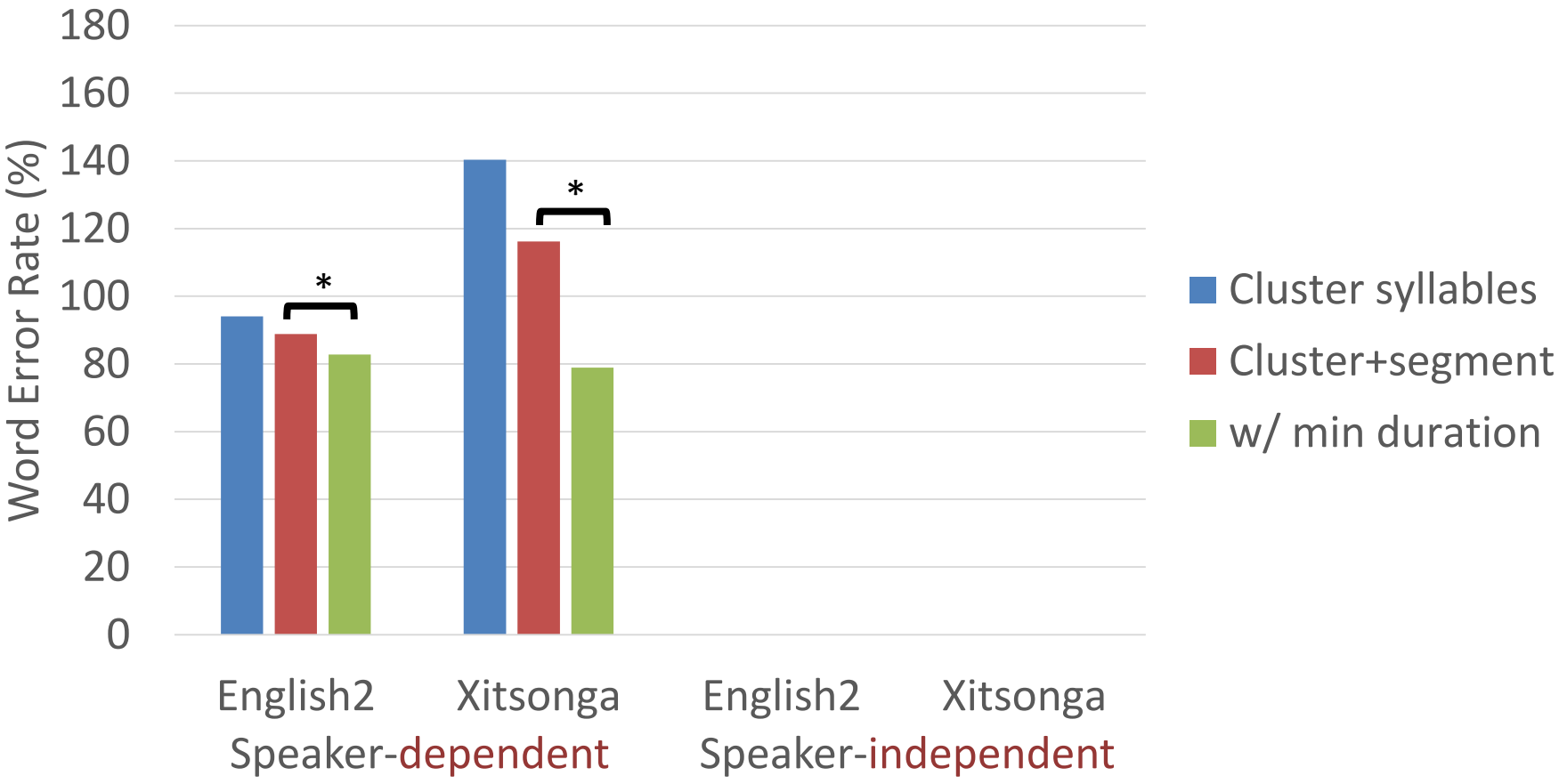
  - Align discovered sequences to true sequences to compute standard WER:
    - Count inserts, deletes, substitutes: can be > 100%!

Model comparisons

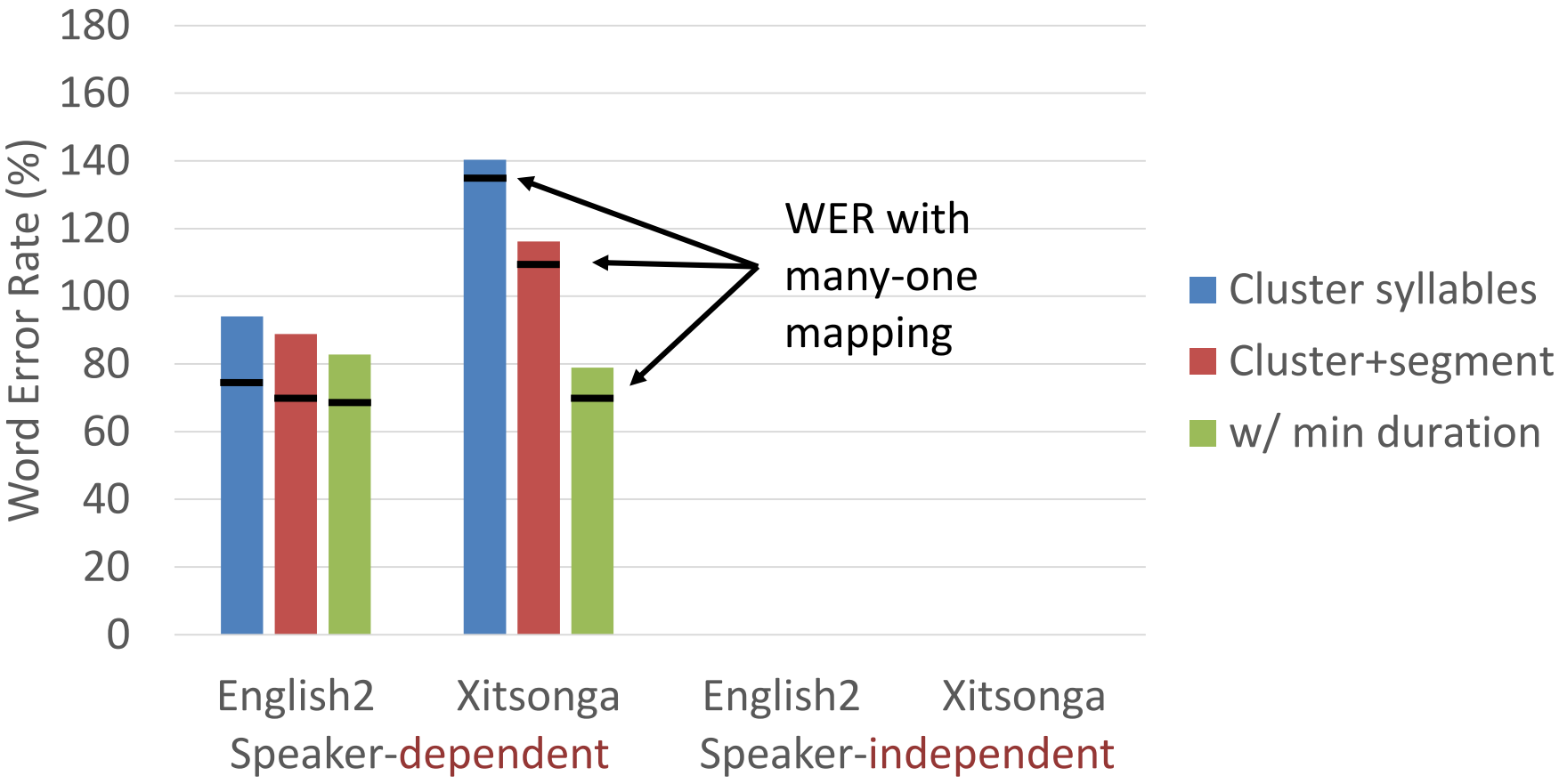
Model comparisons



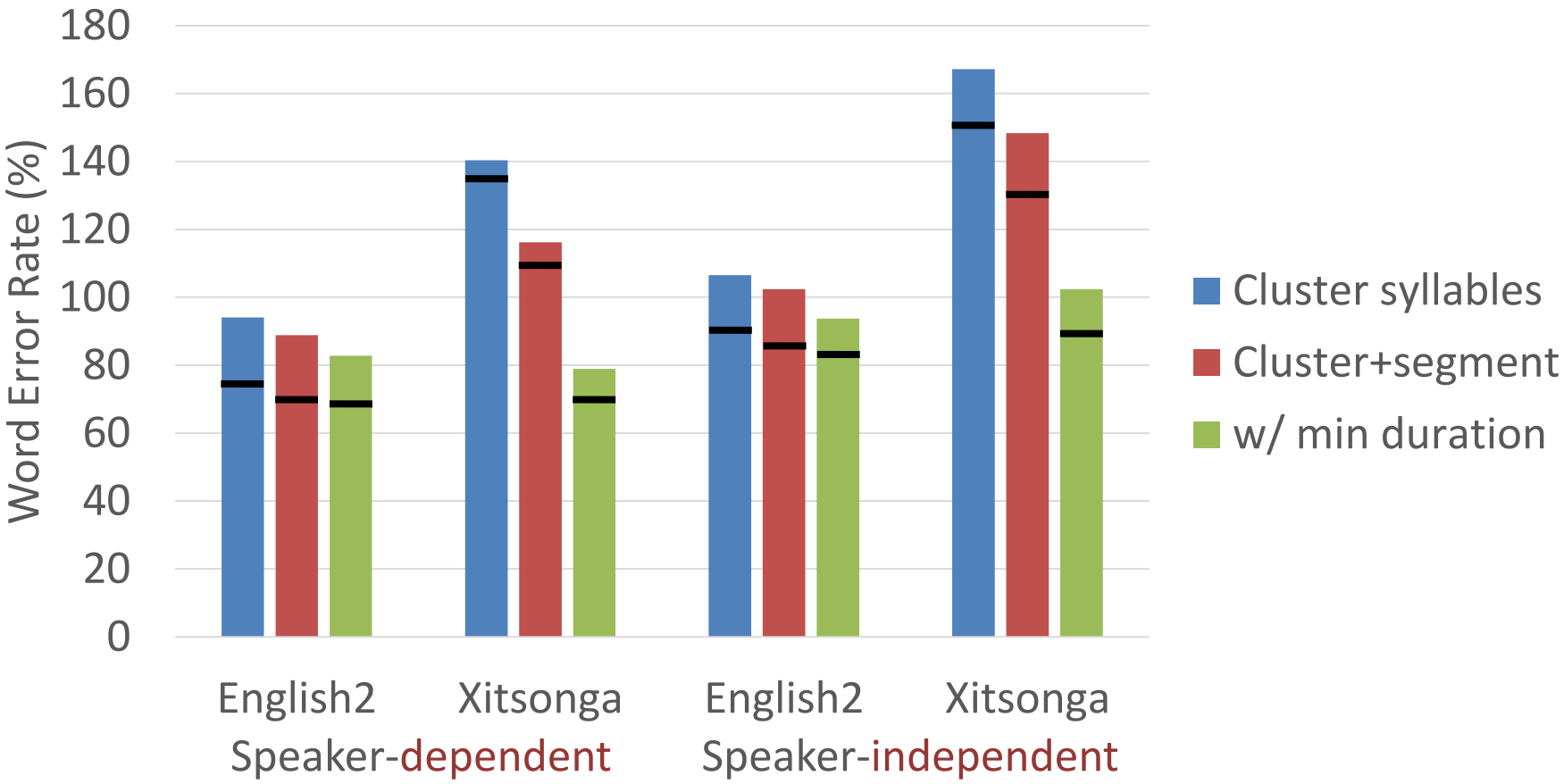
Model comparisons



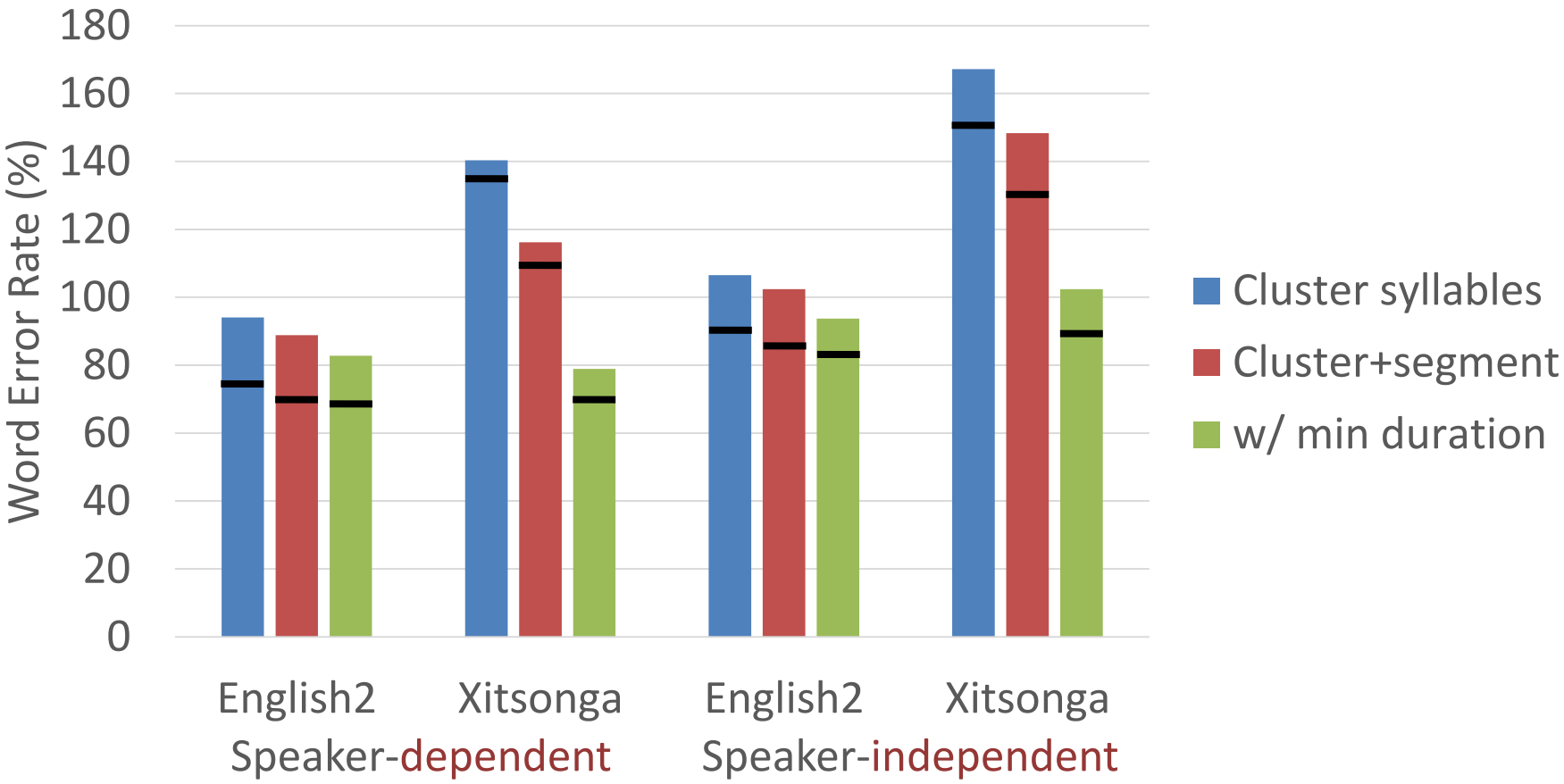
Model comparisons



Model comparisons



Model comparisons



- Also: For SI only, CAE (learned) features as good or better than MFCC, with more mixed clusters

Summary

- Probabilistic model with acoustic embeddings.
- Combines both top-down and bottom-up information in joint segment/cluster model.
- First unsupervised full-coverage ASR system for large vocab, multi-speaker data.
  - Also outperforms previous systems on single speaker data.

Outline

1. Background
2. Unsupervised monolingual models
3. Using information from other languages
   - Multilingual training to improve subword representations
   - Speech-to-text translation for low-resource languages

E. Hermann and S. Goldwater. Multilingual bottleneck features for subword modeling in zero-resource languages. In *Proceedings of Interspeech*. 2018.

E. Hermann, H. Kamper, and S. Goldwater. Multilingual and unsupervised subword modeling for zero-resource languages. In submission (and on arXiv).

Recap

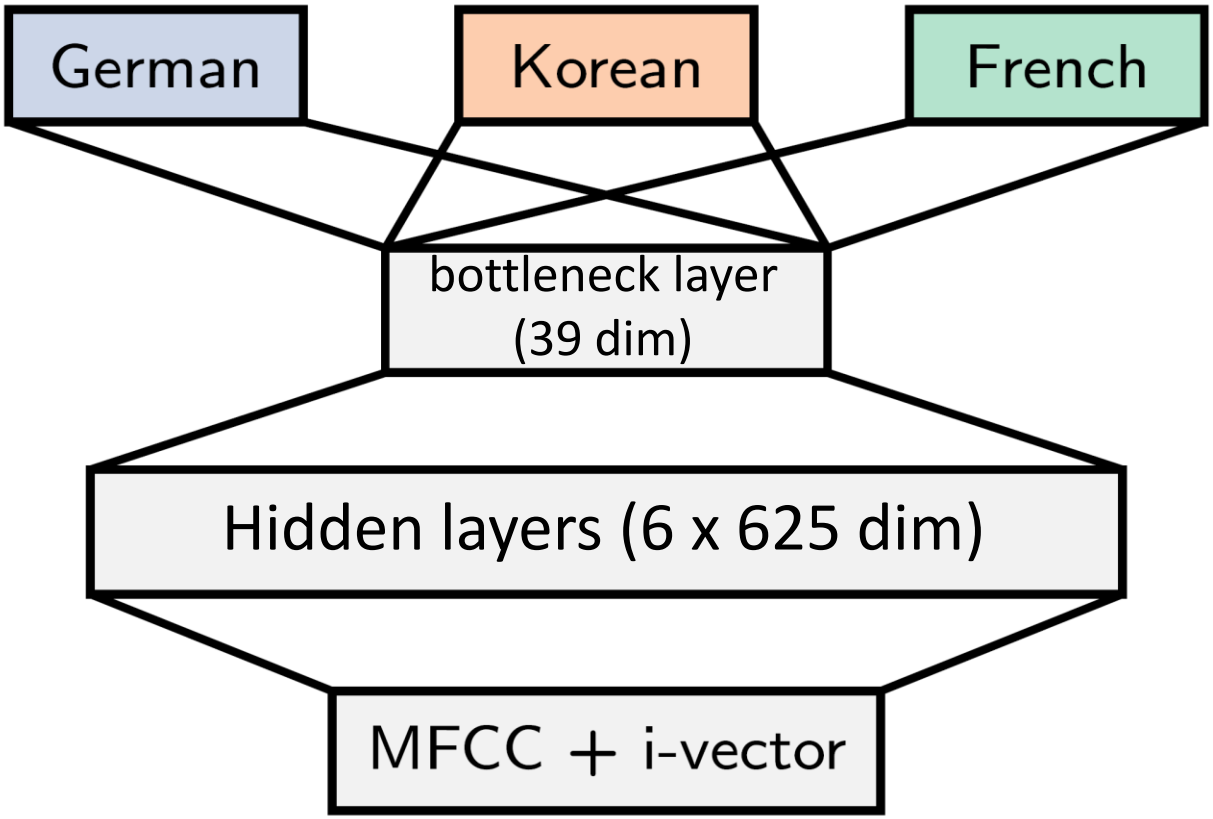
- Feature extraction from speech is hard
  - aim for phonetic information
  - abstract away from speaker, gender, channel noise, phonetic context
- We saw that unsupervised top-down info from same language can help.
- What about supervision in other languages?

Cross-lingual supervision

- Could help with variability due to speaker, gender, noise…
- …even if not with language-specific phonetic acoustic mapping and dynamics
- Hypothesis: for equivalent total data, using data from multiple languages will work better than from one language

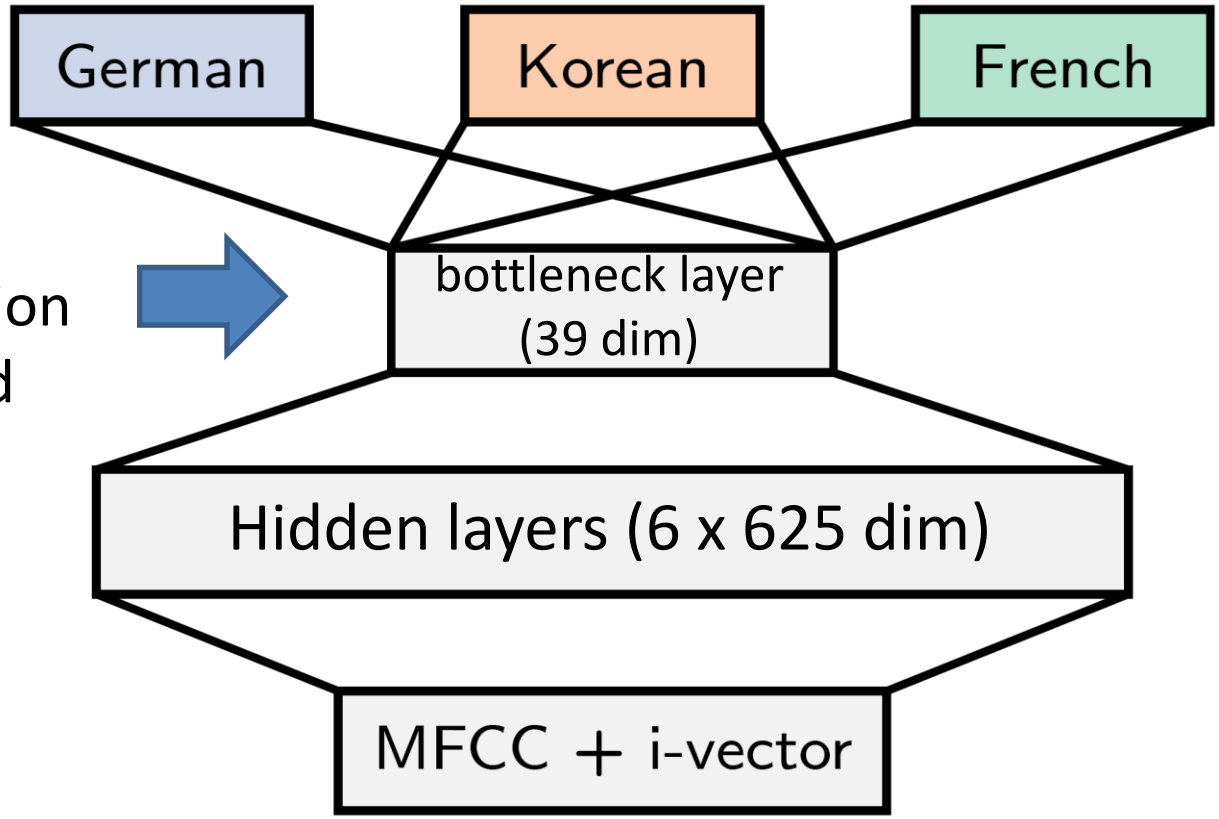
Multilingual bottleneck features (BNFs)

- DNN trained to predict phone labels, using data from high-resource languages:



Multilingual bottleneck features (BNFs)

- DNN trained to predict phone labels, using data from high-resource languages:

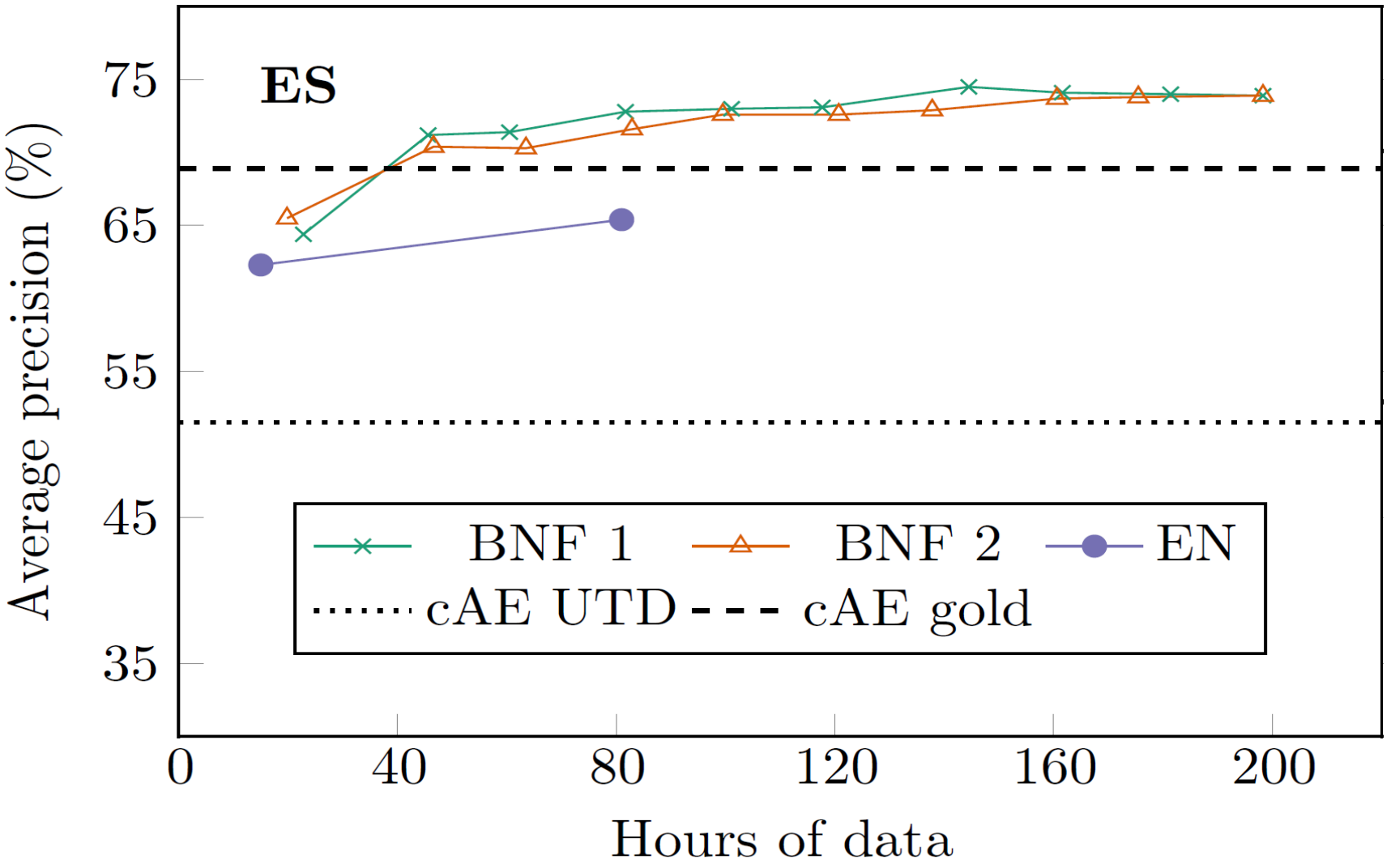


Learned representation then applied to unseen language

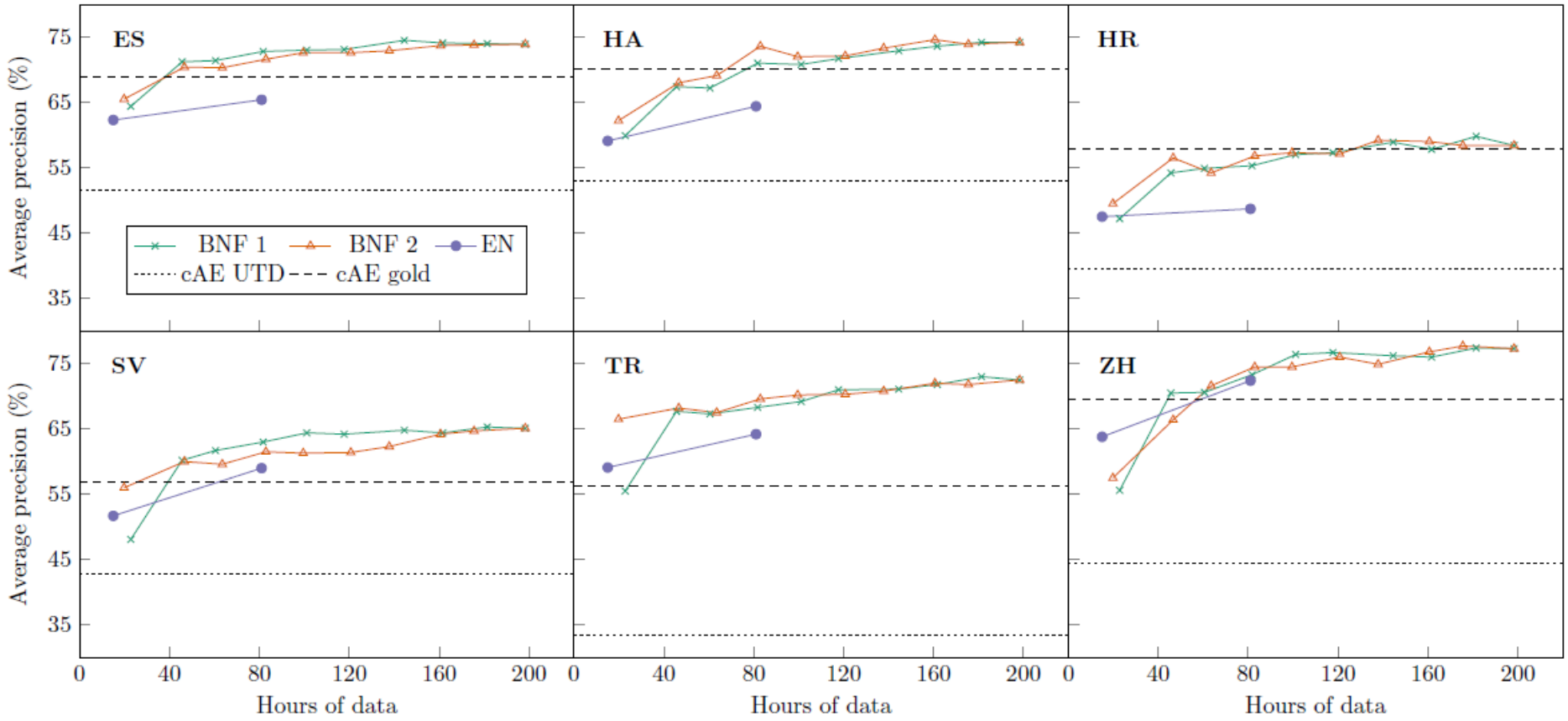
Experimental setup

- Training:
  1. ~20 hrs each of 10 languages
  2. 15 or 80 hrs of one language (English)
- Evaluation:
  1. word discrimination task on 6 unseen languages
  2. plug BNFs into unsupervised segmentation/ clustering system (English and Xitsonga)

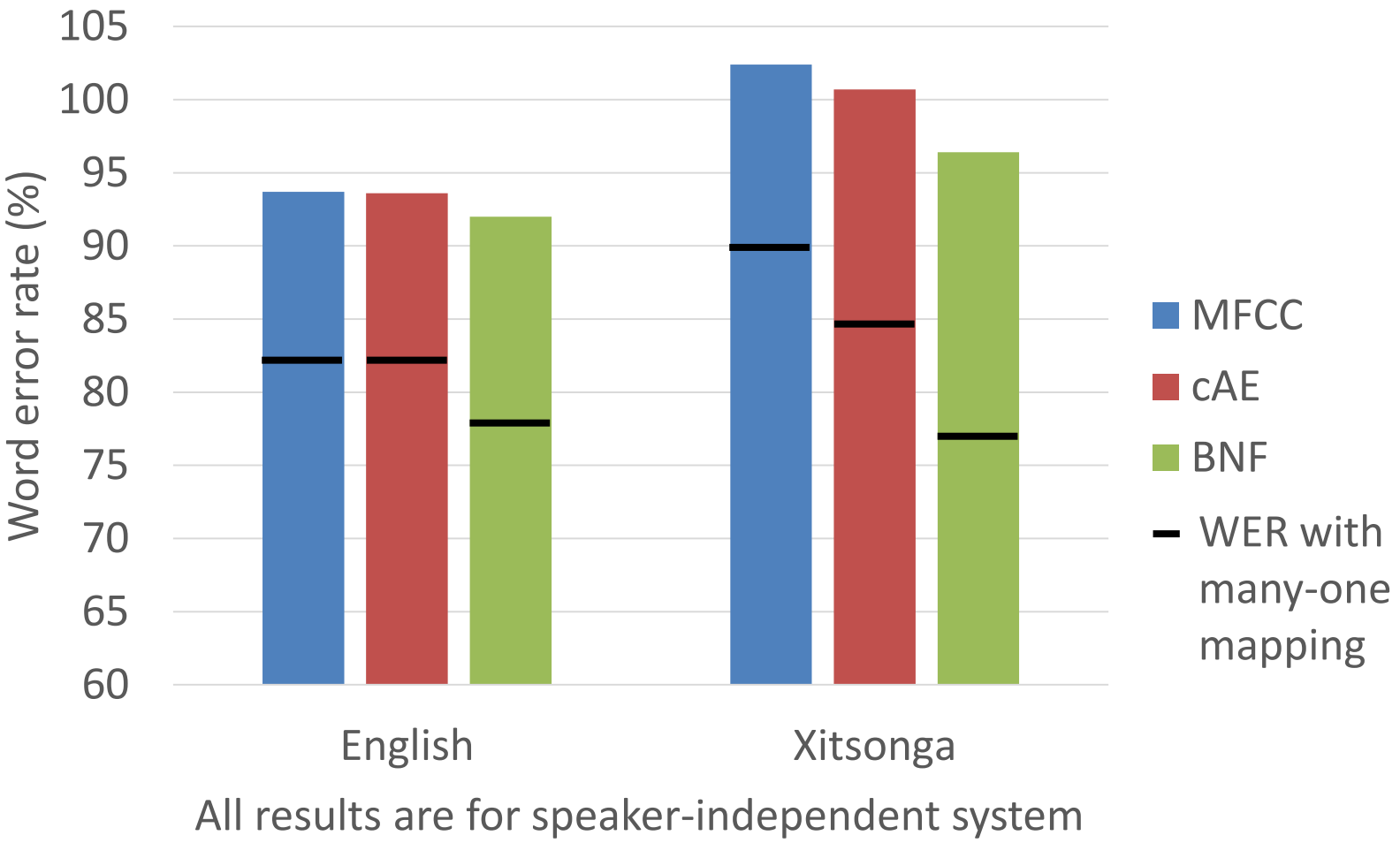
Word discrim results (on Spanish)



Word discrim results (all test langs)



Unsupervised seg/clust results



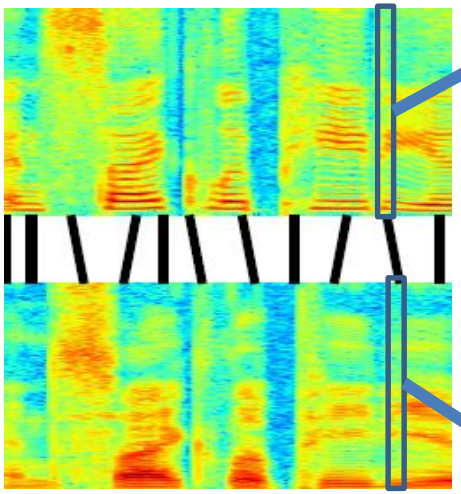
- Boundary accuracy and cluster purity also improve

Combining BNF and cAE

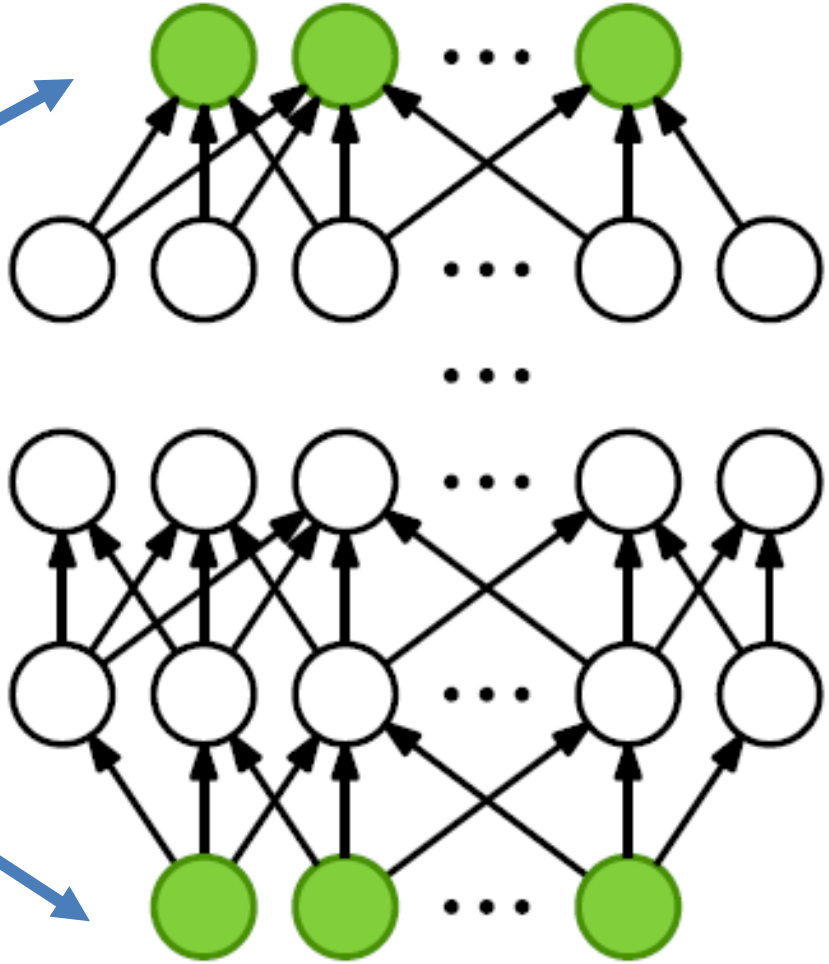
- BNF uses supervision on non-target languages
- cAE uses target language data
- Why not do both?

Reminder: cAE

Two examples of a word



Align frames (use DTW)

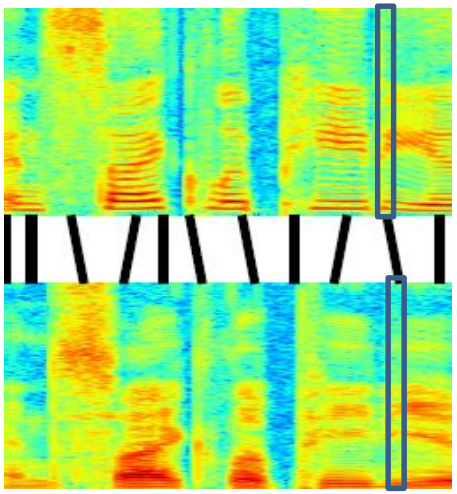


Experimental settings

Two examples of a word

Before: represented using MFCCs

Now: represented using BNFs



Align frames (use DTW)

Experimental settings

Before: represented using MFCCs

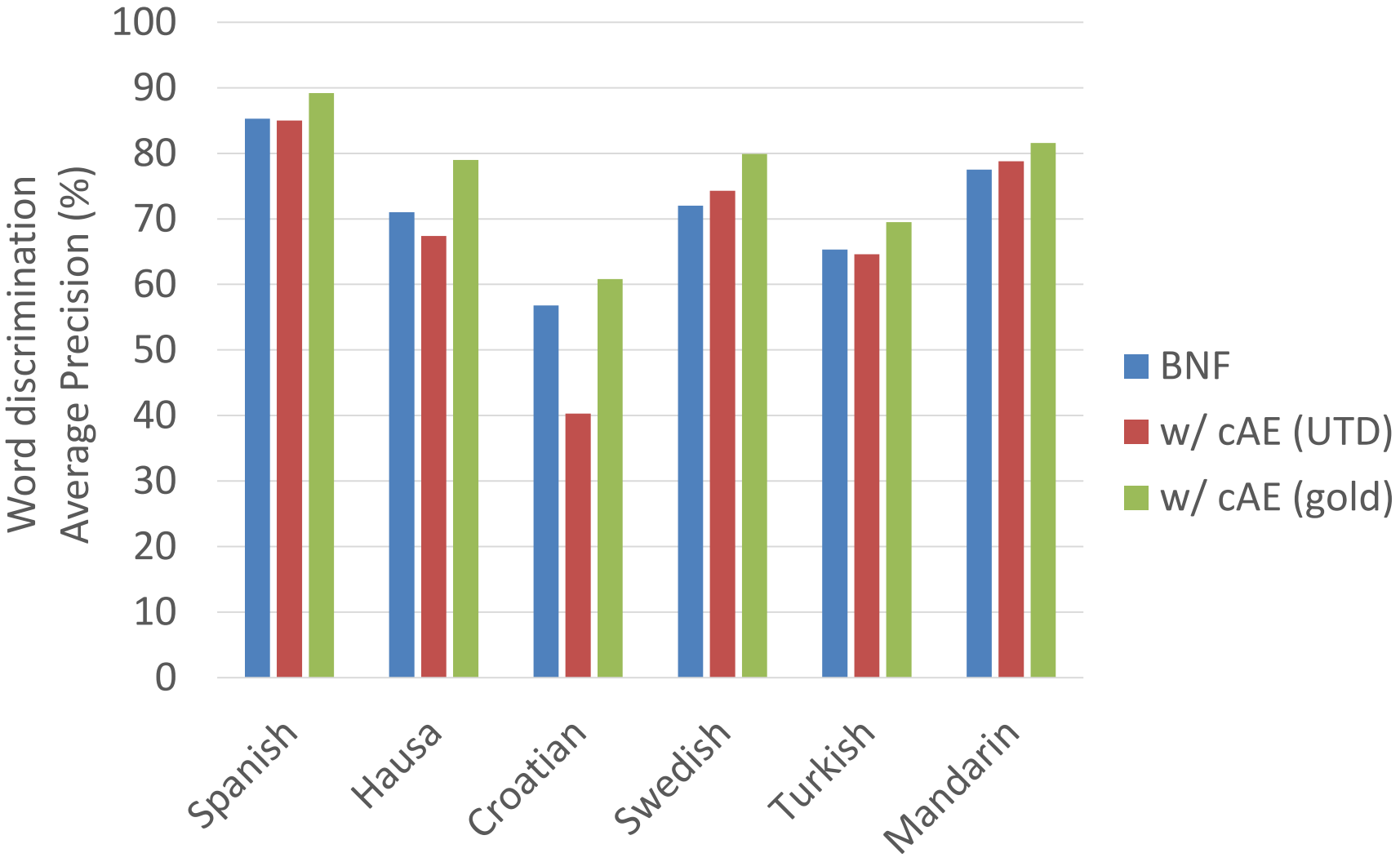
Now: represented using BNFs

Either: using UTD (unsupervised)

Or: using gold standard (supervised)

Align frames (use DTW)

Combining helps (with gold pairs)



Application to African radio data [1]

- Aim: extend system for keyword spotting in call-in shows [2,3] to more languages.
  - Currently in use for UN development/relief programmes in Uganda.
  - But using full ASR system (i.e., transcribed data)

[1] R. Menon, H. Kamper, J. Quinn, & T. Niesler. Almost zero-resource ASR-free keyword spotting using multi-lingual bottleneck features and correspondence autoencoders. *arXiv preprint arXiv:1811.08284,* 2018.
[2] R. Menon et al. Radio-browsing for developmental monitoring in Uganda. *Proc. ICASSP*, 2017.
[3] A. Saeb et al. Very low resource radio browsing for agile developmental and humanitarian monitoring. *Proc. Interspeech*, 2017.

Application to African radio data [1]

- Can an ASR-free system be developed with much more limited supervision?

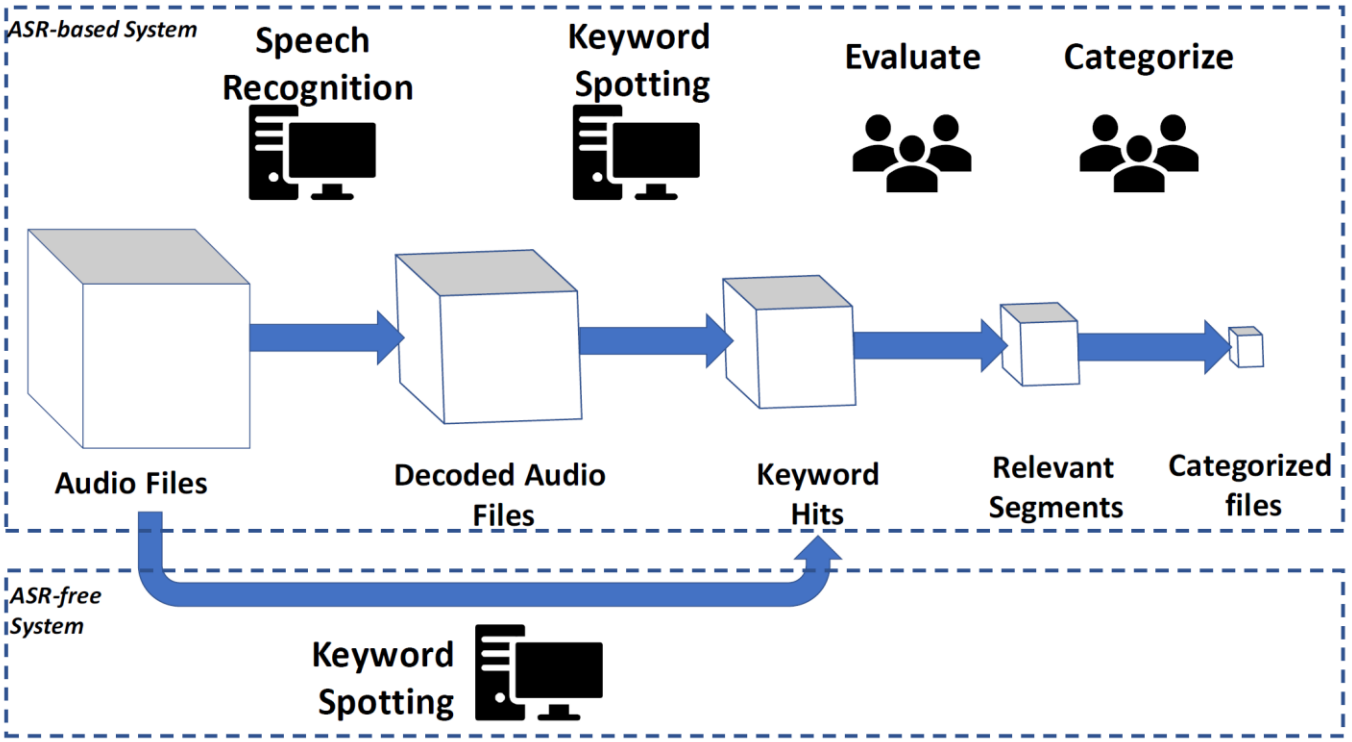
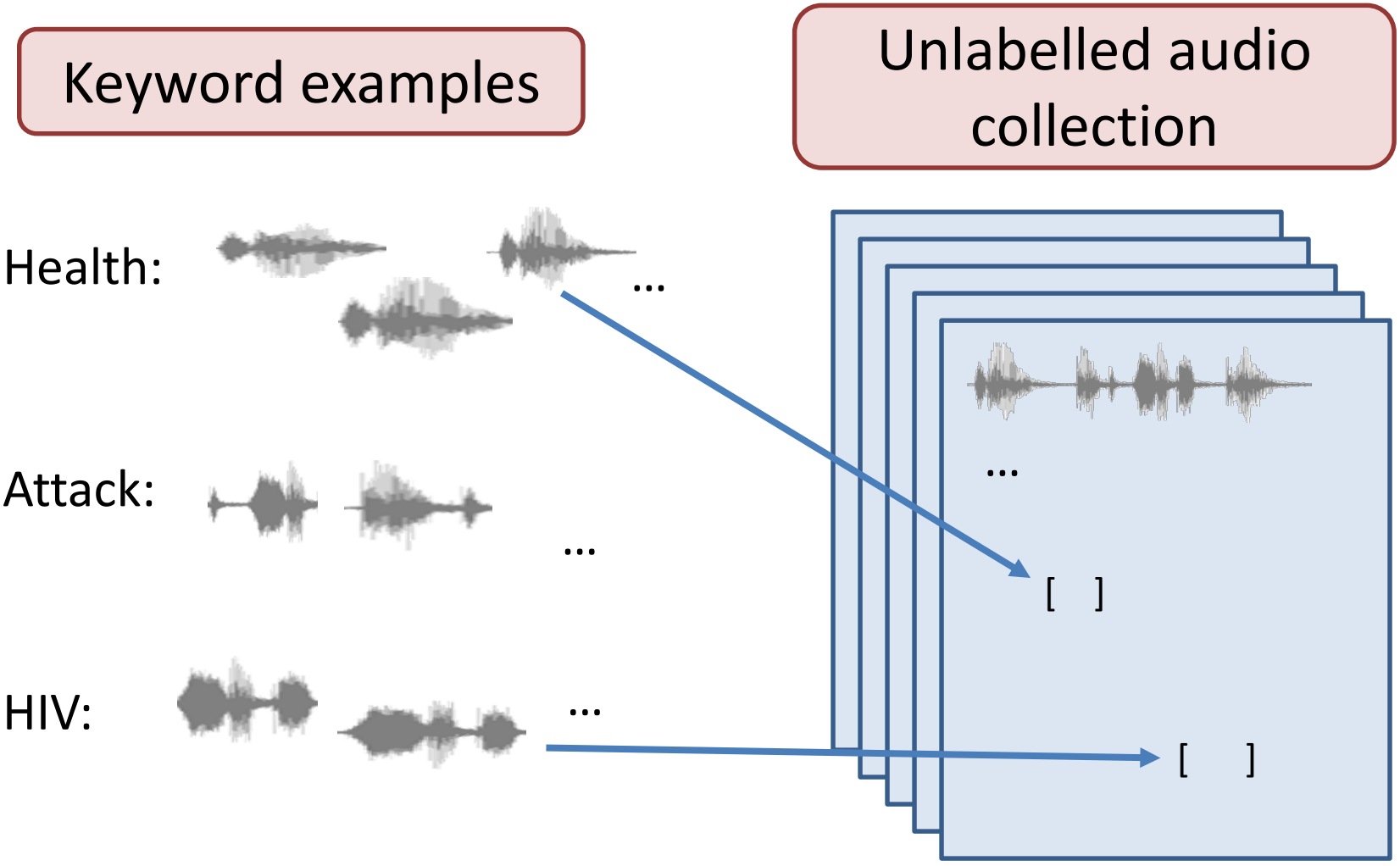


Figure from [1]

[1] R. Menon, H. Kamper, J. Quinn, & T. Niesler. Almost zero-resource ASR-free keyword spotting using multi-lingual bottleneck features and correspondence autoencoders. *arXiv preprint arXiv:1811.08284,* 2018.

Keyword spotting system



Experiments

- Test data: South African Broadcast News (Eng)
- Labelled data: 40 keywords
  - 24 speakers recorded two examples of each

Experiments

- Test data: South African Broadcast News (Eng)
- Labelled data: 40 keywords
  - 24 speakers recorded two examples of each
- Results: BNF+cAE = big improvements!

	AUC	Prec@10
MFCC	75.2	17.0
CAE on MFCC	76.0	25.0
BNF	77.0	22.8
CAE on BNF	86.4	45.7

Summary

- With no labeled target language data, multilingual BNFs
  - greatly improve word discrimination (even more than cAE)
  - also help unsupervised segmentation/clustering and keyword spotting
- Using more languages is better than just more data from one language
- Features can be further improved using just a handful of labelled words in target language

Outline

1. Background
2. Unsupervised monolingual models
3. Using information from other languages
   - Multilingual training to improve subword representations
   - Speech-to-text translation for low-resource languages

S. Bansal, H. Kamper, K. Livescu, A. Lopez, and S. Goldwater. Low-Resource Speech-to-Text Translation. *Proceedings of Interspeech*. 2018.

S. Bansal, H. Kamper, K. Livescu, A. Lopez, and S. Goldwater. Pre-training on high-resource speech recognition improves low-resource speech-to-text translation. In submission (and on arXiv).

Traditional speech translation

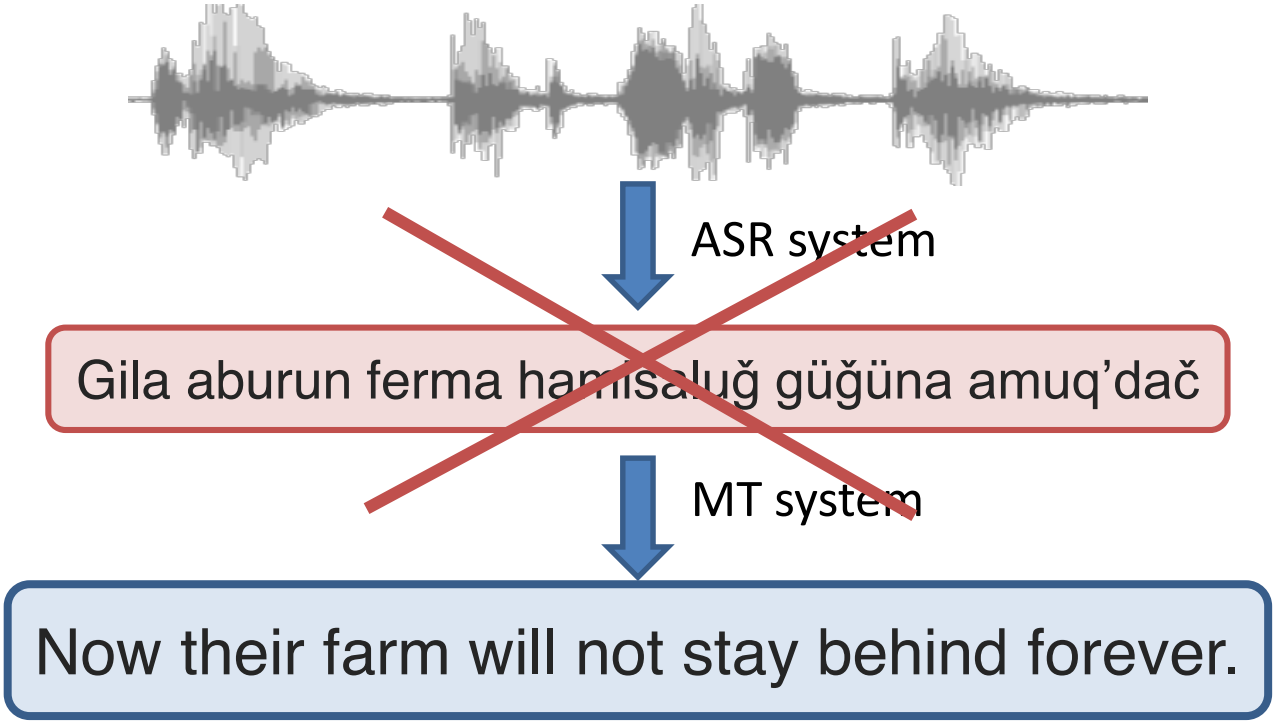
ASR system

Gila aburun ferma hamišaluğ güğüna amuq'dač

MT system

Now their farm will not stay behind forever.

End-to-end speech translation



End-to-end speech translation

ST system

Now their farm will not stay behind forever.

End-to-end speech translation

ST system

Now their farm will not stay behind forever.

- Potential uses:
  - Endangered language documentation
  - Translation for languages without writing systems

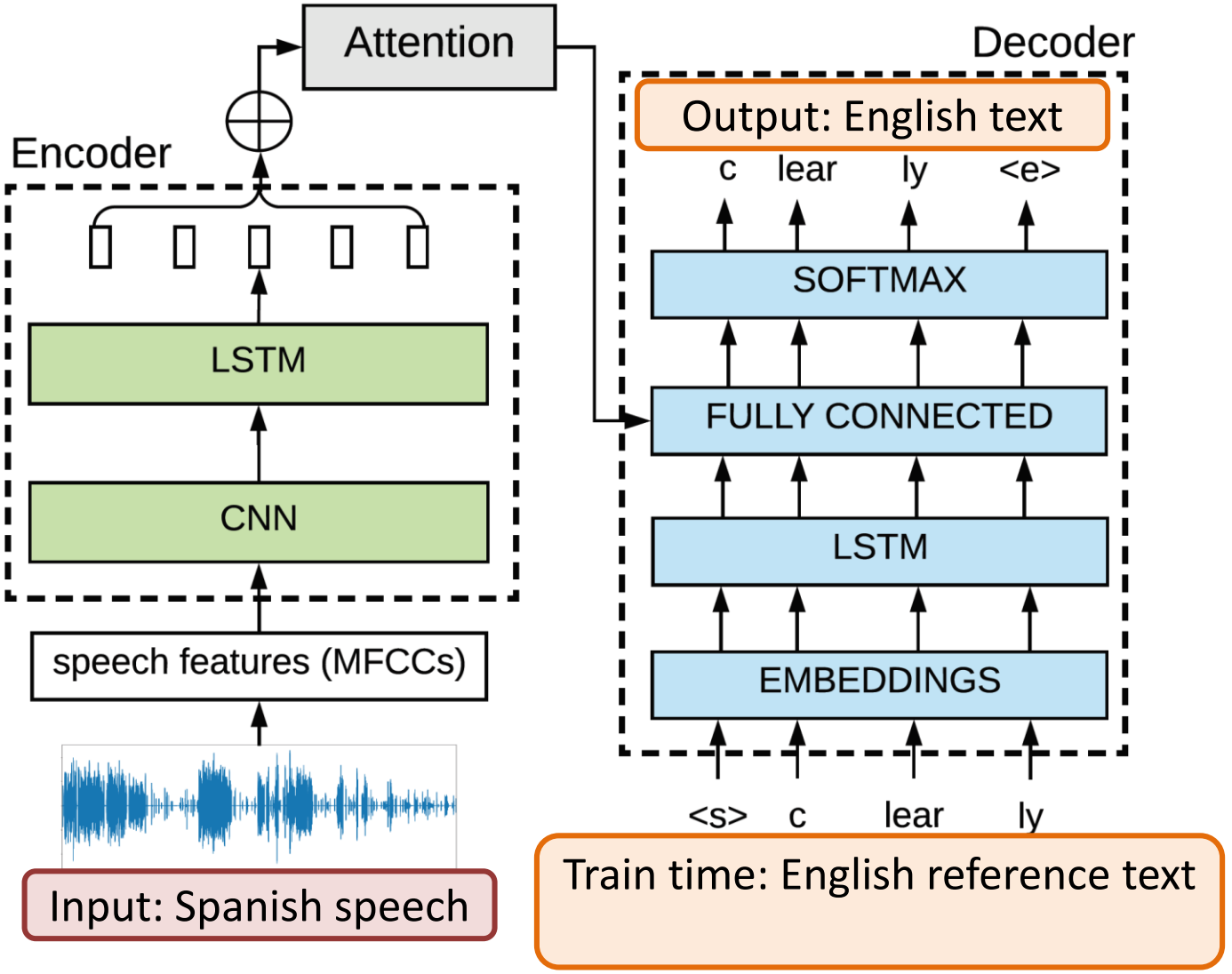
Low-resource scenario

- What is possible with very little parallel data?
- Can cross-lingual pre-training/transfer help?

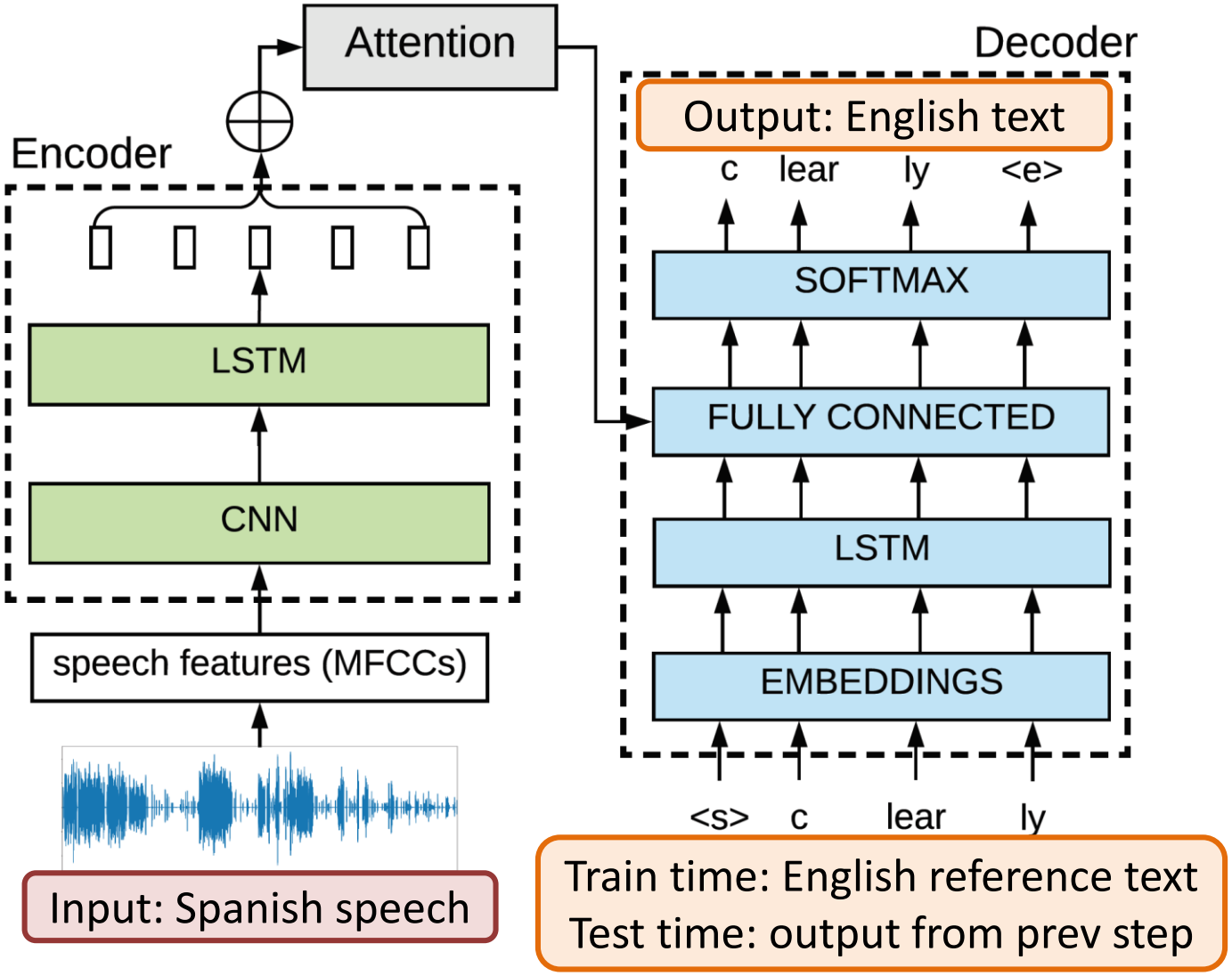
Low-resource scenario

- What is possible with very little parallel data?
- Can cross-lingual pre-training/transfer help?
- Datasets used:
  1. Up to 50 hrs of Spanish-English parallel data
  2. 4 hrs of Mboshi-French parallel data
  3. Additional transcribed data from English (300hrs) and French (20hrs)
- Use a deep sequence-to-sequence network

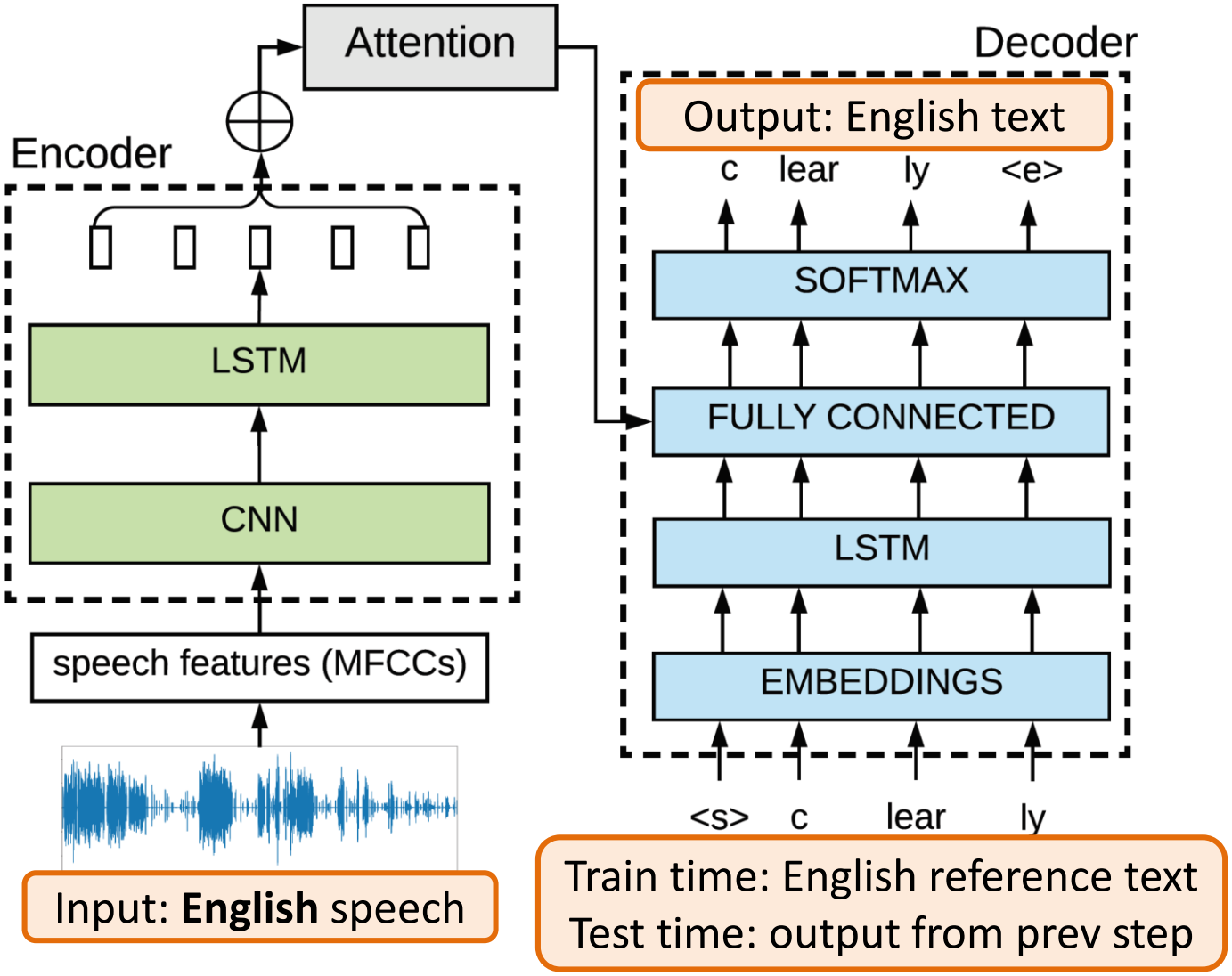
ST network architecture



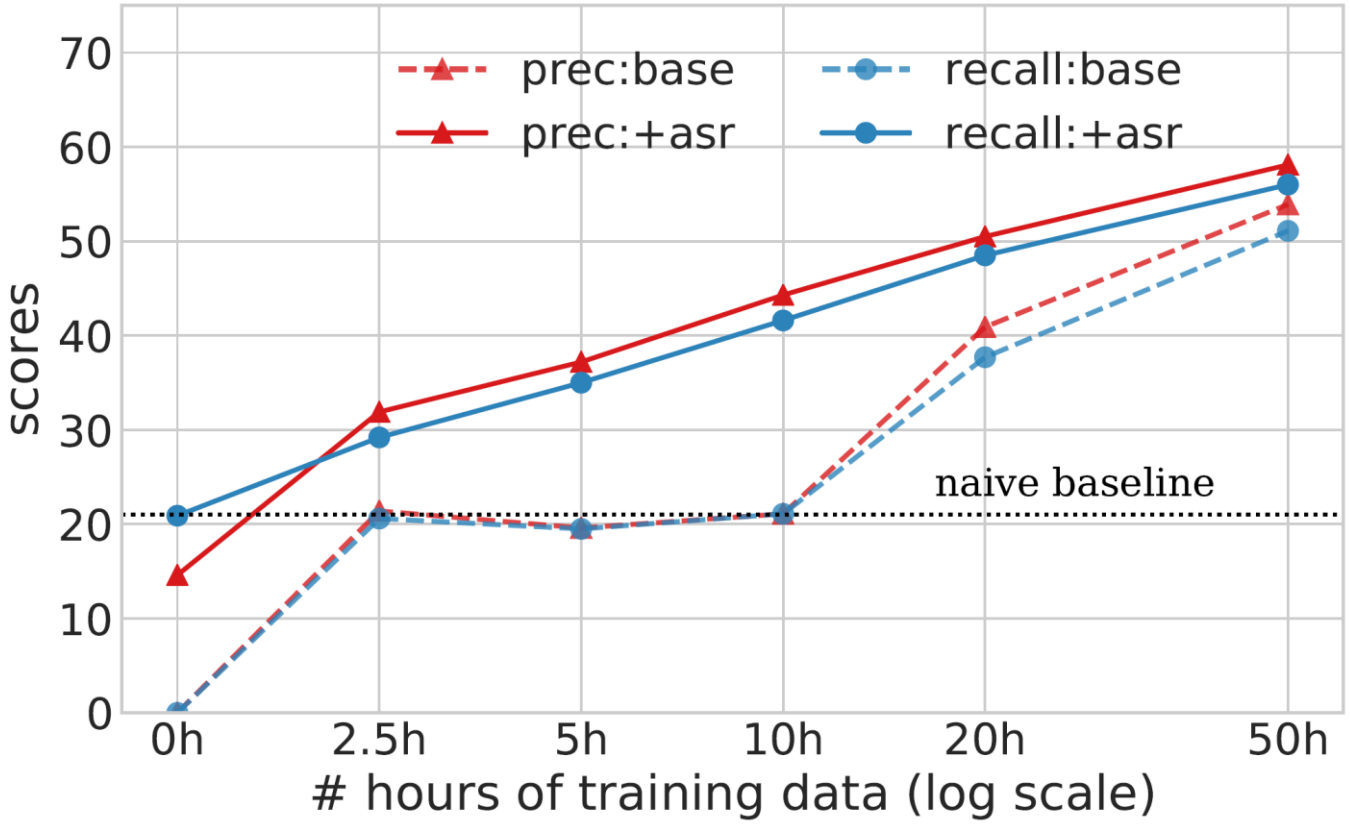
ST network architecture



ASR network architecture



How much data is needed? (Sp-En)

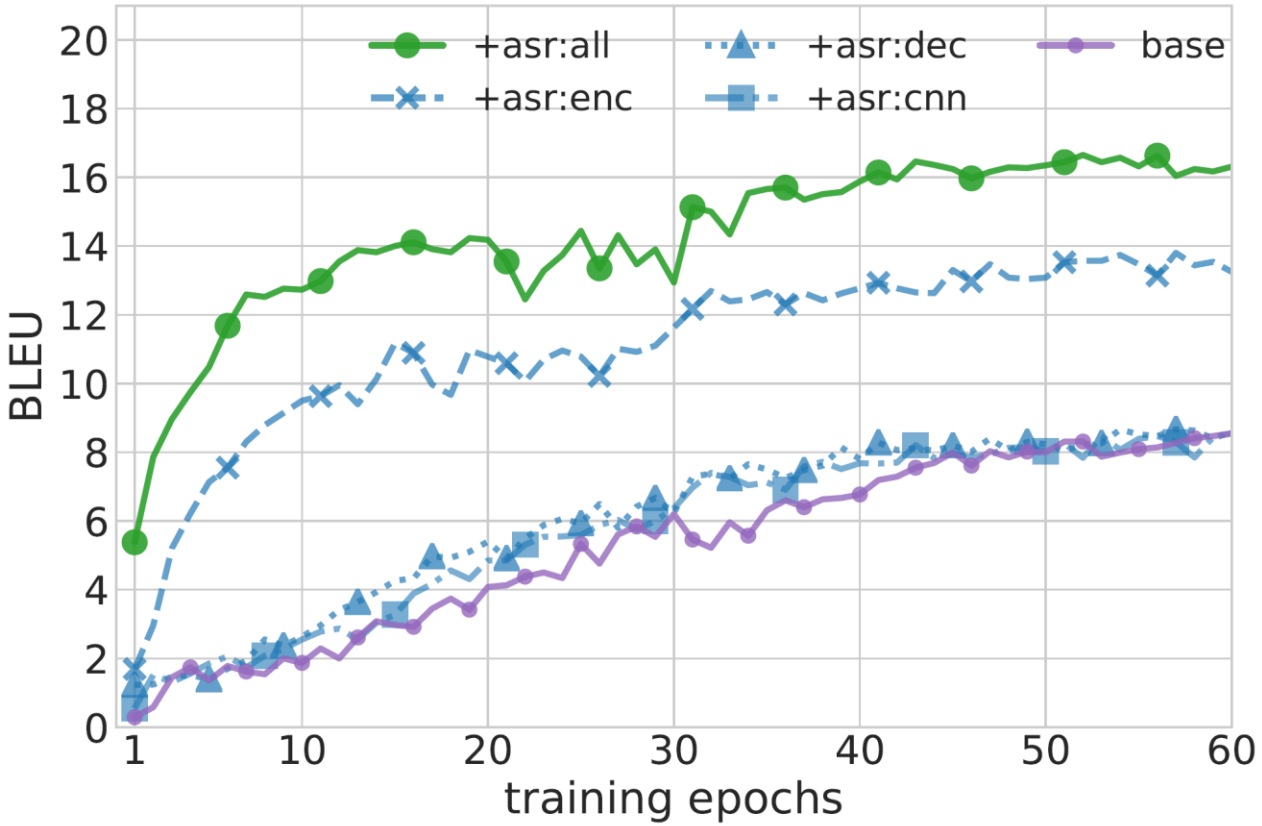


- >50% of tokens translated correctly w/ 50hrs
- Pretraining on English ASR helps a lot
  - here: unigram scores; even more for BLEU scores

Example output

Spanish	sí y usted hace mucho tiempo que que vive aquí
English	yes and have you been living here a long time
20h	yes i've been a long time what did you come here
20h+asr	yes and you have a long time that you live here
50h	yes you are a long time that you live here
50h+asr	yes and have you been here long

Encoder parameters help the most



- Even though ST and ASR both decode to English!
- Implies: speech variability is the bigger problem.

Results on Mboshi-French

- Only 4 hours of training data!
- Best results from combining pretrained EN encoder and FR decoder, then fine-tuning:

	Precision	Recall
ST, no pretraining	18.6	19.4
Top 8 FR words	23.5	22.2
ST with pretraining	26.7	23.1

Summary

- Ok-ish translations are possible with just 20-50 hrs of parallel data.
- Pretraining to help speech side is really important with this little data.
- Current/future work:
  - Can we translate test data well enough to (say) do cross-lingual keyword spotting or topic modelling?
  - Could multilingual pre-training improve further?

Conclusions

- Low-resource speech technology is important for universal access and language documentation.
- Pure bottom-up systems won't work. Need another source of information:
  - Monolingual: top-down info to improve features, joint learning for segmentation and clustering
  - Other languages: multilingual BNFs or pretraining to improve features
- Lots of room for future work and applications!

Thanks!

- To the MLSS organizers,
- My coauthors on this work:

  Sameer Bansal, Micha Elsner, Enno Hermann, Herman Kamper, Karen Livescu, Adam Lopez, Aren Jansen, and Daniel Renshaw

- And to these folks for their cool followup:

  Raghav Menon, Herman Kamper, John Quinn, and Thomas Niesler