NATURAL LANGUAGE PROCESSING WITH DEEP LEARNING

Dr.Minaei, IUST, Fall 2020

Presenting by Sara Rajaei

Most materials of this mini-course are provided by "Natural Language Processing" course, Mohammad Taher Pilehvar, IUST, Fall 2019 and "Natural Language Processing with Deep Learning" course, Stanford, Winter 2020

REVIEW – ONE-HOT ENCODING

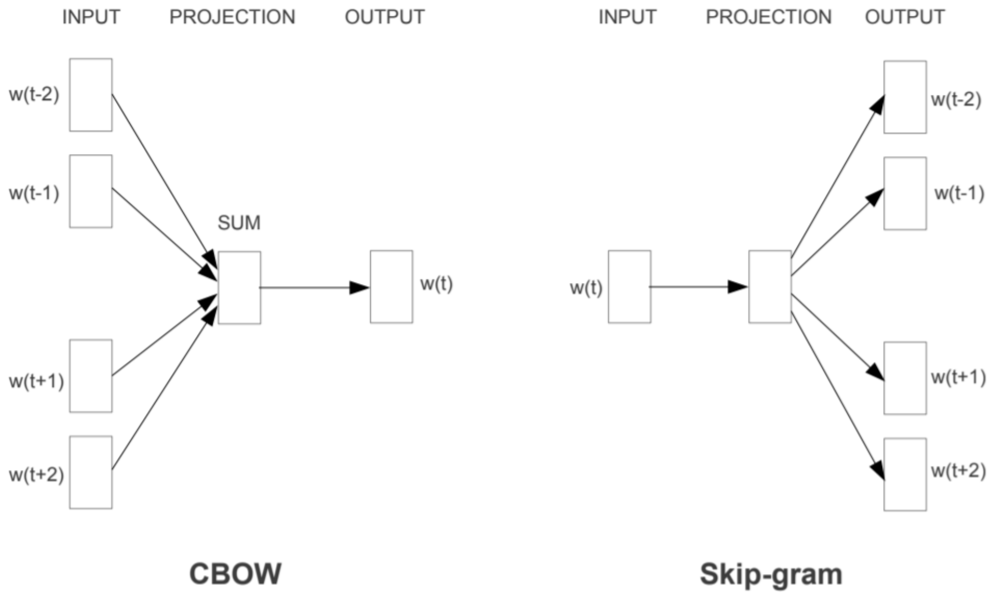
Representing each token with an unique index, and then turning this index to a vector of size N.

The biggest box

	1	2	3
The	1	0	0
biggest	0	1	0
box	0	0	1

REVIEW – WORD2VEC

- A two-layer neural network learns a vector for every word in the vocabulary.
- The target task on which word2vec's model is trained is Language modeling.
- Word2vec has two main method:
  - Continues bag of words(CBOW)
  - Skip-gram

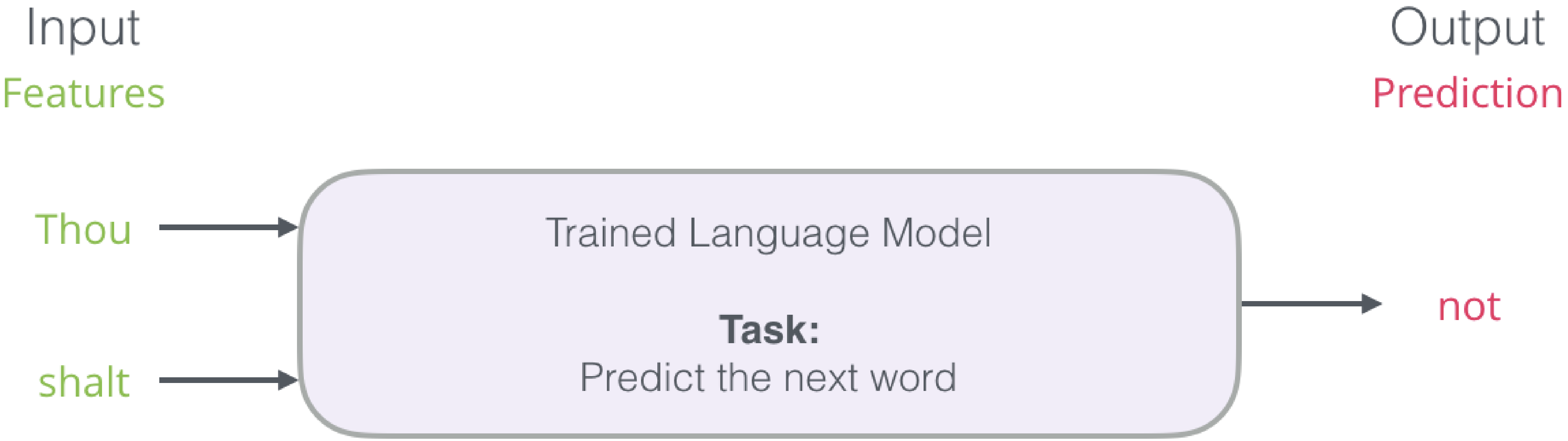


WORD2VEC

MORE DETAILS

input/feature #1 input/feature #2 output/label

Thou shalt ______



Input

Features

Thou

shalt

Trained Language Model

Task:

Predict the next word

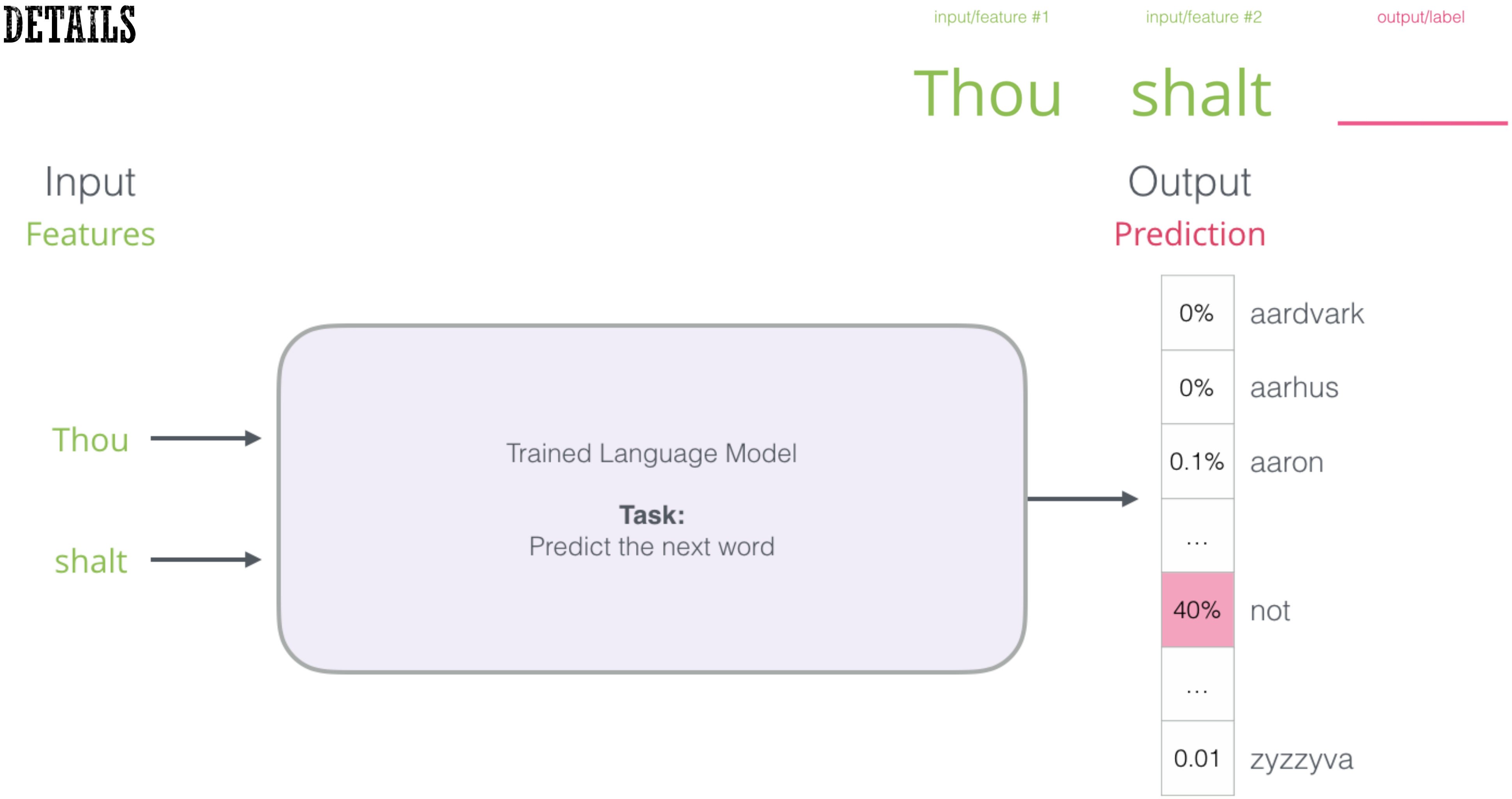
Output

Prediction

not

WORD2VEC

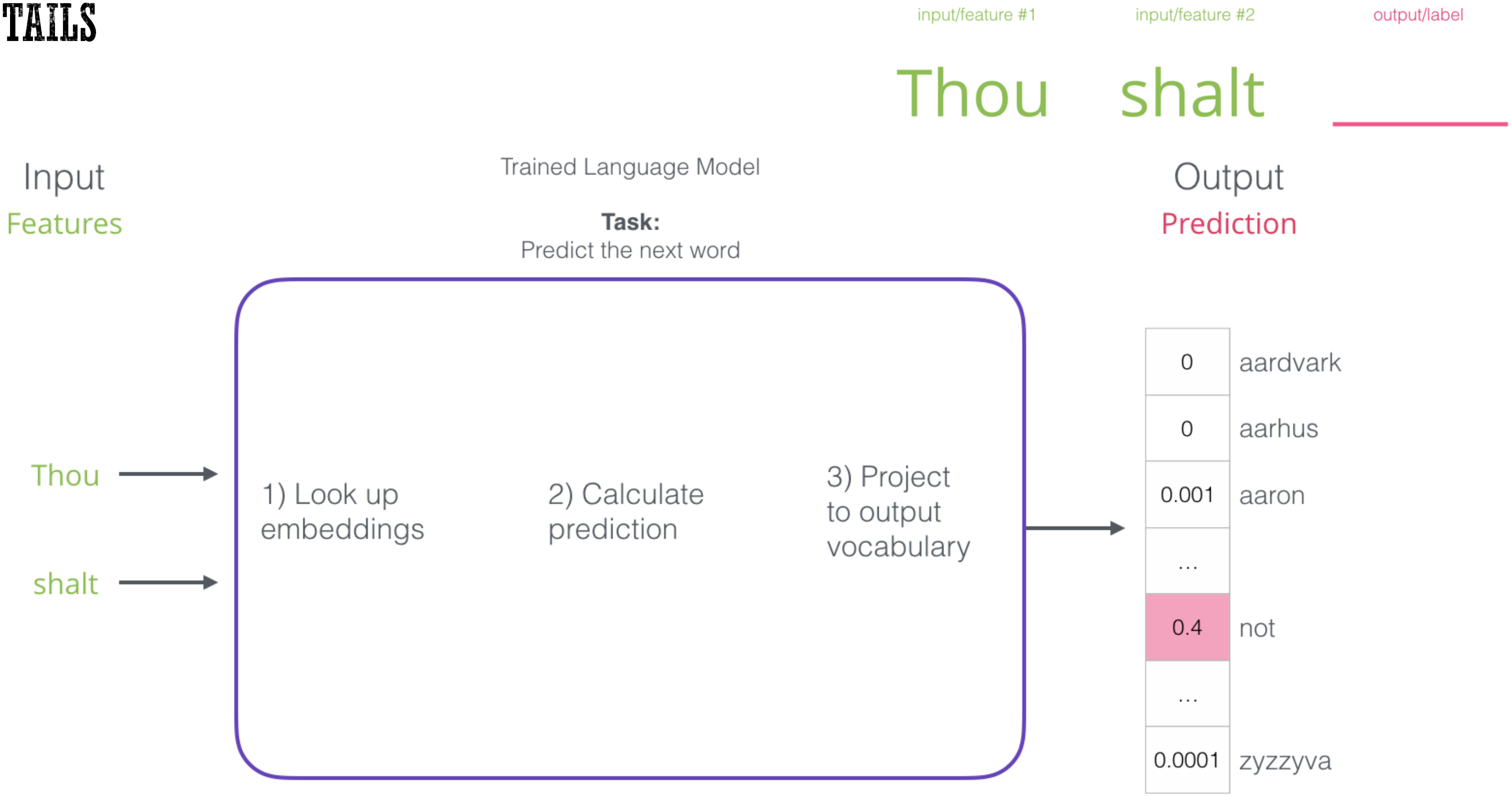
MORE DETAILS



0%	aardvark
0%	aarhus
0.1%	aaron
…	
40%	not
…	
0.01	zyzzyva

WORD2VEC

MORE DETAILS



input/feature #1 input/feature #2 output/label

Thou shalt ______

Input

Features

Trained Language Model

Task:
Predict the next word

Output

Prediction

Thou

shalt

1) Look up embeddings

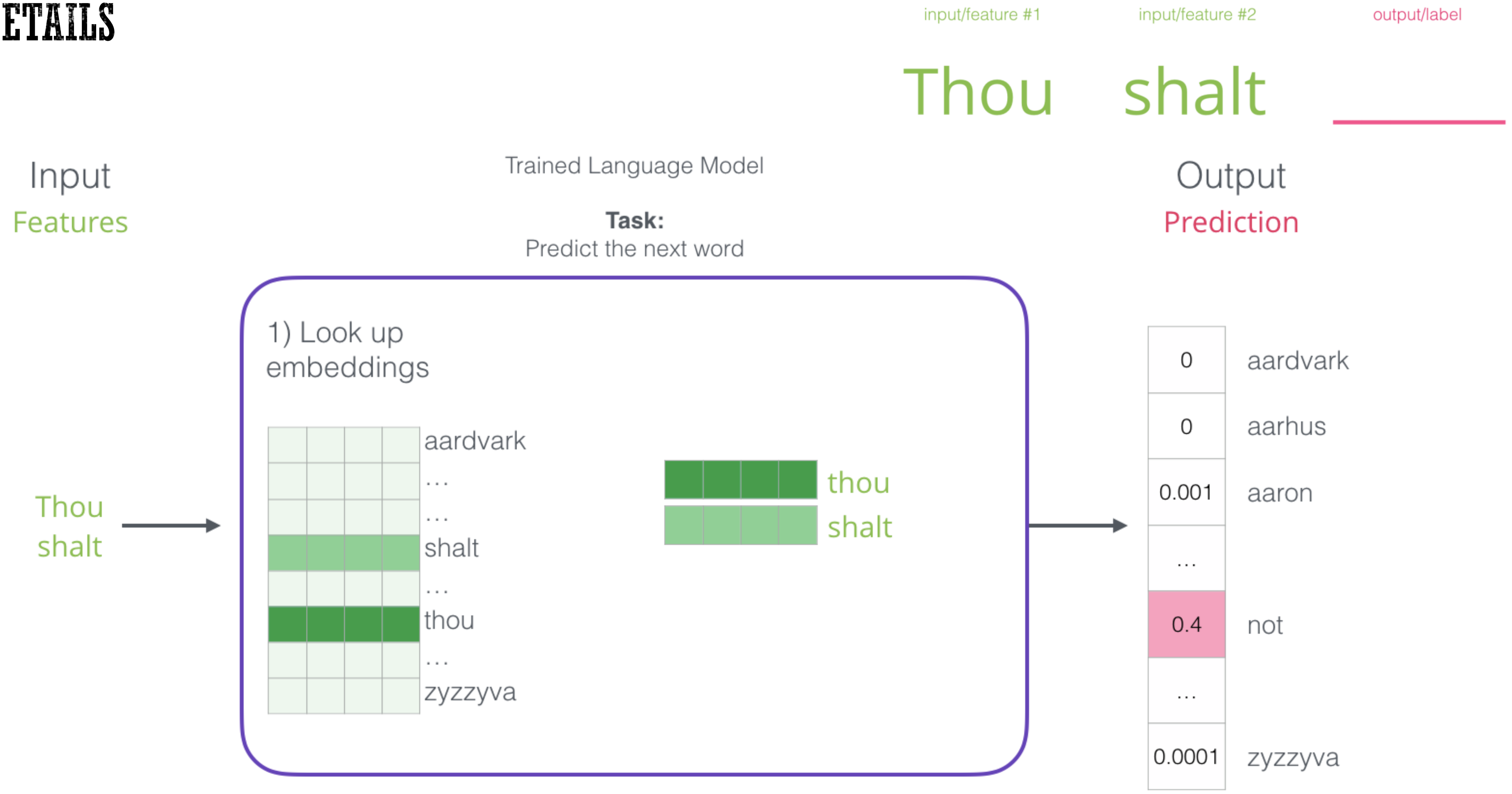
2) Calculate prediction

3) Project to output vocabulary

0	aardvark
0	aarhus
0.001	aaron
…	
0.4	not
…	
0.0001	zyzzyva

WORD2VEC

MORE DETAILS



WORD2VEC
CBOW AND SKIP-GRAM

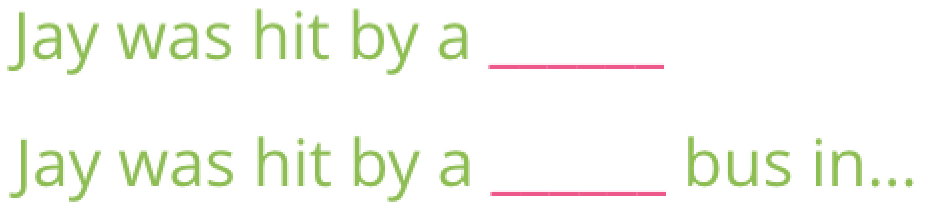
- Instead of looking unidirectionally at the context(language model), we can look at the context bidirectionally(CBOW)

Jay was hit by a ______

Jay was hit by a ______ bus

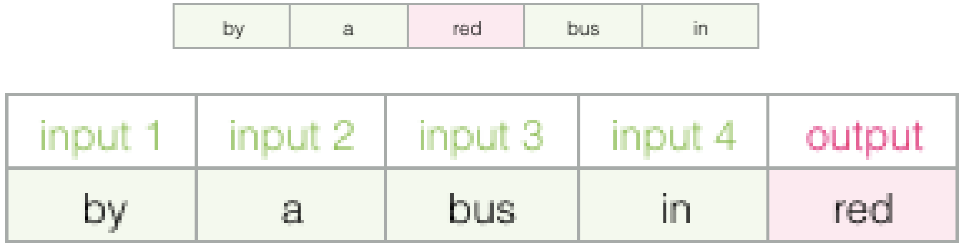
WORD2VEC
CBOW AND SKIP-GRAM

- Instead of looking unidirectionally at the context(language model), we can look at the context bidirectionally(CBOW)



Jay was hit by a ______

Jay was hit by a ______ bus in...



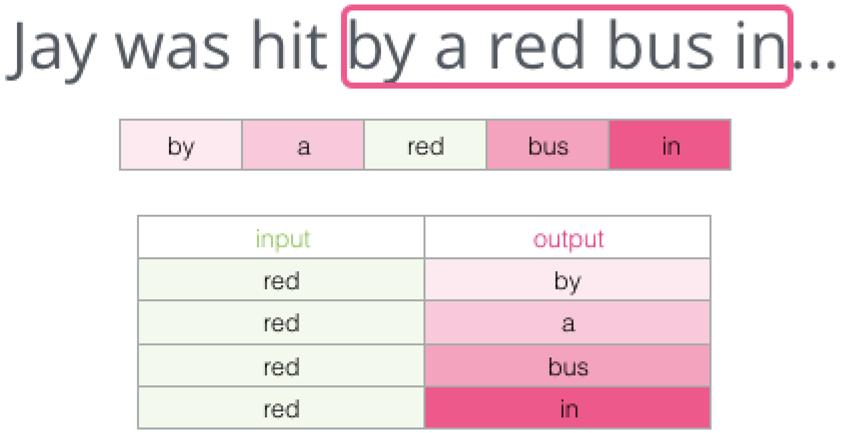
by	a	red	bus	in

input 1	input 2	input 3	input 4	output
by	a	bus	in	red

WORD2VEC
CBOW AND SKIP-GRAM

- Instead of looking unidirectionally at the context(language model), we can look at the context bidirectionally(CBOW)

- Instead of guessing a word based on its context (the words before and after it), this other architecture tries to guess neighboring words using the current word.



Jay was hit by a red bus in...

by	a	red	bus	in

input	output
red	by
red	a
red	bus
red	in

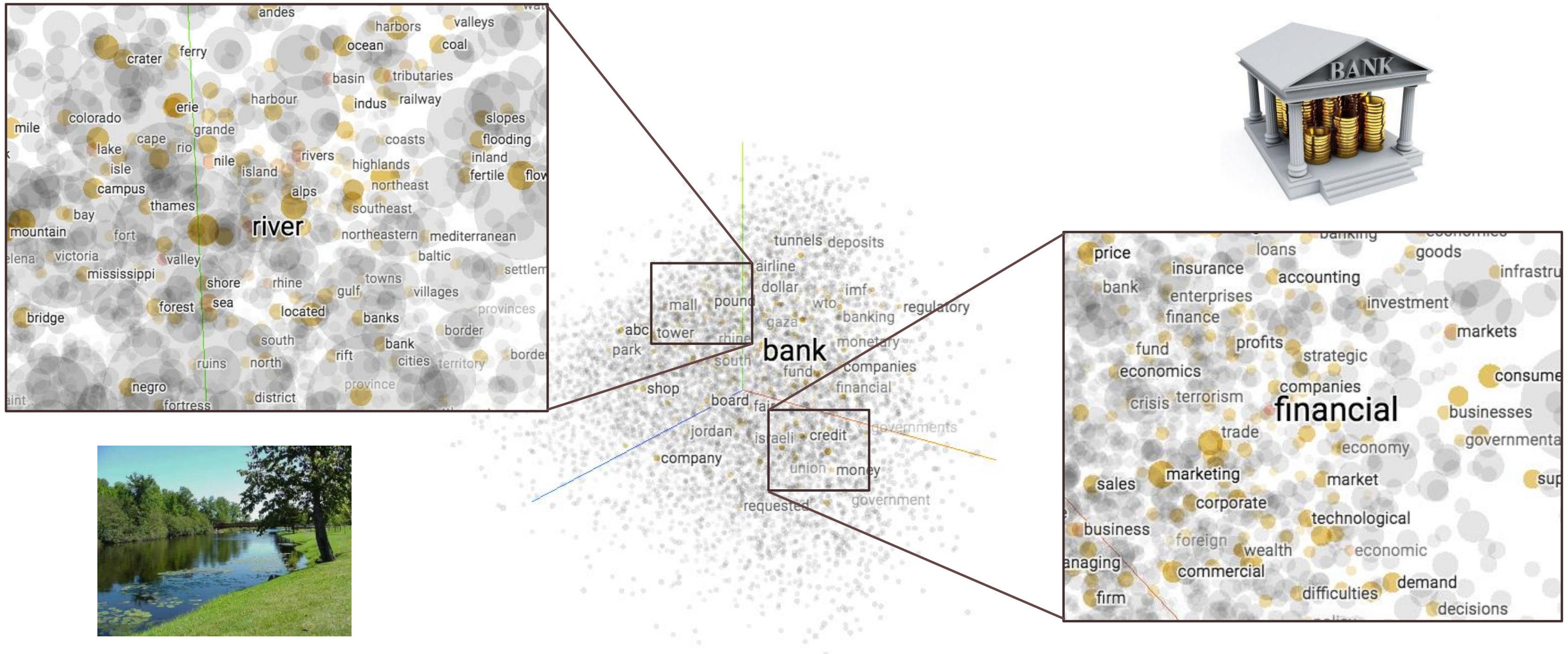
STATIC EMBEDDING

- Word2Vec and Glove are a static embedding.
- You can consider a static embedding like a dictionary.

  Every word of vocabulary has a vector.
- What’s the problem?
  - The vocabulary is finite.
  - Ignoring the role of context in triggering specific meanings of words (Polysemy).
  - Due to restricting the semantic barriers to individual words, it is difficult for the model to capture higher order semantic phenomena

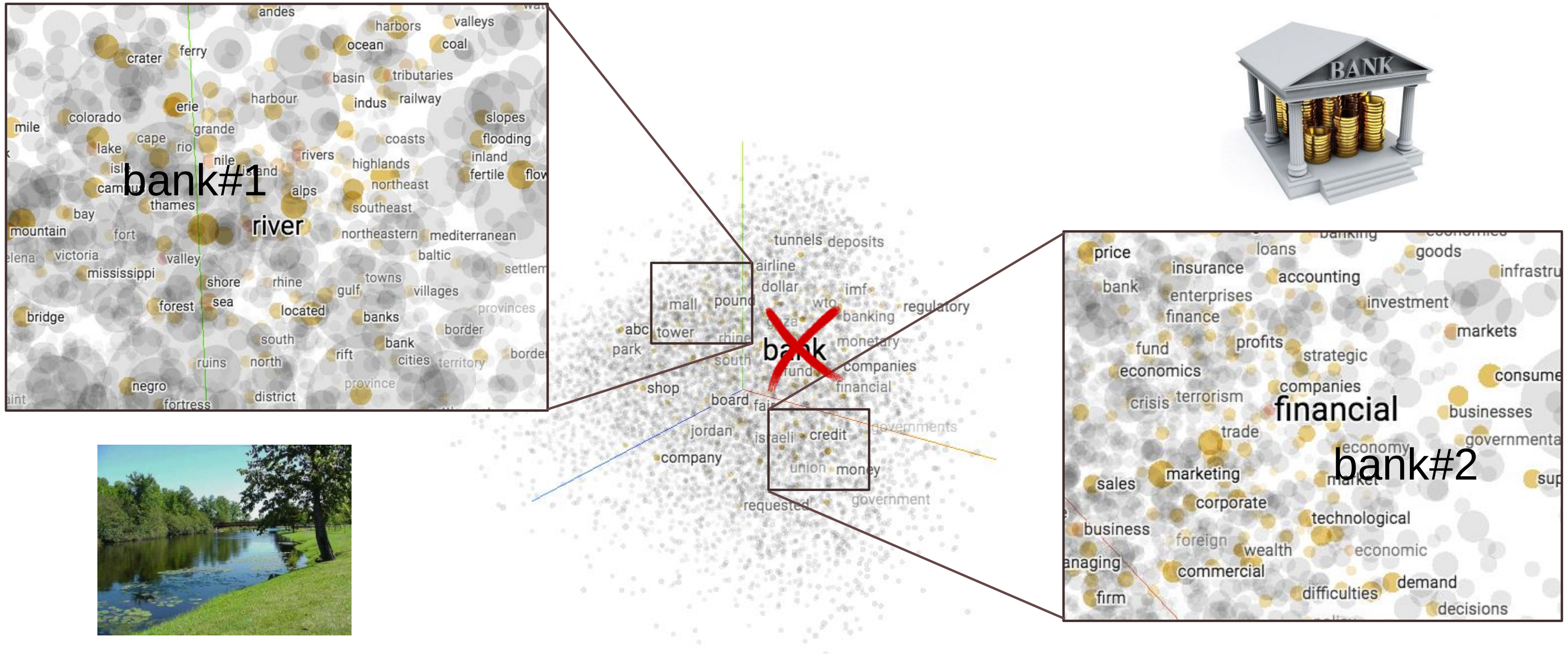
MEANING CONFLATION DEFICIENCY

Word representations **conflate different meanings** of a word into a single representation: they cannot capture **polysemy**



WORD SENSE REPRESENTATIONS

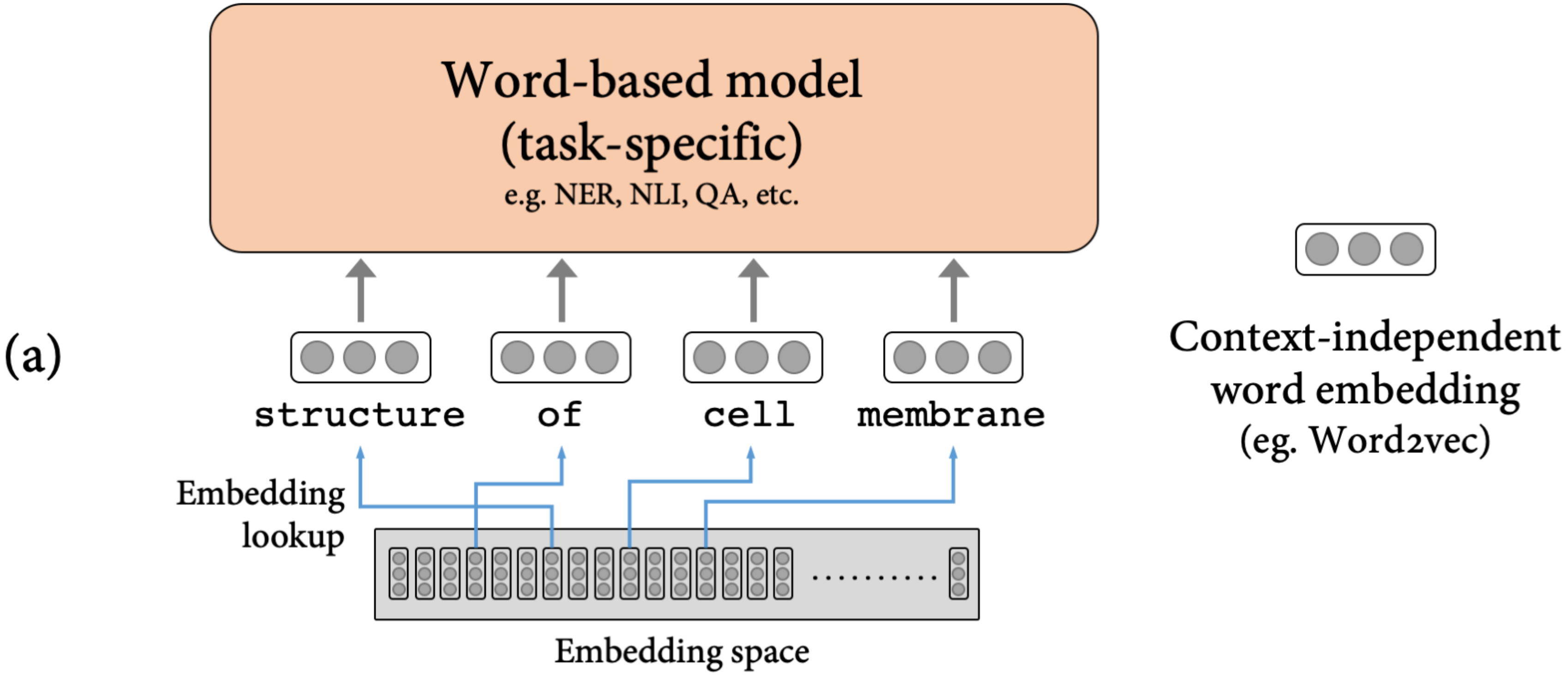
One can address this deficiency by representing **individual meanings of words (word senses)**



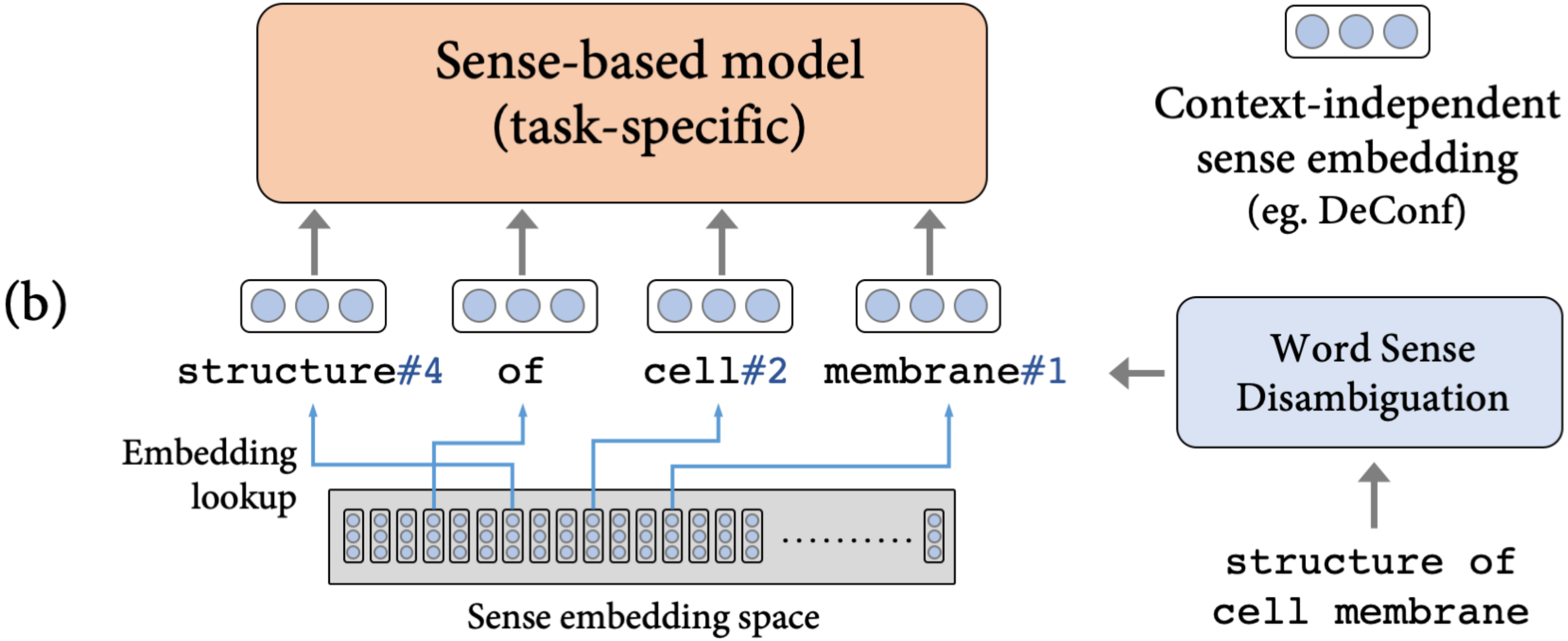
WORD SENSE REPRESENTATIONS

- What are the disadvantages of the sense representations method?
  1. The model needs to carry an additional step, a word sense disambiguation module has to identify the intended meaning of ambiguous words.
  2. word sense disambiguation is far from being optimal; the initial stage of mapping words to word senses introduces inevitable noise to the pipeline
  3. It is not straightforward to benefit from raw texts, which are available at scale, to directly improve these representations.
  4. These representations are still not fully contextualized.

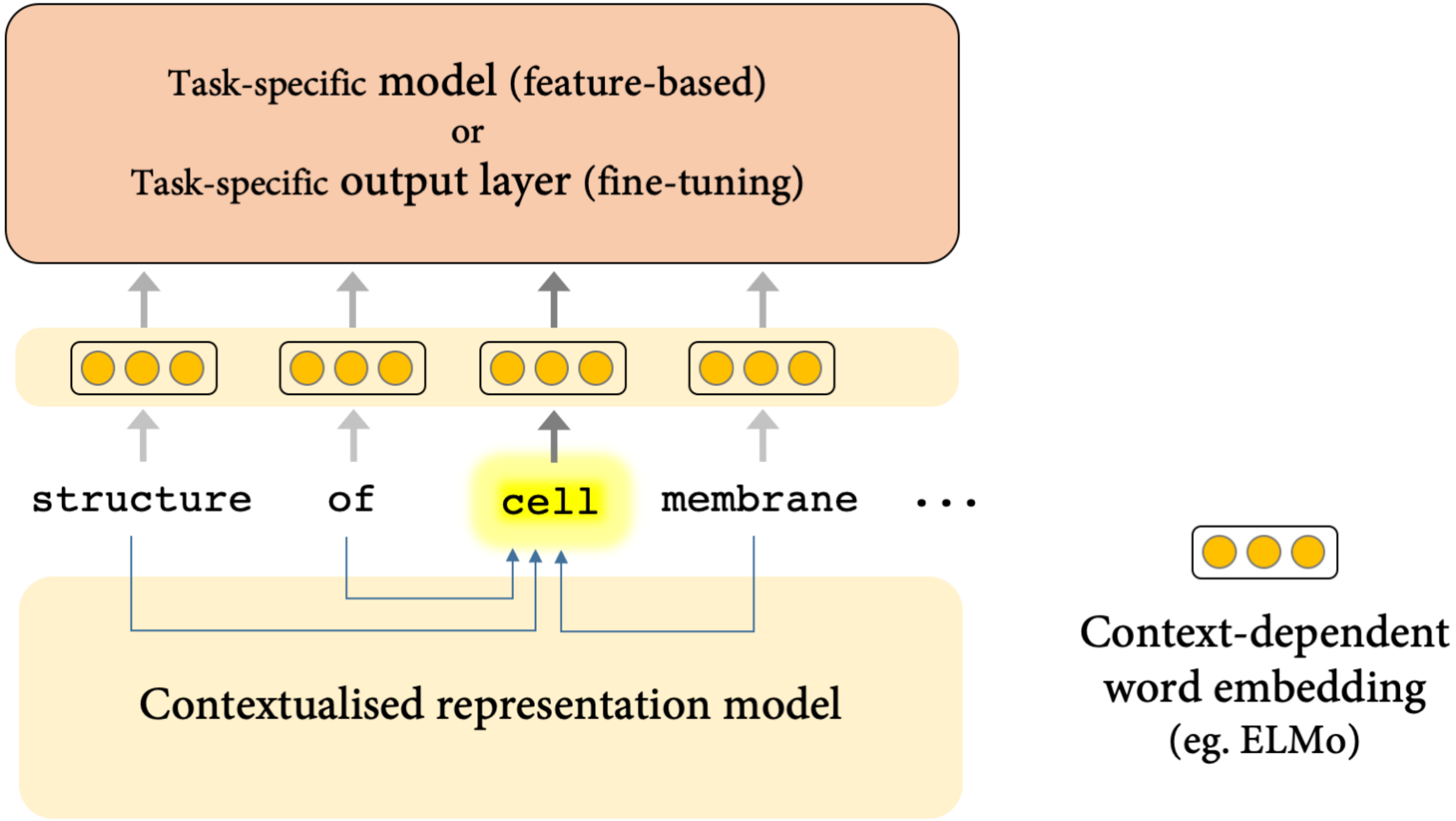
CONTEXTUALIZED REPRESENTATIONS



CONTEXTUALIZED REPRESENTATIONS



CONTEXTUALIZED REPRESENTATIONS



Task-specific model (feature-based)
or
Task-specific output layer (fine-tuning)

Contextualised representation model

Context-dependent word embedding (eg. ELMo)

EMBEDDING LAYER IN TENSORFLOW 2.0

```
from tensorflow.keras.layers import Embedding
embedding_layer = Embedding(1000, 64)
```

- A dictionary that maps integer indices (which stand for specific words) to dense vectors.
- It takes integers as input, it looks up these integers in an internal dictionary, and it returns the associated vectors. It's effectively a dictionary lookup.

EMBEDDING LAYER IN TENSORFLOW 2.0

```
from tensorflow.keras.layers import Embedding
embedding_layer = Embedding(1000, 64)
```

- A dictionary that maps integer indices (which stand for specific words) to dense vectors.
- It takes integers as input, it looks up these integers in an internal dictionary, and it returns the associated vectors. It's effectively a dictionary lookup.

EMBEDDING LAYER IN TENSORFLOW 2.0

- Takes as input a 2D tensor of integers, of shape `(samples, sequence_length)`

- `(32, 10)`
  - (batch of 32 sequences of length 10)

- The output is a 3D tensor with shape `(batch_size, sequence_length, output_dim)`

- All sequences in a batch must have the same length, though (because you need to pack them into a single tensor), so sequences that are shorter than others should be padded with zeros, and sequences that are longer should be truncated.

EMBEDDING LAYER IN TENSORFLOW 2.0

- When you instantiate an Embedding layer, its weights (its internal dictionary of token vectors) are initially random, just as with any other layer.

- During training, these word vectors are gradually adjusted via backpropagation, structuring the space into something the downstream model can exploit.

EMBEDDING LAYER

REFERENCES AND FURTHER RESOURCES

Websites:

1. http://jalammar.github.io/illustrated-word2vec/
2. https://ruder.io/word-embeddings-1/index.html
3. https://code.google.com/archive/p/word2vec/

Papers and Books:

1. Embedding in Natural Language Processing
2. From Word to Sense Embeddings: A Survey on Vector Representations of Meaning
3. Deep learning with python 2nd edition