CMU CS11-737: Multilingual NLP

Improved Modeling/Learning Methods for Sequence Labeling and Classification

Graham Neubig

Carnegie Mellon University
Language Technologies Institute

Text Classification

- Given an input text *X*, predict an output label *y*

Topic Classification

I like peaches and pears → food / politics / music / ...

I like peaches and herb → food / politics / music / ...

Language Identification

I like peaches and pears → English / Japanese / German / ...

桃と梨が好き → English / Japanese / German / ...

Sentiment Analysis (sentence/document-level)

I like peaches and pears → positive / neutral / negative

I hate peaches and pears → positive / neutral / negative

... and many many more!

Sequence Labeling

- Given an input text *X*, predict an output label sequence *Y of equal length!*

<u>Part of Speech Tagging</u>

He	saw	two	birds
↓	↓	↓	↓
PRON	VERB	NUM	NOUN

<u>Lemmatization</u>

He	saw	two	birds
↓	↓	↓	↓
he	see	two	bird

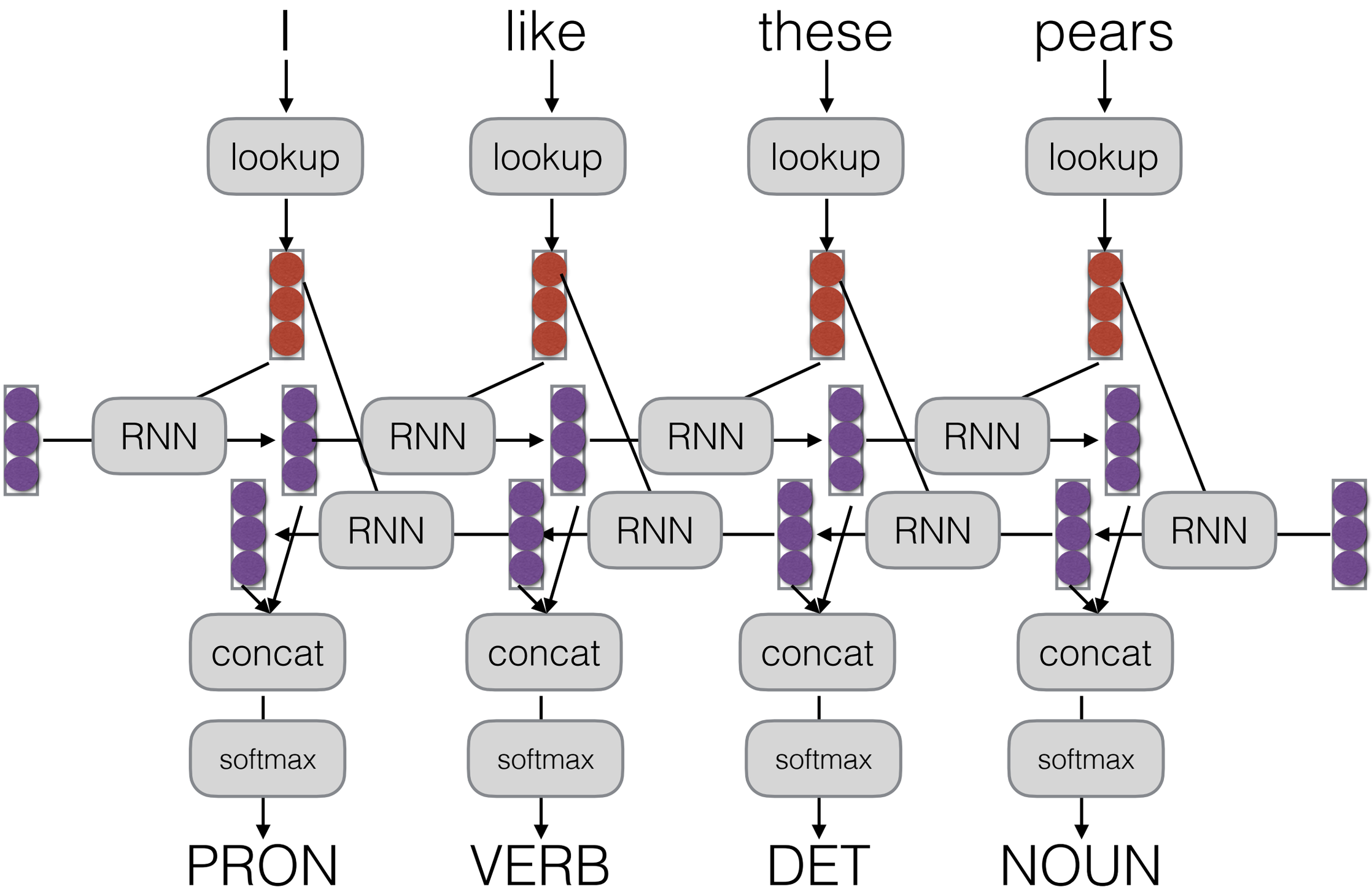
<u>Morphological Tagging</u>

He	saw	two	birds
↓	↓	↓	↓
PronType=prs	Tense=past, VerbForm=fin	NumType=card	Number=plur

... and more!

Reminder: Bi-RNNs

- Simple and standard model for sequence labeling (or classification)



Issues w/ Simple BiRNN

- **RNN model** may be insufficient/inefficient
  - ➡ Alternative models such as bag-of-ngrams, CNNs (this lecture), or transformers
- **Rare words (esp. w/ rich morphology)** are difficult to learn
  - ➡ Character-based word embeddings or subword segmentation
- **Little to no labeled data** exists for many languages
  - ➡ Unsupervised learning or cross-lingual transfer
- **Labels depend on each-other**, and predicting independently is not efficient
  - ➡ Structured prediction algorithms such as CRFs

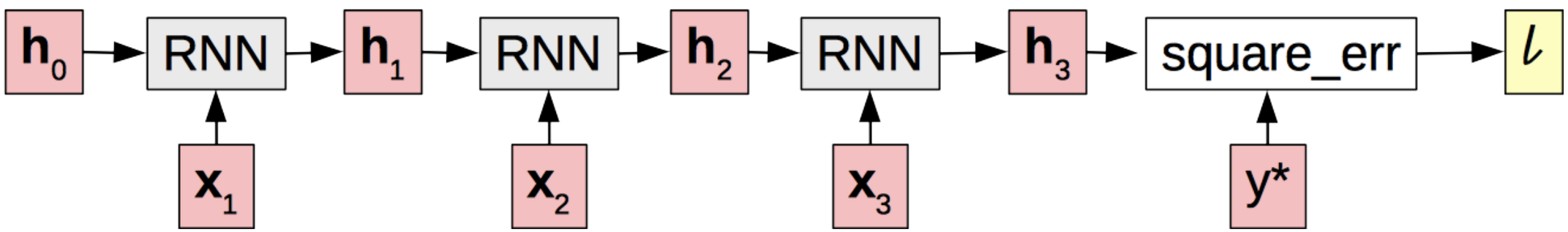
Alternative Sequence Models: Bag-of-ngrams and CNN

(a very brief summary)

Problem w/ RNN: Vanishing Gradient

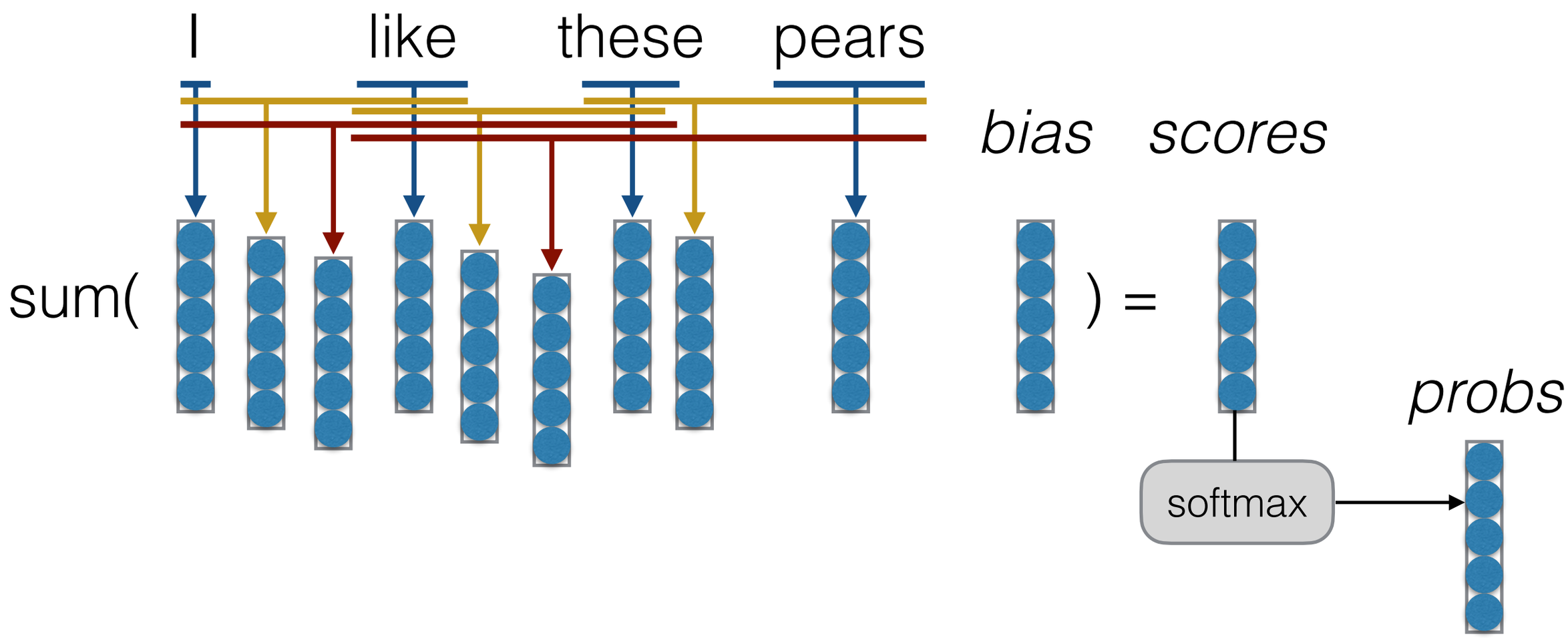
- Gradients decrease as they get pushed back

$$\frac{dl}{d_{h_0}} = \text{tiny} \qquad \frac{dl}{d_{h_1}} = \text{small} \qquad \frac{dl}{d_{h_2}} = \text{med.} \qquad \frac{dl}{d_{h_3}} = \text{large}$$



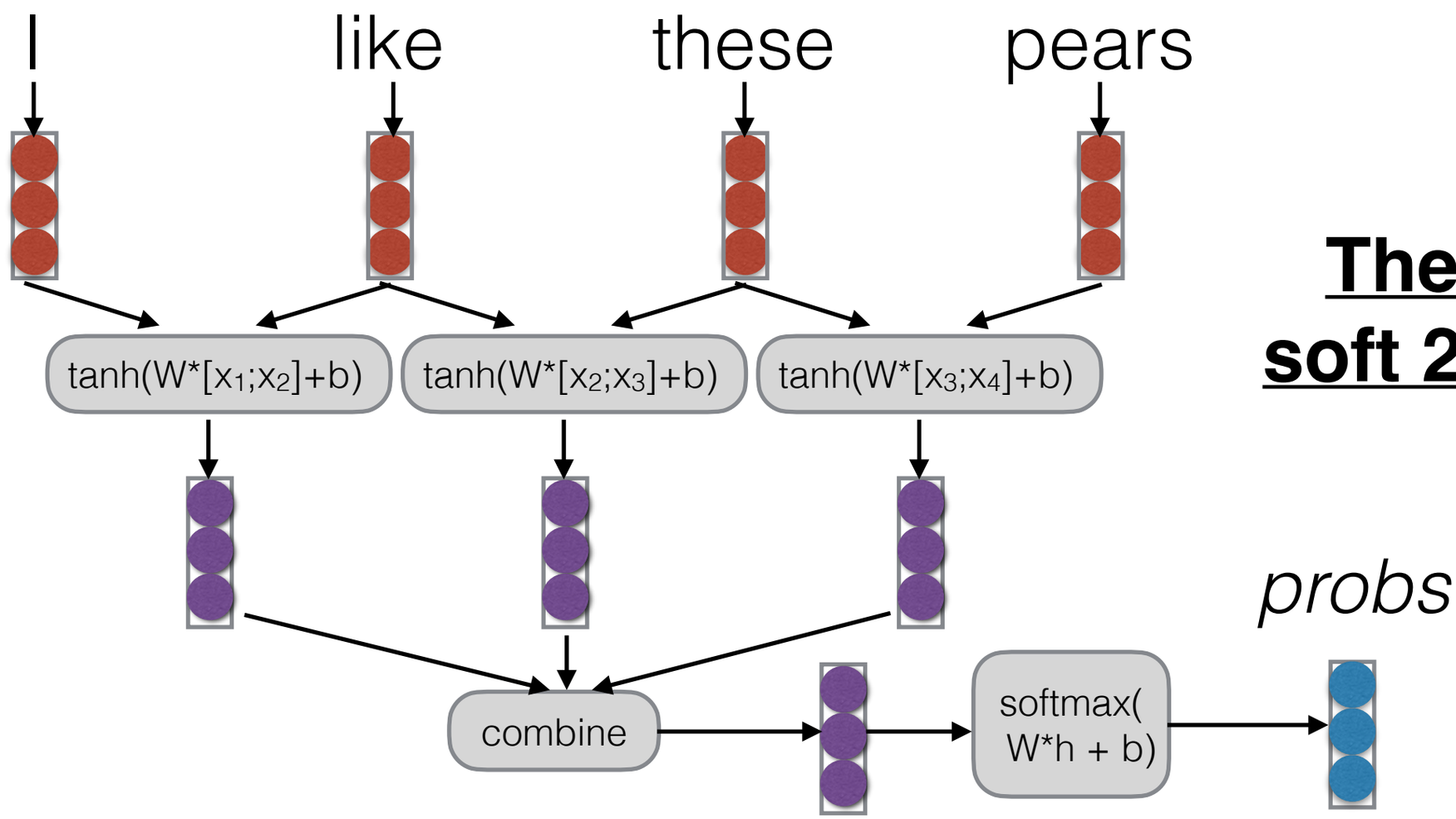
- Why? “Squashed” by non-linearities or small weights in matrices
- Alternative architectures such as LSTMs/GRUs provide solutions, but still indirect

Alternative: Bag of n-grams



- **Advantage:** Much more direct modeling of short text fragments
- **Disadvantage:** Huge number of parameters for large corpora

1-dimensional Convolutions/Time-delay Networks



These are soft 2-grams!

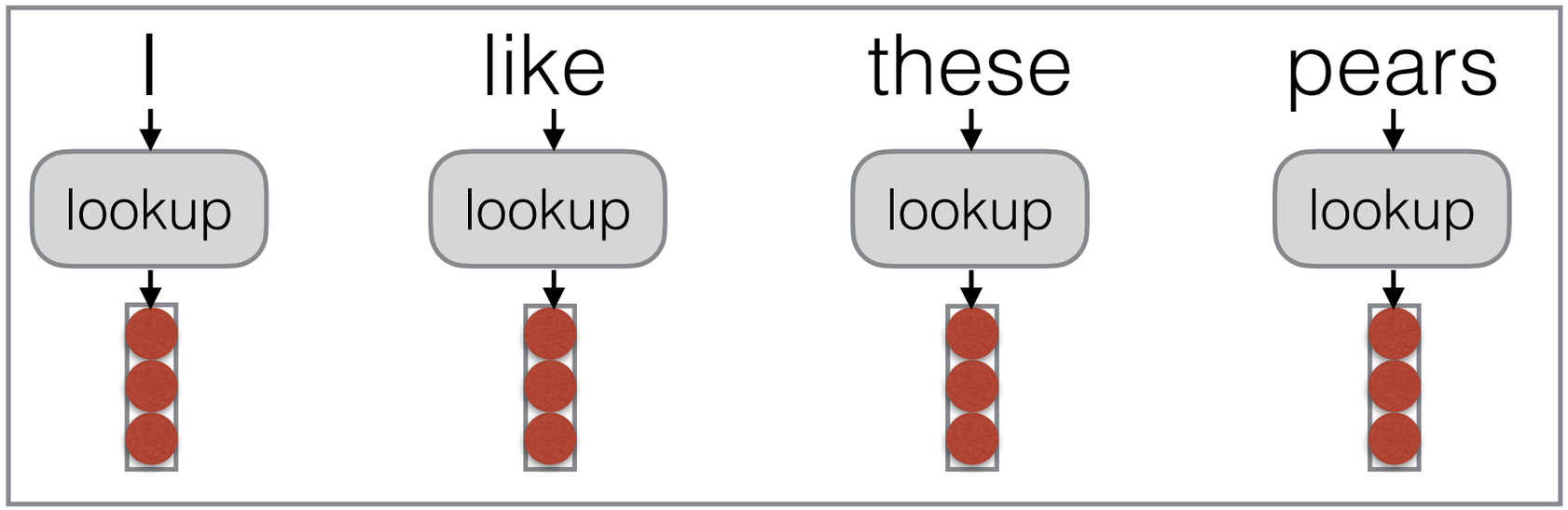
- **Advantage:** Can generalize better than bag-of-ngrams, compose multiple layers together
- **Disadvantage:** Less direct than bag of n-grams when remembering individual n-grams important

Waibel, Alex, et al. "Phoneme recognition using time-delay neural networks." *IEEE transactions on acoustics, speech, and signal processing* 37.3 (1989): 328-339.

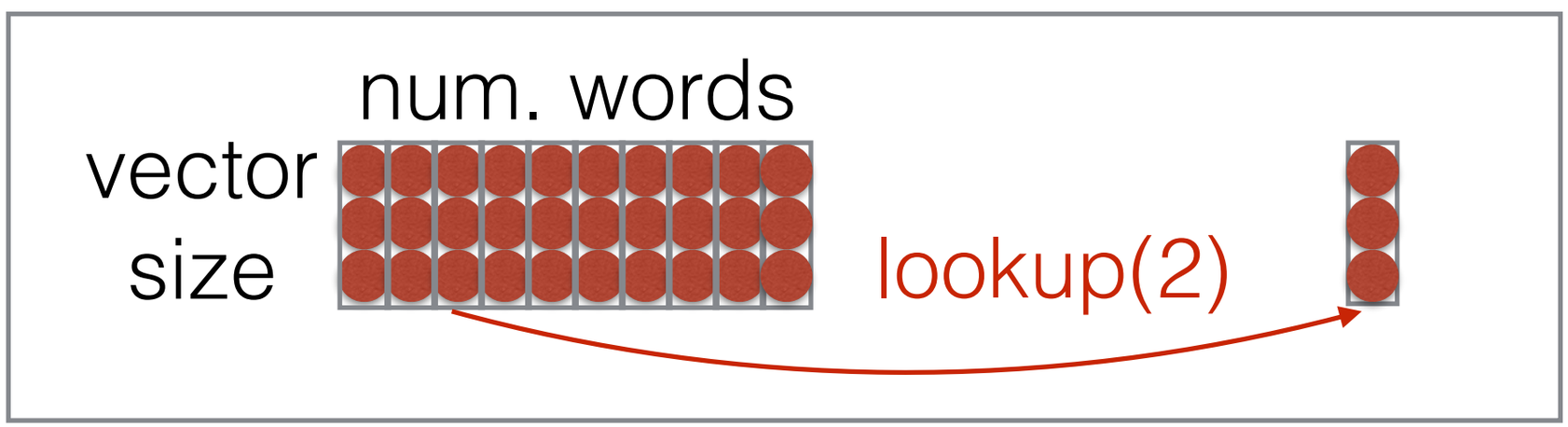
Modeling Vocabulary Items

Converting Words into Vectors

- Each word, independently, gets its own vector rep.



- Standard lookup entails



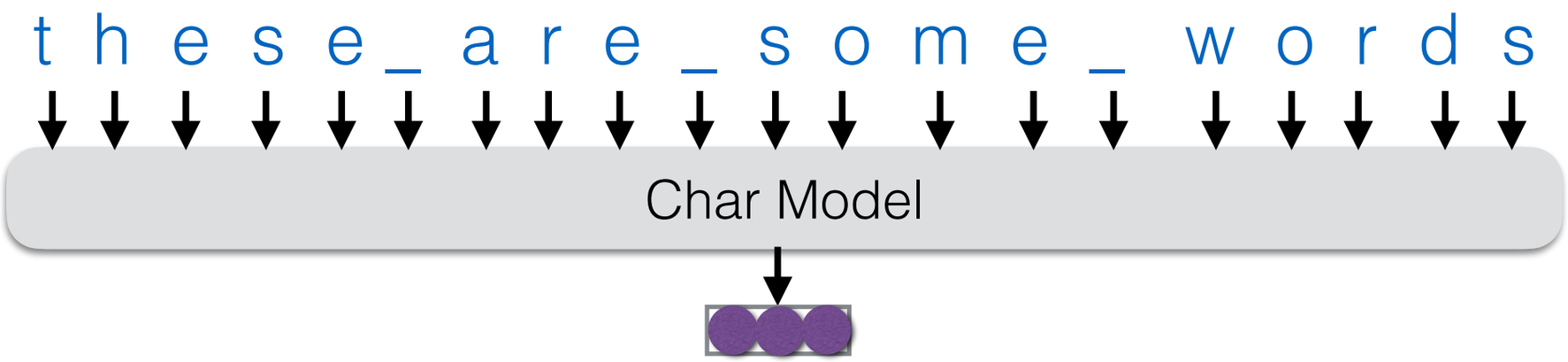
Unknown Words

- Necessity for UNK words
  - We won’t have all the words in the world in training data
  - Larger vocabularies require more memory and computation time
- Common ways:
  - Frequency threshold (usually UNK <= 1)
  - Rank threshold

Solution 1:

Fully Character-based Models

- Basic idea: just encode characters for the entire sentence



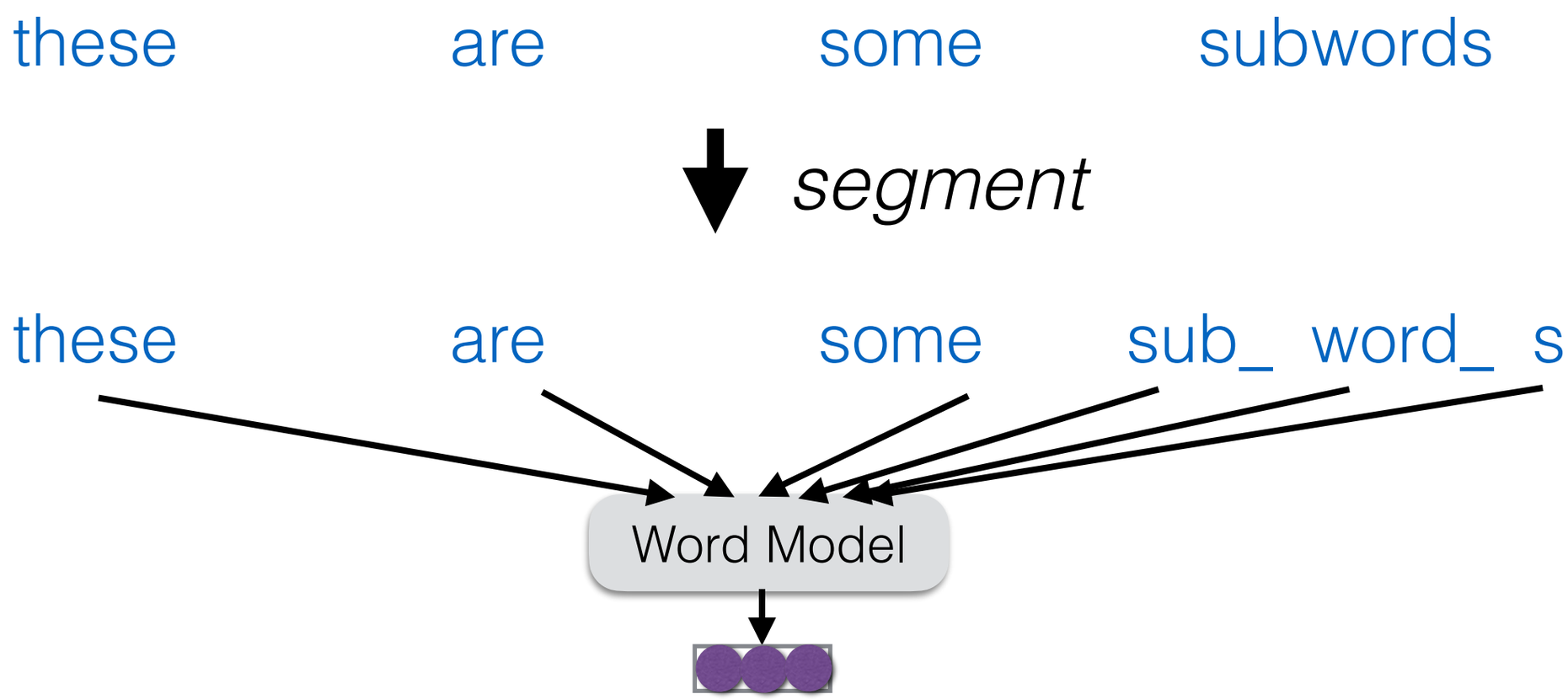
- Issues: slow, ignores natural word boundaries, can be hard to train

Lee et al. "Fully character-level neural machine translation without explicit segmentation." TACL 2017.

Solution 2:

Sub-word Segmentation

- Basic idea: separate rarer words into subwords, then embed



- **Supervised segmentation:** use supervised word segmentation (e.g. Jieba for Chinese, KyTea for Japanese), or morphological analysis
- **Unsupervised segmentation:** use data driven algorithms to discover which units to segment

Unsupervised Subword Segmentation Algorithms

- **Byte pair encoding:**
  - segment into characters,
  - merge most-frequent subword sequence for fixed number of operations
- **Unigram-based segmentation:**
  - create vocabulary of most-frequent character n-grams
  - use EM algorithm to optimize probabilities, remove subwords with low probability
- Both implemented in "sentencepiece" toolkit

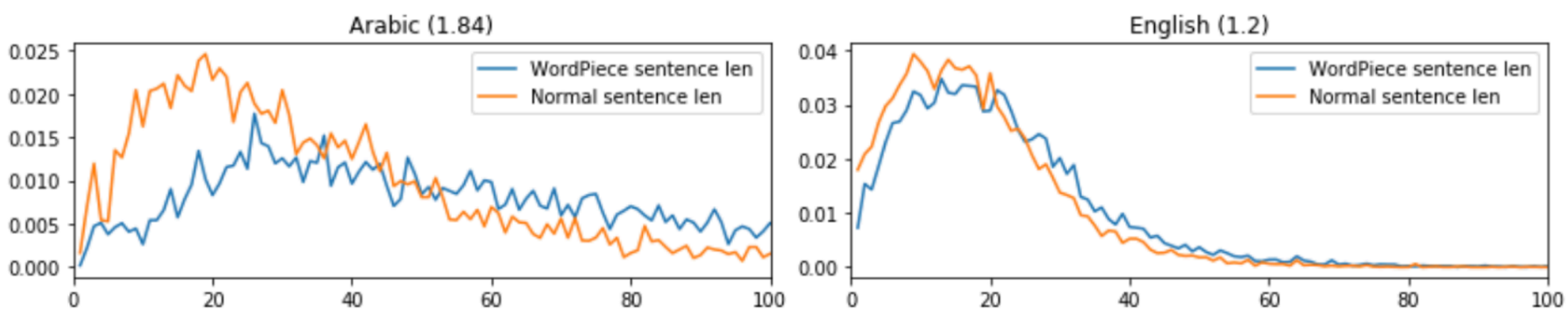
https://github.com/google/sentencepiece

Sennrich et al. "Neural Machine Translation of Rare Words with Subword Units." ACL 2016.

Kudo. "Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates." ACL 2017.

(Multilingual) Subword Segmentation Pros/Cons

- **Pros:** very simple (just pre-processing!), fast, somewhat effectve
- **Cons:**
  - Cannot handle non-concatenative morphology (e.g. goose -> gee_ se)
  - When trained multilingually, few subwords for less-resourced languages -> over-segmentation



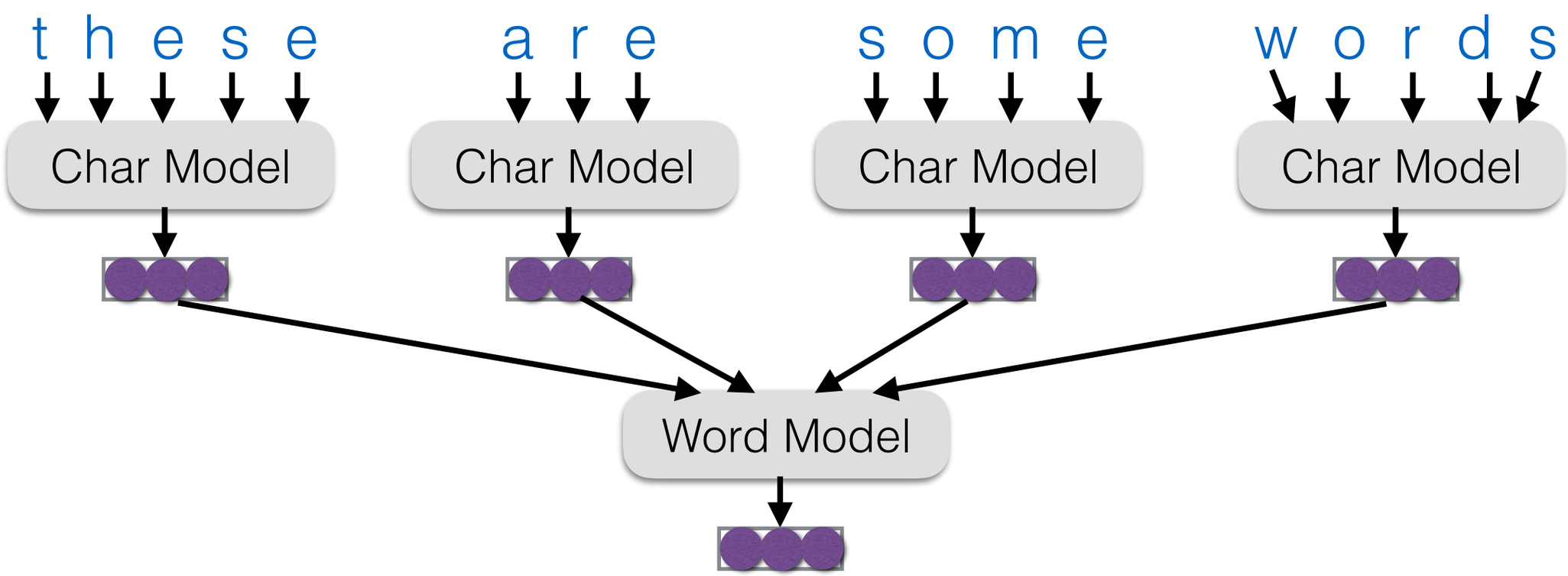
http://juditacs.github.io/2019/02/19/bert-tokenization-stats.html

  - Inconsistent segmentation across languages

Solution 3:

Sub-word Based Embeddings

- Basic idea: two-level embedding
  - Characters -> word embeddings
  - Word -> sentence embeddings



Sub-word Based Embedding Models

- BiRNN/LSTM models

Ling, Wang, et al. "Character-based neural machine translation." *arXiv preprint arXiv:1511.04586* (2015).

- CNN models

Ma, Xuezhe, et al. "End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF" *ACL 2016* (2016).

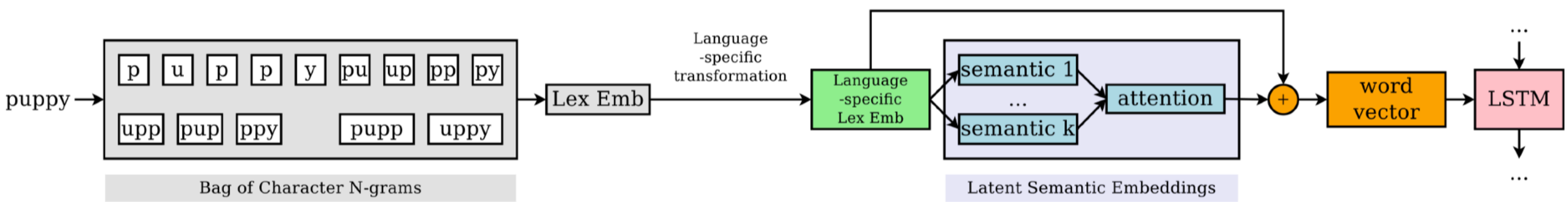
- Bag-of-ngram embedding models

Wieting, John, et al. "Charagram: Embedding Words and Sentences via Character n-grams" *EMNLP 2016* (2016).

- My impression: more direct methods tend to perform better: bag-of-ngram > CNN > BiRNN
- Bag-of-ngrams used in widely-used tools such as fasttext

Embeddings for Cross-lingual Learning: Soft Decoupled Encoding

- Desiderata for cross-lingual lexical learning
  - Spellings are similar (eng "Madrid", spa "Madrid"),
  - with some consistent differences (eng "revolution", spa "revolución"),
  - and some words completely different (eng "hand", spa "mano").



Handles spelling similarity

Handles consistent variations b/t languages

Attempts to capture latent "concepts"

Wang et al. "Multilingual Neural Machine Translation With Soft Decoupled Encoding." *ICLR 2019*.

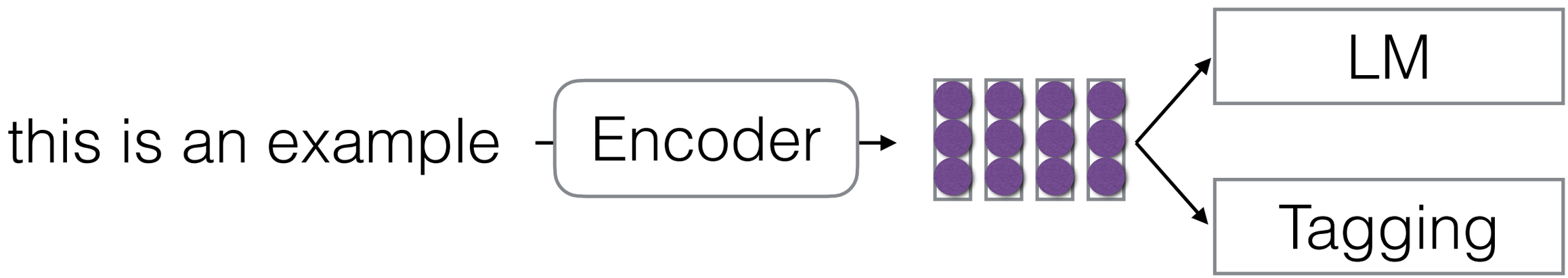
Unsupervised Learning of Representations

Labeled/Unlabeled Data

- **Problem:** we have very little labeled data for most analysis tasks for most languages
- **One Solution:** semi-supervised learning where we also learn from unlabeled data
- Types of semi-supervised learning:
  - Learning of the *neural feature extractor* on a different task (e.g. language modeling)
  - Learning of the *actual output predictor* as well (e.g. form a generative model)
- Former is very easy, latter is more complicated but can be effective

Joint Multi-task Learning

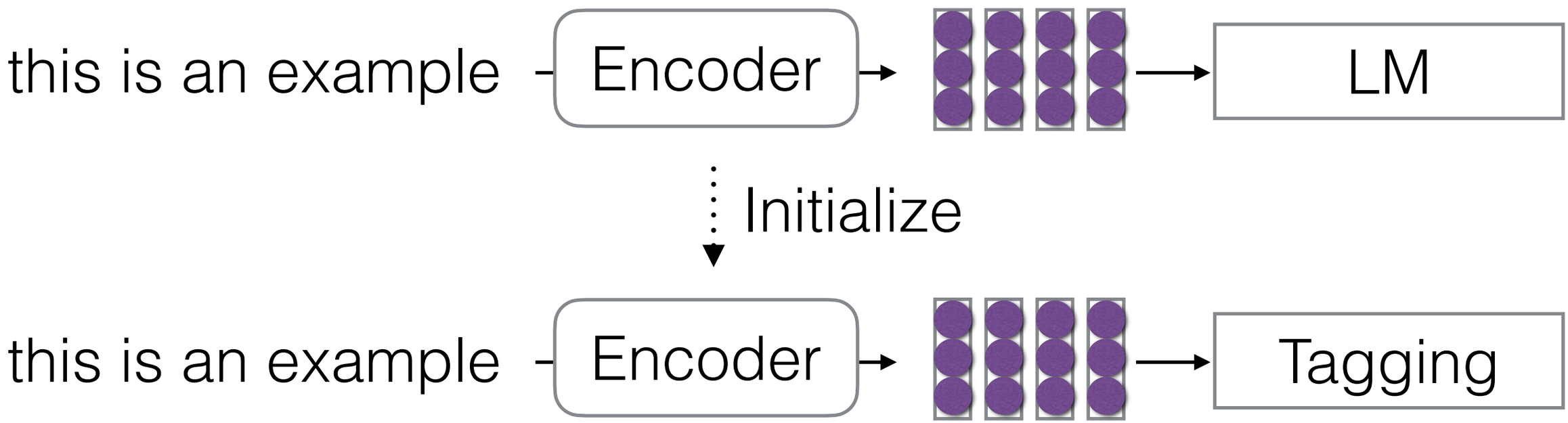
- Train representations to do well on multiple tasks at once



- In general, as simple as randomly choosing minibatch from one of multiple tasks

Pre-training

- First train on one task, then train on another



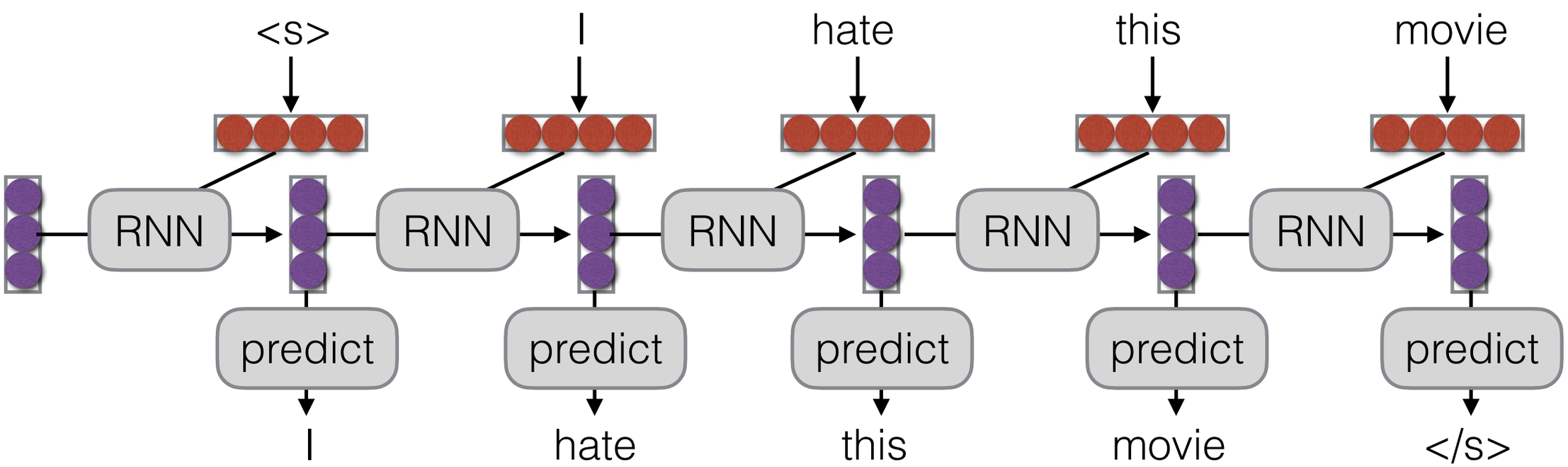
- Often uses language models, masked language models, or other methods

Unsupervised Objective (1): Language Modeling

- Predict the probability of a sequence, e.g. by predicting next word left to right

$$P(X) = \prod_{i=1}^{I} P(\underbrace{x_i}_{\text{Next Word}} \mid \underbrace{x_1, \ldots, x_{i-1}}_{\text{Context}})$$

- Like tagging, where tag is the next word!



- *Disadvantage:* cannot use bidirectional context

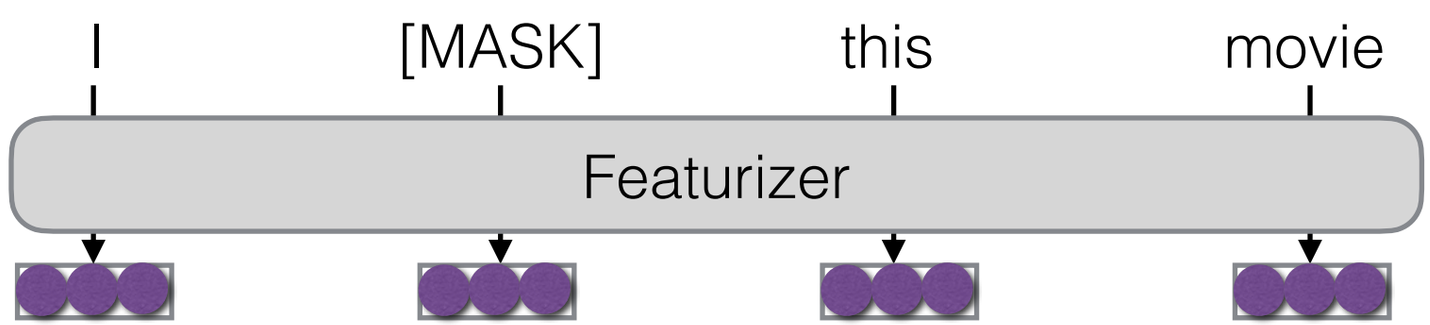
Unsupervised Objective (2): Masked Language Modeling

- Denoising auto-encoding is a task of trying to predict clean data from a noised data

$$P(\underline{X}|\underline{X'})$$

Clean Data Noised Data

- In NLP, masked language modeling a typical example:



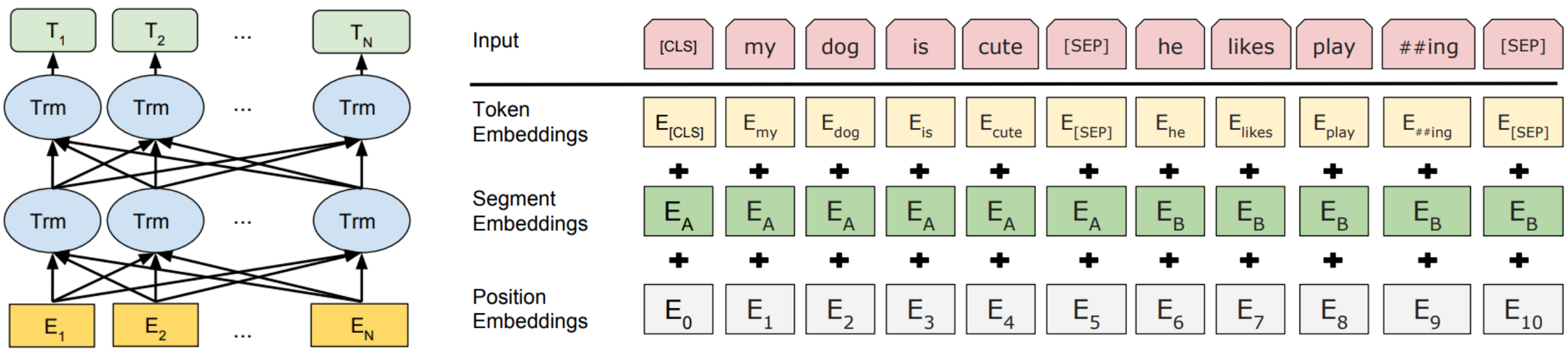
- *Advantage:* can easily use bidirectional context
- *Disadvantage:* is not actually a language model, cannot easily do generation or sequence scoring

Thinking about Multi-tasking, and Pre-trained Representations

- Many methods have names like BERT, mBERT, XLM, XLM-R, Unicoder along with pre-trained models
- These often refer to a combination of
  - **Model:** The underlying neural network architecture
  - **Training Objective:** What objective is used to pre-train
  - **Data:** What data the authors chose to use to train the model
- Remember that these are often conflated (and don't need to be)!

BERT

- **Model:** Multi-layer self-attention. Input sentence or pair, w/ [CLS] token, subword representation



- **Objective:** Masked language modeling + next-sentence prediction
- **Data:** BooksCorpus + English Wikipedia

Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." NAACL 2019.

Other Monolingual BERTs

- BERT has been trained in many languages!
- BERT (en), CamemBERT (fr), BERTje (nl), BETO (es), Ru-BERT (ru), Chinese BERT (zh), BERTurk (tr), and GreekBERT (el)

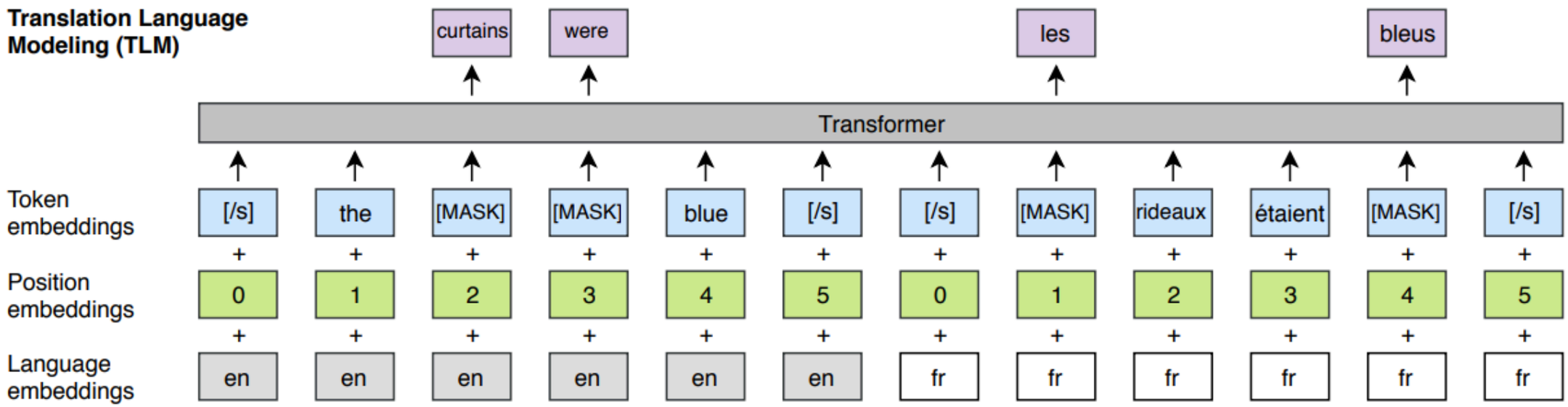
mBERT

- **Model:** Same as BERT, subword vocabulary created slightly upweighting low-resource languages
- **Objective:** Same as BERT
- **Data:** 100 languages worth of Wikipedia
- **Results:** Quite impressive! (for the time), see below paper for details:

Wu and Dredze. "Beto, Bentz, Becas: The Surprising Cross-Lingual Effectiveness of BERT." EMNLP 2018.

XLM

- **Model:** Same as mBERT
- **Objective:** Masked language modeling with parallel sentences:



- **Data:** multilingual Wikipedia (regular MLM), parallel data such as UN/EuroParl/etc. (translation LM)

Lample et al. "Cross-lingual Language Model Pretraining." NeurIPS 2019.

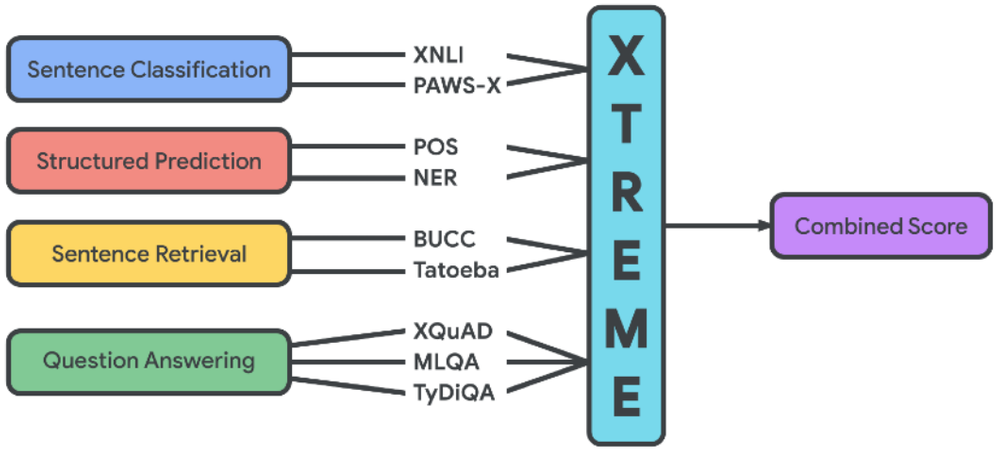
XLM-R

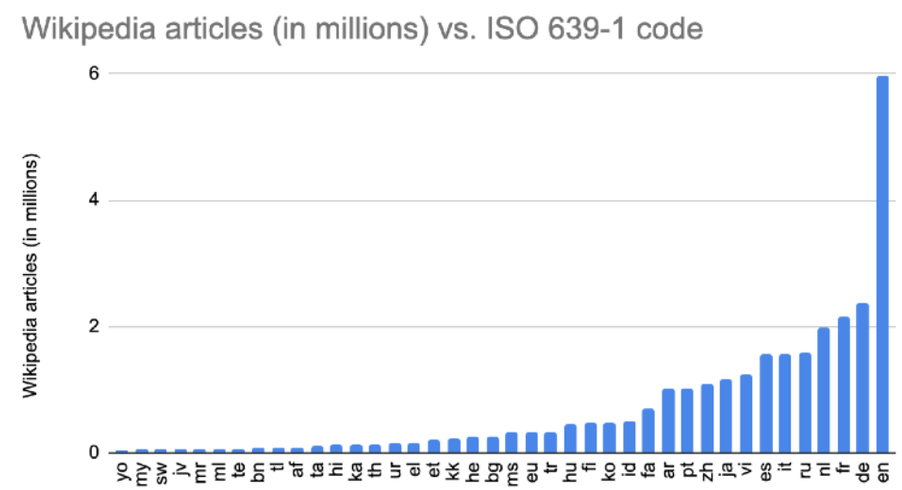
- **Model:** Same as mBERT
- **Objective:** Same as mBERT (not XLM!)
- **Data:** CommonCrawl + Wikipedia

Conneau et al. "Unsupervised Cross-lingual Representation Learning at Scale." ACL 2020.

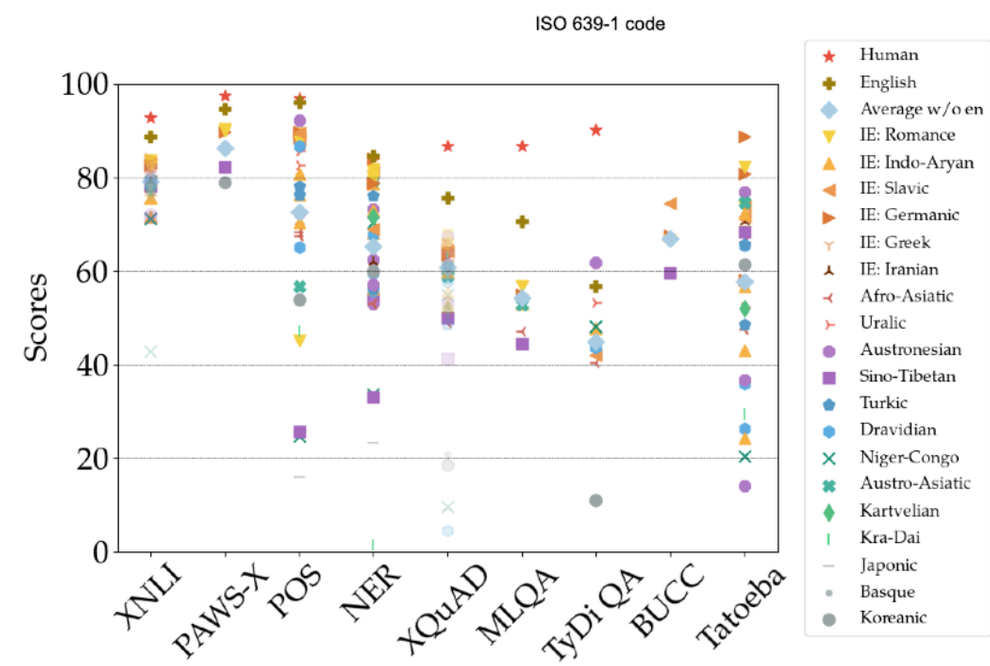
XTREME: Comparing Multilingual Representations

- 10 tasks, 40 languages w/ different data availability





- Many conclusions, e.g.:
  - Success of pre-training+transfer varies greatly by task/language
  - Different models better at different tasks



Hu et al. XTREME: A Massively Multilingual Multi-task Benchmark for Evaluating Cross-lingual Generalization. ICML 2020.

Structured Prediction Models

Why Call it “Structured” Prediction?

- Classes are to numerous to enumerate
- Need some sort of method to exploit the *problem structure* to learn efficiently
- Example of “structure”, the following two outputs are similar:

PRP VBP DT NN
PRP VBP VBP NN

Why Model Interactions in Output?

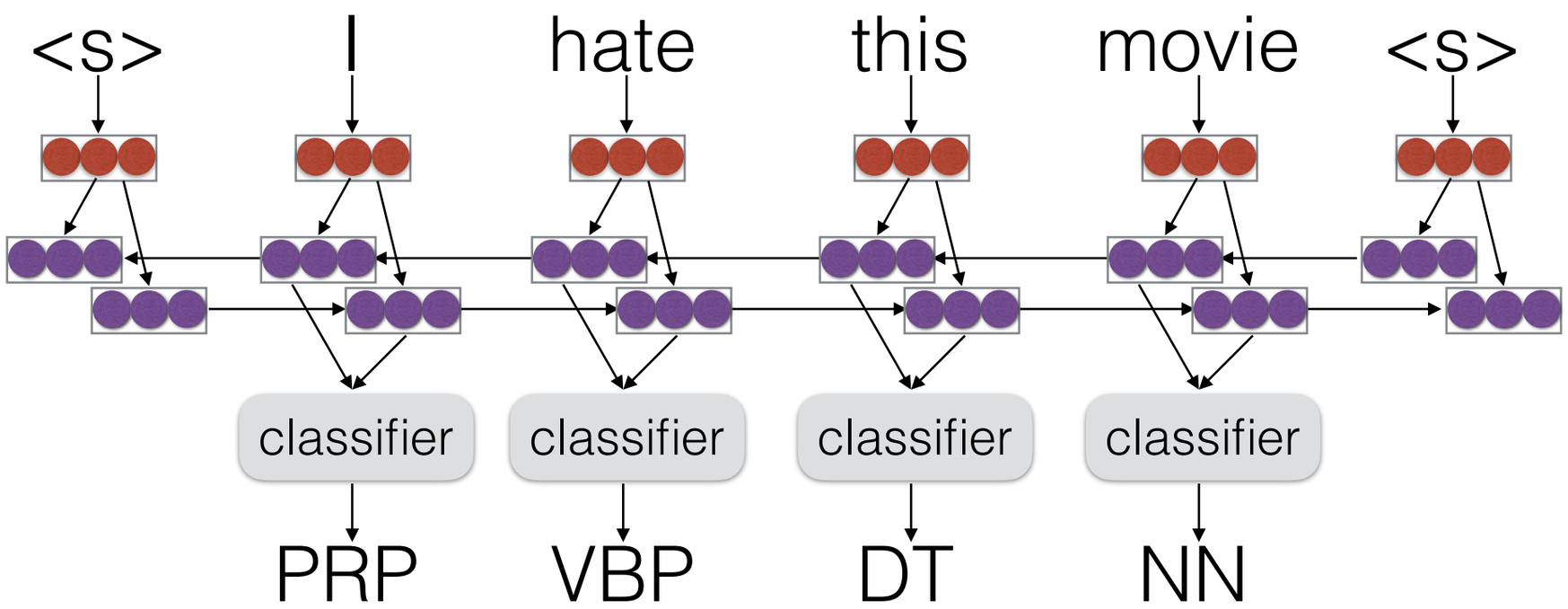
- Consistency is important!

time	flies	like	an	arrow	
NN	VBZ	IN	DT	NN	(time moves similarly to an arrow)
NN	NNS	VB	DT	NN	(“time flies” are fond of arrows)
VB	NNS	IN	DT	NN	(please measure the time of flies similarly to how an arrow would)

↓ max frequency

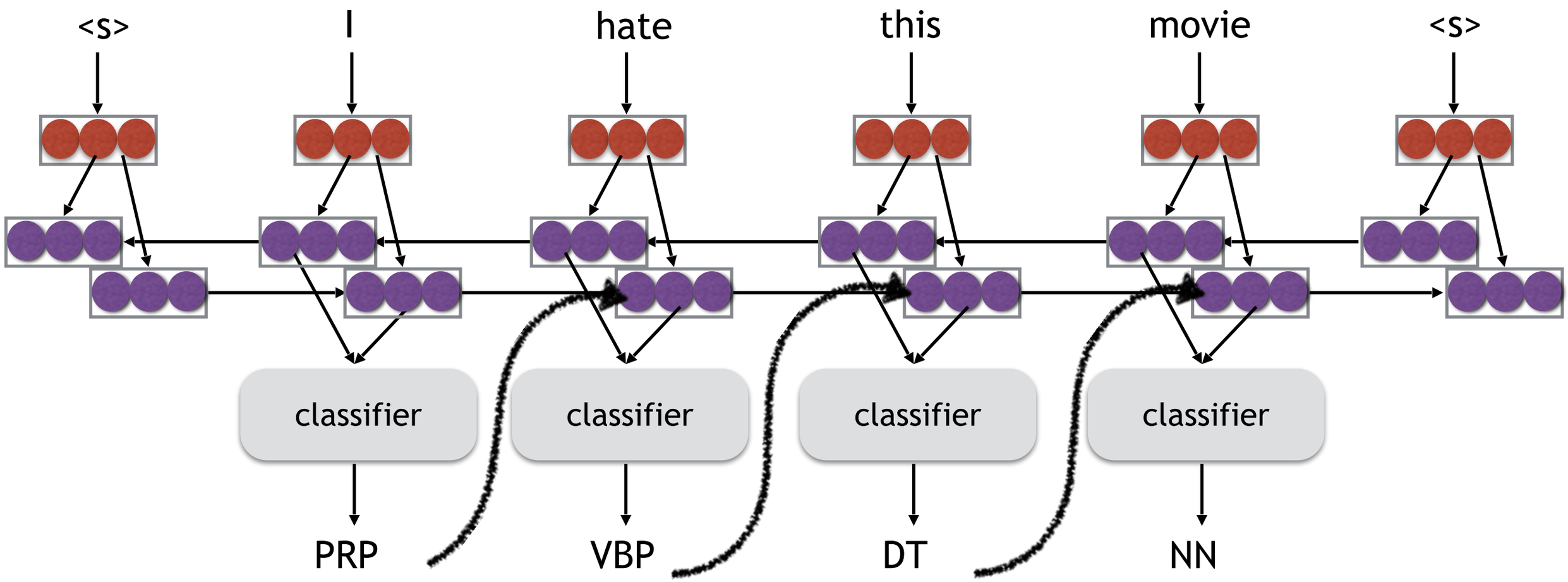
NN	NNS	IN	DT	NN	(“time flies” that are similar to an arrow)

Sequence Labeling w/ BiRNNs



- Still not modeling output structure! Outputs are independent

Recurrent Decoder



Local Normalization vs. Global Normalization

- **Locally normalized models:** each decision made by the model has a probability that adds to one

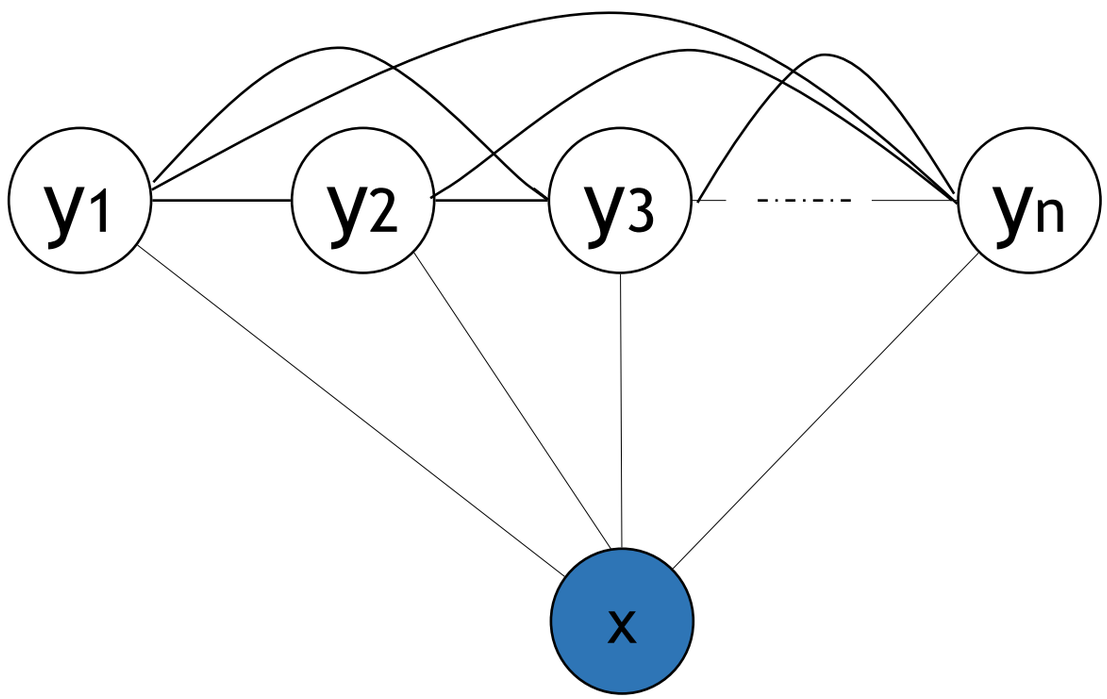
$$P(Y \mid X) = \prod_{j=1}^{|Y|} \frac{e^{S(y_j|X,y_1,\ldots,y_{j-1})}}{\sum_{\tilde{y}_j \in V} e^{S(\tilde{y}_j|X,y_1,\ldots,y_{j-1})}}$$

- **Globally normalized models (a.k.a. energy-based models):** each sequence has a score, which is not normalized over a particular decision

$$P(Y \mid X) = \frac{e^{\sum_{j=1}^{|Y|} S(y_j|X,y_1,\ldots,y_{j-1})}}{\sum_{\tilde{Y} \in V*} e^{\sum_{j=1}^{|\tilde{Y}|} S(\tilde{y}_j|X,\tilde{y}_1,\ldots,\tilde{y}_{j-1})}}$$

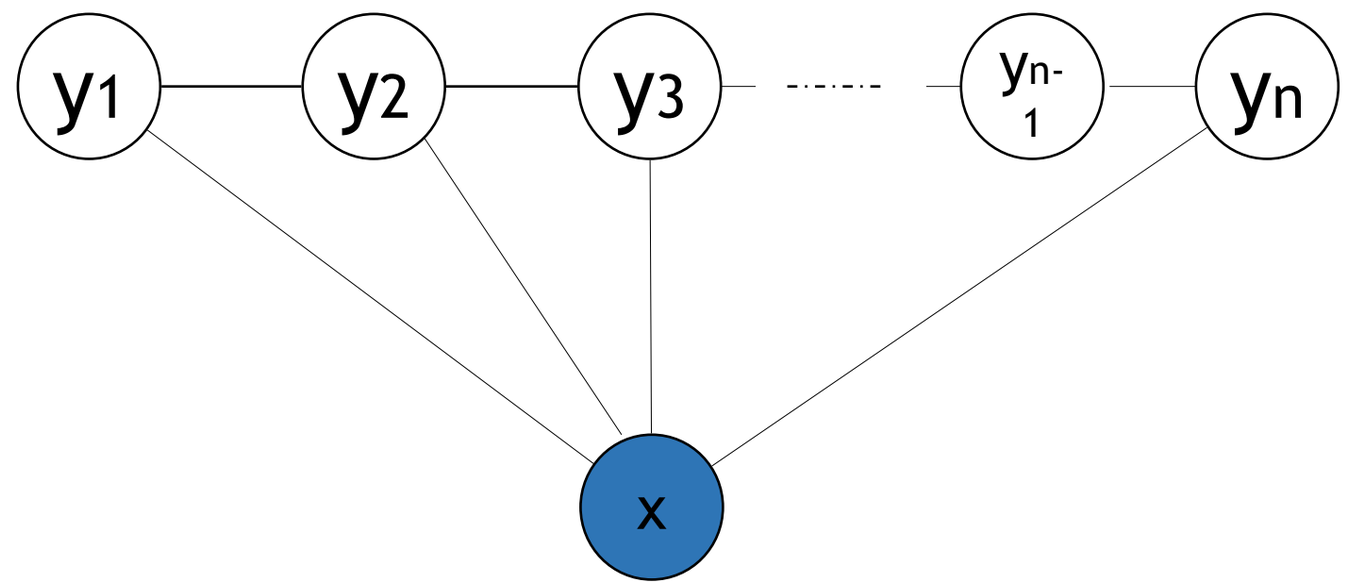
Conditional Random Fields

General form of globally normalized model



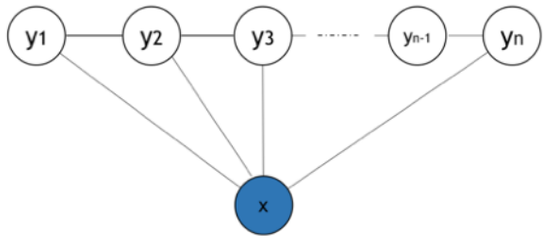
$$P(Y|X) = \frac{\psi(Y,X)}{\sum_{Y'} \psi(Y',X)}$$

First-order linear CRF



$$P(Y|X) = \frac{\prod_{i=1}^{L} \psi_i(y_{i-1}, y_i, X)}{\sum_{Y'} \prod_{i=1}^{L} \psi_i(y'_{i-1}, y'_i, X)}$$

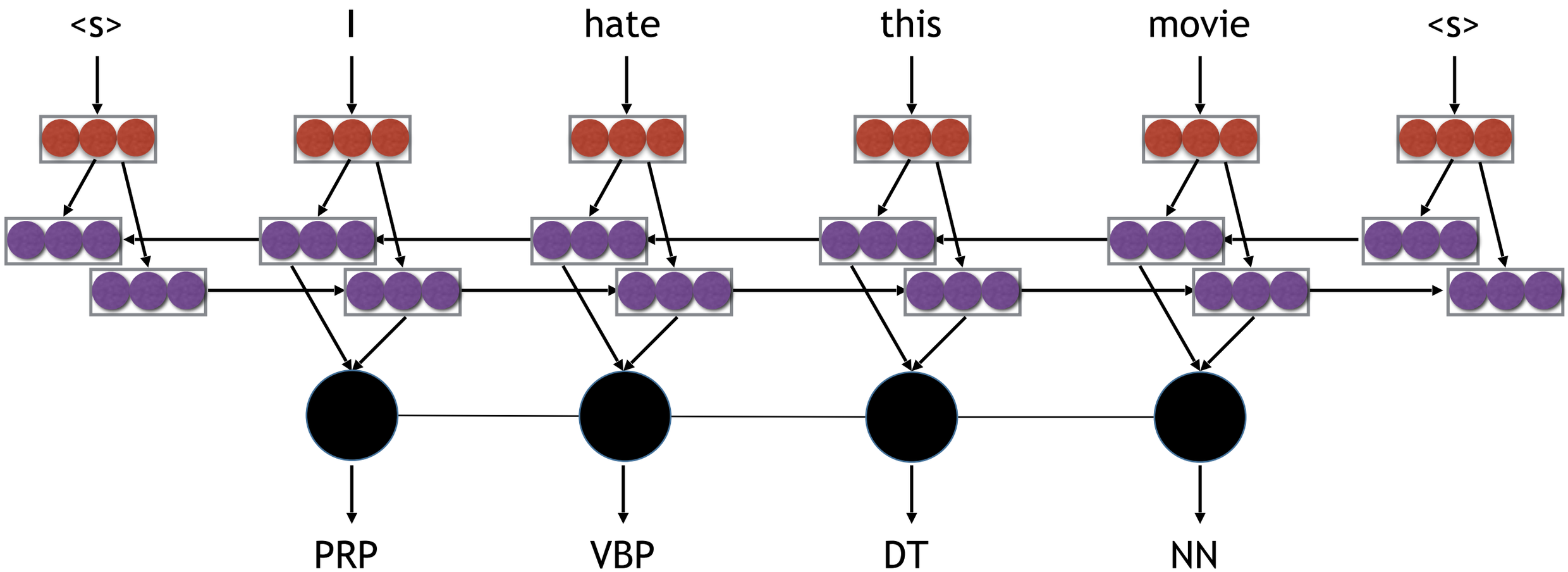
Potential Functions



"Transition" "Emission"

- $\psi_i(y_{i-1}, y_i, X) = \exp\left(\boxed{W^T T(y_{i-1}, y_i, X, i)} + \boxed{U^T\ S(y_i, X, i)} + b_{y_{i-1}, y_i}\right)$

BiLSTM-CRF for Sequence Labeling



Huang et al. "Bidirectional LSTM-CRF Models for Sequence Tagging." arXiv 2015.

CRF Training & Decoding

- $P(Y|X) = \frac{\prod_{i=1}^{L} \psi_i(y_{i-1}, y_i, X)}{\sum_{Y'} \prod_{i=1}^{L} \psi_i(y'_{i-1}, y'_i, X)} = \frac{\boxed{\prod_{i=1}^{L} \psi_i(y_{i-1}, y_i, X)}}{\boxed{Z(X)}}$

Easy to compute

Hard to compute

Go through the output space of Y which grows exponentially with the length of the input sequence using dynamic programming.

Discussion Questions

Discussion

- We have covered three topics in improving sequence models. Read *at least one* of the following papers:
  - Multilingual Neural Machine Translation With Soft Decoupled Encoding (Topic: Subword Models)
  - Cross-lingual Language Model Pretraining (Topic: Unsupervised Learning of Representations)
  - XTREME: A Massively Multilingual Multi-task Benchmark for Evaluating Cross-lingual Generalization (Topic: Unsupervised Learning of Representations)
  - Bidirectional LSTM-CRF Models for Sequence Tagging (Topic: Structured Prediction Models)
- For the paper you read, what is one problem with a standard sequence labeling model that the methods presented or discussed therein resolve? What is one problem that you think that the discussed methods would still *not* be able to resolve?

Thank You!