

Natural Language Processing with Deep Learning

IFT6289, Winter 2022

Lecture 5: Sentence Embedding
Bang Liu

2 Lecture outline

1. Tasks using sentence representations
2. From word embedding to sentence embedding
3. From RNN to CNN
4. Context matters: Elmo
5. Structure matters: hierarchical sentence factorization
6. Multi-task learning for sentence embedding

3 Certain Slides Adapted From or Referred To...

- ◎ **CMU CS11-747 - Neural Networks for NLP, Graham Neubig**
 - Spring 2020: <http://www.phontron.com/class/nn4nlp2020/schedule/contextualword-sentemb.html>
- ◎ **NTU S-108 Applied Deep Learning, Yun-Nung (Vivian) Chen**
 - Spring 2020: <https://www.csie.ntu.edu.tw/~miulab/s108-adl/syllabus>, lecture 3, 5
- ◎ <https://amitness.com/2020/06/universal-sentence-encoder/>

Tasks Using Sentence Representations



5 Where would we need sentence representations?

- Sentence classification
- Paraphrase identification
- Semantic matching
- Entailment
- Retrieval

6 Sentence classification

- Classify sentences according to various traits
- Topic, sentiment, subjectivity/objectivity, etc.



I paid 100 Euros for a really flavourless food and not so delightful ambience.



Food was fine and I wouldn't say it was the best place I have ever tried.



We loved the food. Menu is perfect in here, something for everyone. Visiting this one again.

7 Paraphrase identification

- Identify whether sentence A and sentence B mean the same thing
- Note: exactly the same thing is too restrictive. Therefore, usually we use a loose sense of similarity.

Paraphrases in Twitter (PIT-2015)

Task: Given two sentences from Twitter, predict whether they imply the same meaning.

Roberto Mancini gets the boot from Man City

Roberto Mancini has been sacked by Manchester City with the Blues saying

Yes!

WORLD OF JENKS IS ON AT 11

World of Jenks is my favorite show on tv

No!

Setup:

- ◆ 18k training/dev data:
 - well balanced: about 35% paraphrases, 65% non-paraphrases
 - representative: semi-randomly selected from Twttier's trends
 - annotated by 5 Amazon Mechanical Turkers (good correlation with experts)
- ◆ 1k test data:
 - from a different time period
 - annotated by expert
- ◆ 2 baselines:
 - Supervised: Logistic Regression
 - Unsupervised: Weighted Textual Matrix Factorization

Organizers: Wei Xu, Chris Callison-Burch, Bill Dolan



Microsoft Research

<https://alt.qcri.org/semeval2015/task1/>

8 Semantic similarity/relatedness

- Do two sentences have similar meanings?
- Like paraphrase identification, but with shades of gray.

STSbenchmark dataset

<http://ixa2.si.ehu.es/stswiki/index.php/STSbenchmark>

5	<i>The two sentences are completely equivalent, as they mean the same thing.</i>
	The bird is bathing in the sink. Birdie is washing itself in the water basin.
4	<i>The two sentences are mostly equivalent, but some unimportant details differ.</i>
	Two boys on a couch are playing video games. Two boys are playing a video game.
3	<i>The two sentences are roughly equivalent, but some important information differs/missing.</i>
	John said he is considered a witness but not a suspect. “He is not a suspect anymore.” John said.
2	<i>The two sentences are not equivalent, but share some details.</i>
	They flew out of the nest in groups. They flew into the nest together.
1	<i>The two sentences are not equivalent, but are on the same topic.</i>
	The woman is playing the violin. The young lady enjoys listening to the guitar.
0	<i>The two sentences are completely dissimilar.</i>
	The black dog is running through the snow. A race car driver is driving his car through the mud.

Table 1: Similarity scores with explanations and English examples from Agirre et al. (2013).

9 Textual entailment

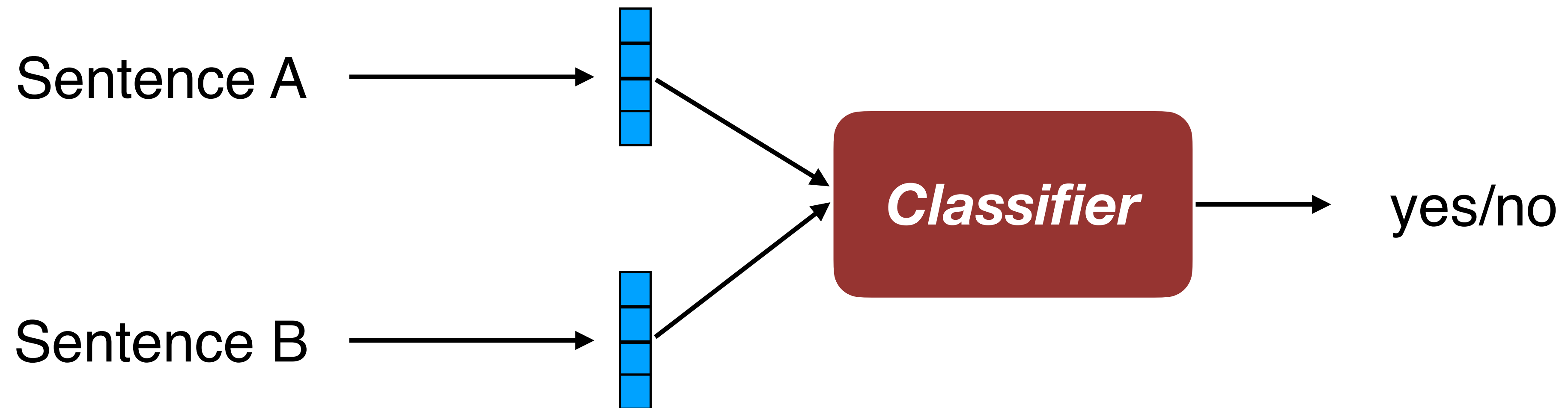
- **Entailment:** if A is true, then B is true (c.f. paraphrase, where opposite is also true)
- **Contradiction:** if A is true, then B is not true
- **Neutral:** cannot say either of the above

P	A woman is talking on the phone while standing next to a dog	
H1	A woman is on the phone	entailment
H2	A woman is walking her dog	neutral
H3	A woman is sleeping	contradiction
P	Tax records show Waters earned around \$65,000 in 2000	
H1	Waters' tax records show clearly that he earned a lovely \$65k in 2000	entailment
H2	Tax records indicate Waters earned about \$65K in 2000	entailment
H3	Waters' tax records show he earned a blue ribbon last year	contradiction

Table 2: Examples from the development sets of SNLI (top) and MultiNLI (bottom). Each example contains one premise that is paired with three hypotheses in the datasets.

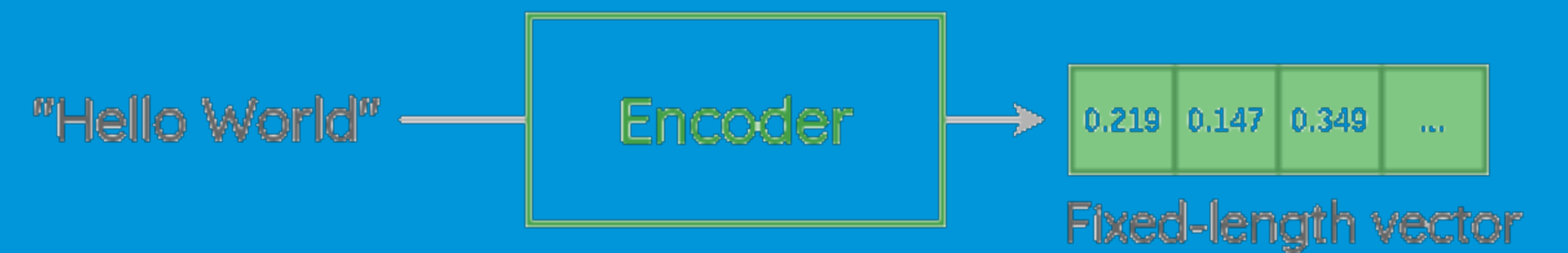
10 Model for Sentence Pair Processing

- Calculate vector representation
- Feed vector representation into classifier



How do we get such a representation?

From Word Embedding to Sentence Embedding



Doc2Vec

13 Bag-of-Words/Bag-of-n-grams

- **Bag-of-Words (BOW):** no word order, different sentences can have same meaning
- **Bag-of-n-grams:** order in short context, data sparsity, high dimensionality, little sense about word semantics

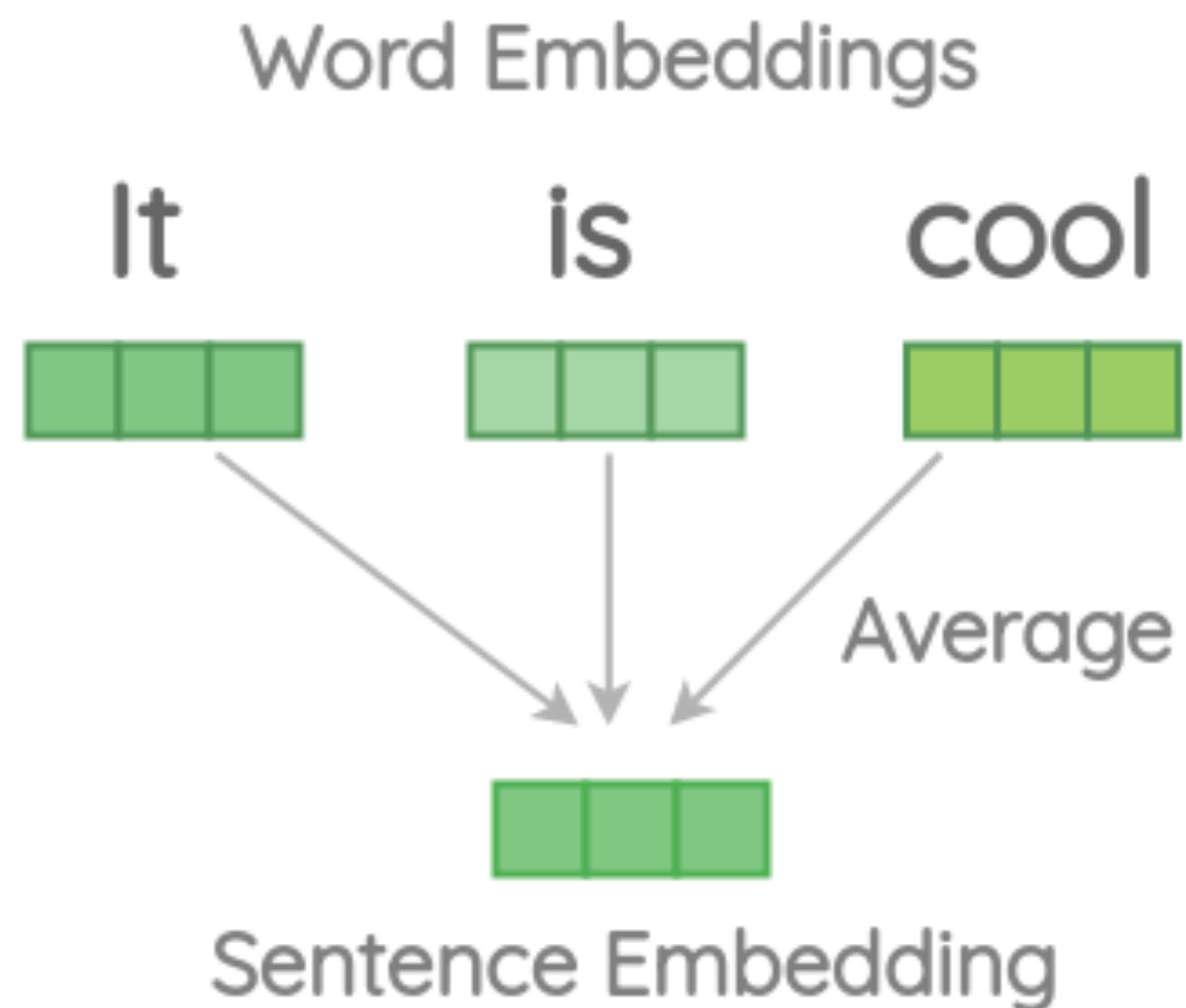


It was the best of times



14 Weighted averaging of word embeddings

- Loses the word order in the same way as the standard bag-of-words models do.



15 Framework of learning word vectors

- Map a word to a unique vector: A particular implementation for training the word vectors: code.google.com/p/word2vec/ (Mikolov et al., 2013a).

- Predict the next word in a sentence

Training objective: maximize the average log probability:

$$\frac{1}{T} \sum_{t-k}^{T-k} \log p(w_t | w_{t-k}, \dots, w_{t+k})$$

$$p(w_t | w_{t-k}, \dots, w_{t+k}) = \frac{e^{y_{w_t}}}{\sum_i e^{y_i}}$$

Each of y_i is un-normalized log-probability for each output word i :

$$y = b + Uh(w_{t-k}, \dots, w_{t+k}; W)$$

Classifier

Average/Concatenate

Word Matrix

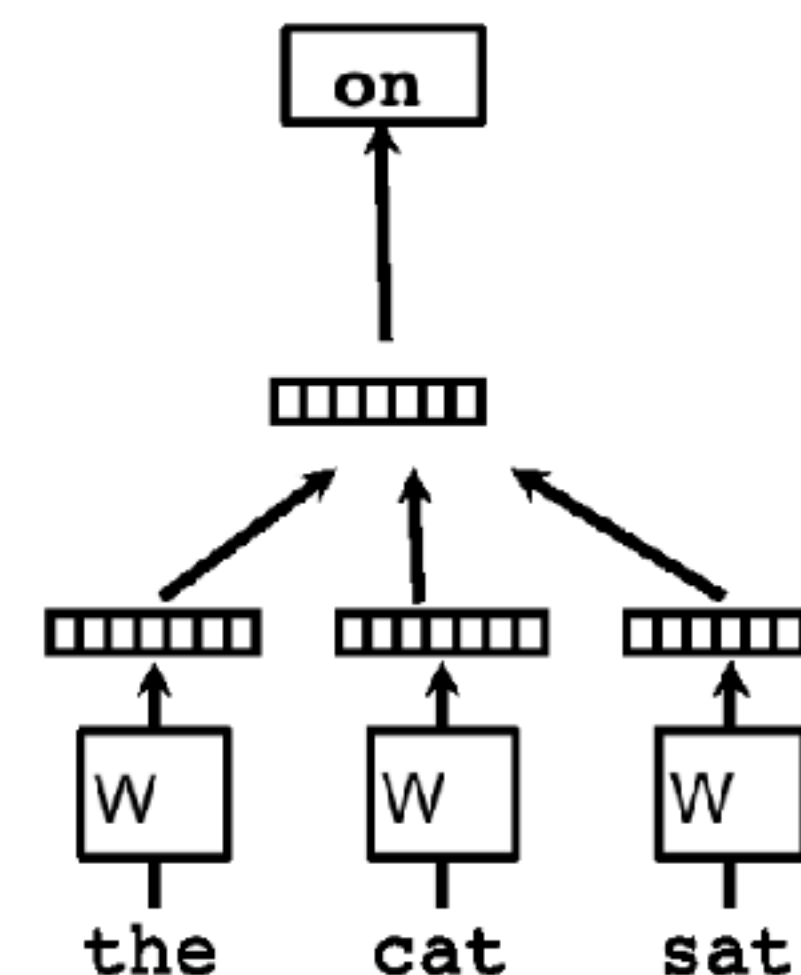


Figure 1. A framework for learning word vectors. Context of three words (“the,” “cat,” and “sat”) is used to predict the fourth word (“on”). The input words are mapped to columns of the matrix W to predict the output word.

From Word2Vec to Doc2Vec

Distributed Memory version of Paragraph Vector

- **Paragraph Vector**, an unsupervised algorithm that learns fixed-length feature representations from variable-length pieces of texts (sentences, paragraphs, documents)
- Assign a paragraph vector while sharing word vectors among all sentences. Then we either average or concatenate them (paragraph vector and words vector) to get the final sentence representation.
- If you notice, it is an extension of the **Continuous Bag-of-Word** type of Word2Vec

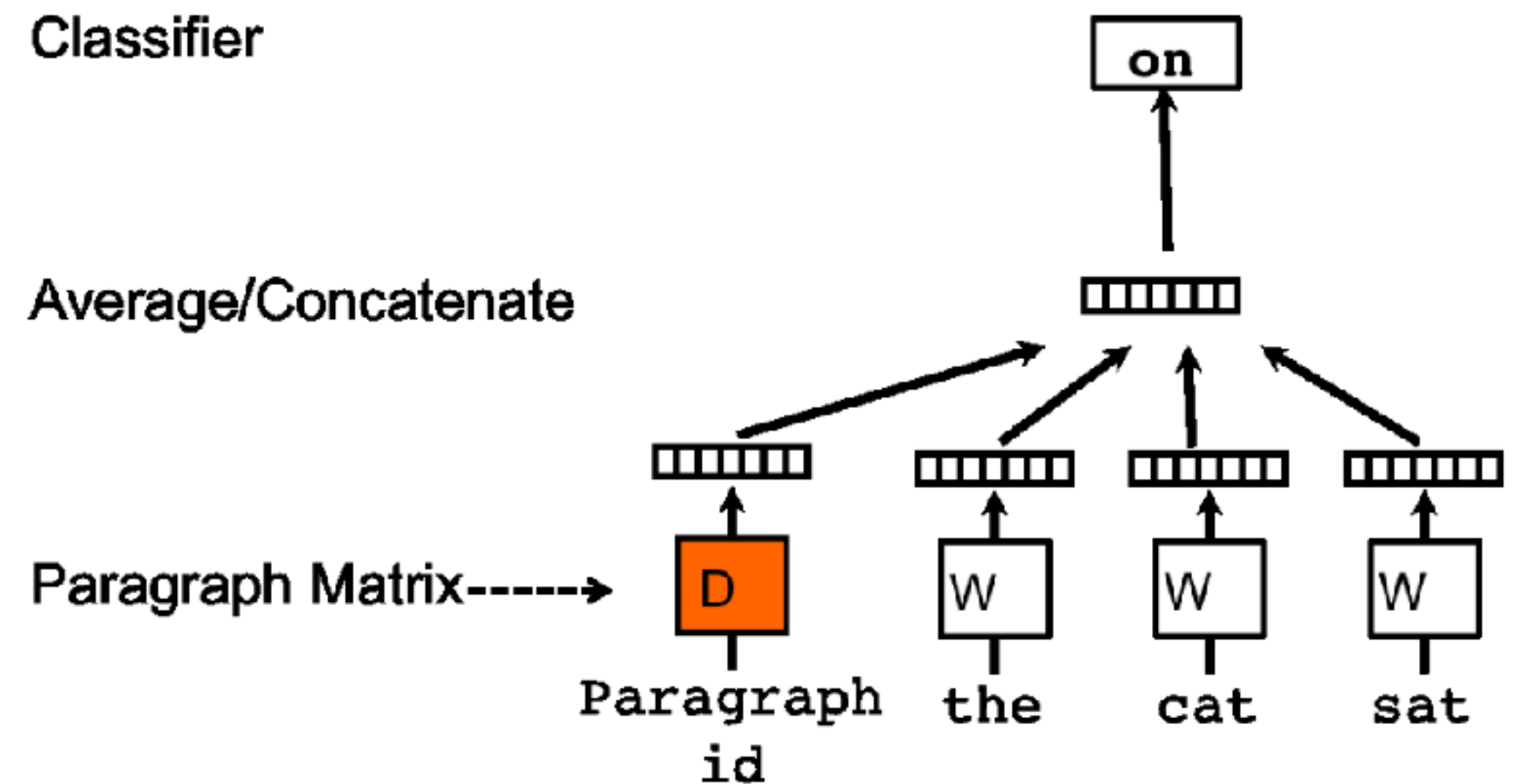


Figure 2. A framework for learning paragraph vector. This framework is similar to the framework presented in Figure 1; the only change is the additional paragraph token that is mapped to a vector via matrix D . In this model, the concatenation or average of this vector with a context of three words is used to predict the fourth word. The paragraph vector represents the missing information from the current context and can act as a memory of the topic of the paragraph.

From Word2Vec to Doc2Vec

Distributed Memory version of Paragraph Vector

- **Training stage:** training to get word vectors W , softmax weights U , b and paragraph vectors D on already seen paragraphs
- **Inference stage:** get paragraph vectors D for new paragraphs (never seen before) by adding more columns in D and gradient descending on D while holding W , U , b fixed.
- D can be utilized for text classification tasks.

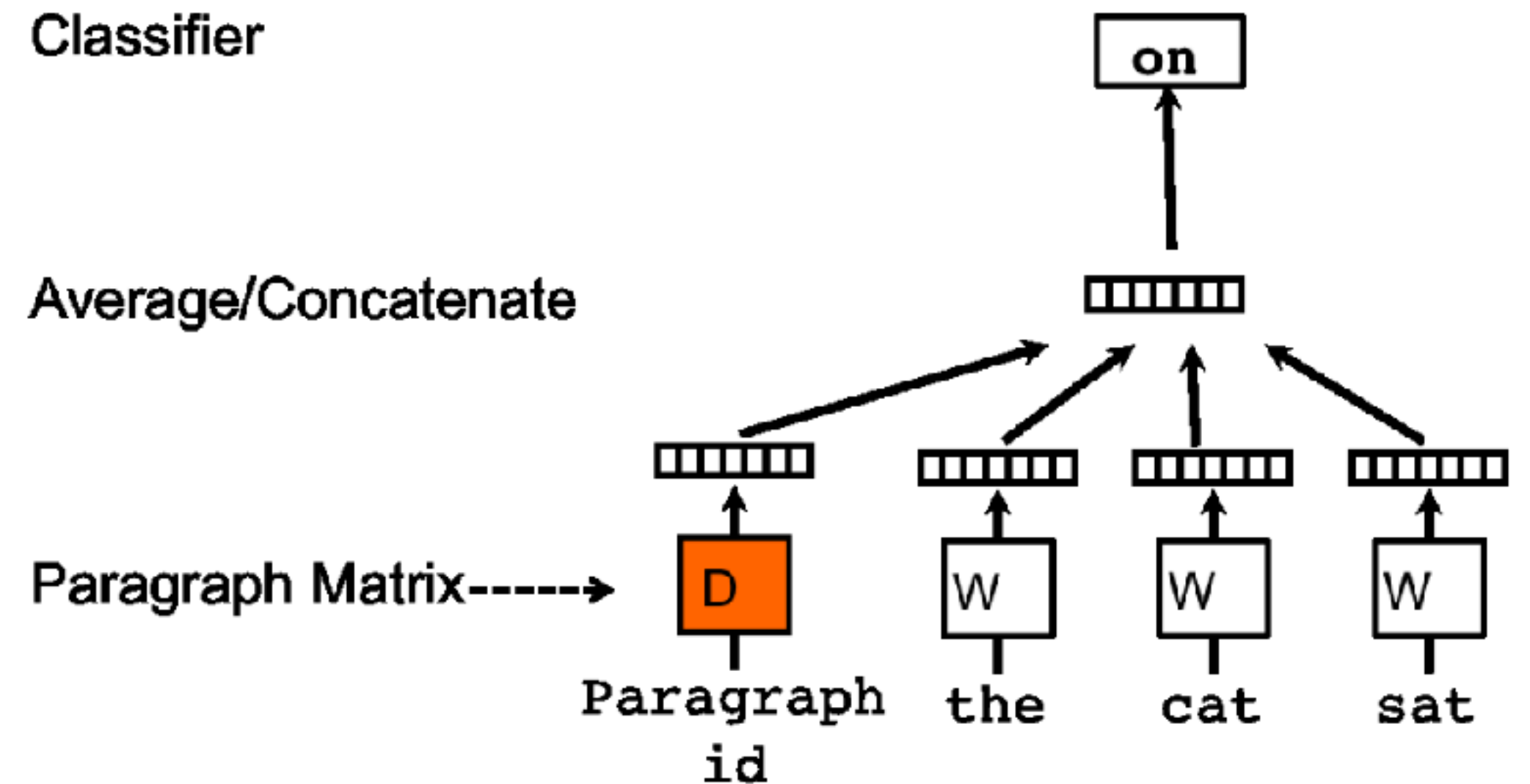


Figure 2. A framework for learning paragraph vector. This framework is similar to the framework presented in Figure 1; the only change is the additional paragraph token that is mapped to a vector via matrix D . In this model, the concatenation or average of this vector with a context of three words is used to predict the fourth word. The paragraph vector represents the missing information from the current context and can act as a memory of the topic of the paragraph.

From Word2Vec to Doc2Vec

Distributed Bag of Words version of Paragraph Vector

- PVDOBW is another extension, this time of the Skip-gram type.
- Here, we just sample random words from the sentence and make the model predict which sentence it came from (a classification task).
- The authors of the paper recommend using both in combination, but state that usually PVDM is more than enough for most tasks.

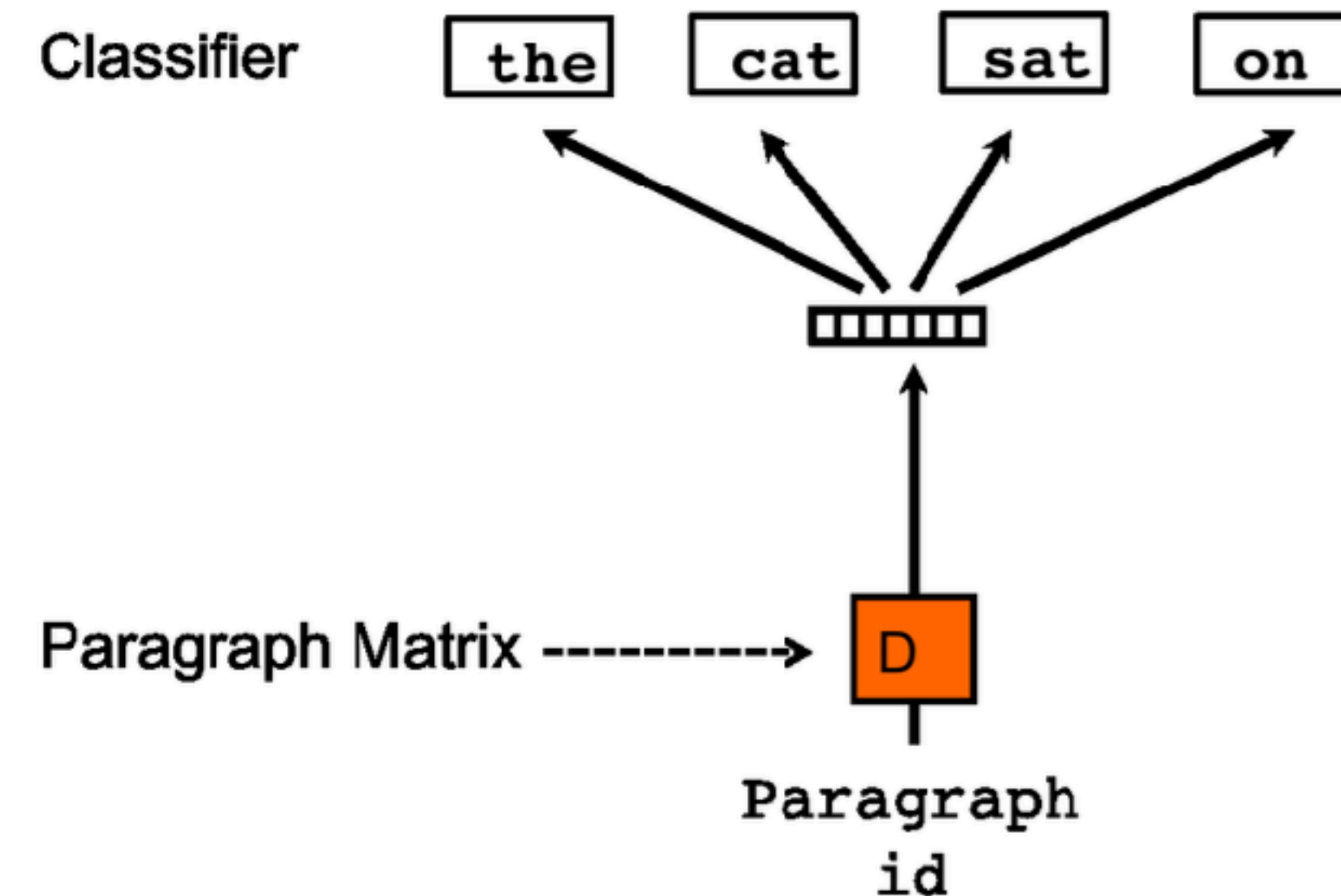


Figure 3. Distributed Bag of Words version of paragraph vectors. In this version, the paragraph vector is trained to predict the words in a small window.

19 Doc2Vec on Sentiment Classification

Table 1. The performance of our method compared to other approaches on the Stanford Sentiment Treebank dataset. The error rates of other methods are reported in (Socher et al., 2013b).

Model	Error rate (Positive/Negative)	Error rate (Fine-grained)
Naïve Bayes (Socher et al., 2013b)	18.2 %	59.0%
SVMs (Socher et al., 2013b)	20.6%	59.3%
Bigram Naïve Bayes (Socher et al., 2013b)	16.9%	58.1%
Word Vector Averaging (Socher et al., 2013b)	19.9%	67.3%
Recursive Neural Network (Socher et al., 2013b)	17.6%	56.8%
Matrix Vector-RNN (Socher et al., 2013b)	17.1%	55.6%
Recursive Neural Tensor Network (Socher et al., 2013b)	14.6%	54.3%
Paragraph Vector	12.2%	51.3%

Table 2. The performance of Paragraph Vector compared to other approaches on the IMDB dataset. The error rates of other methods are reported in (Wang & Manning, 2012).

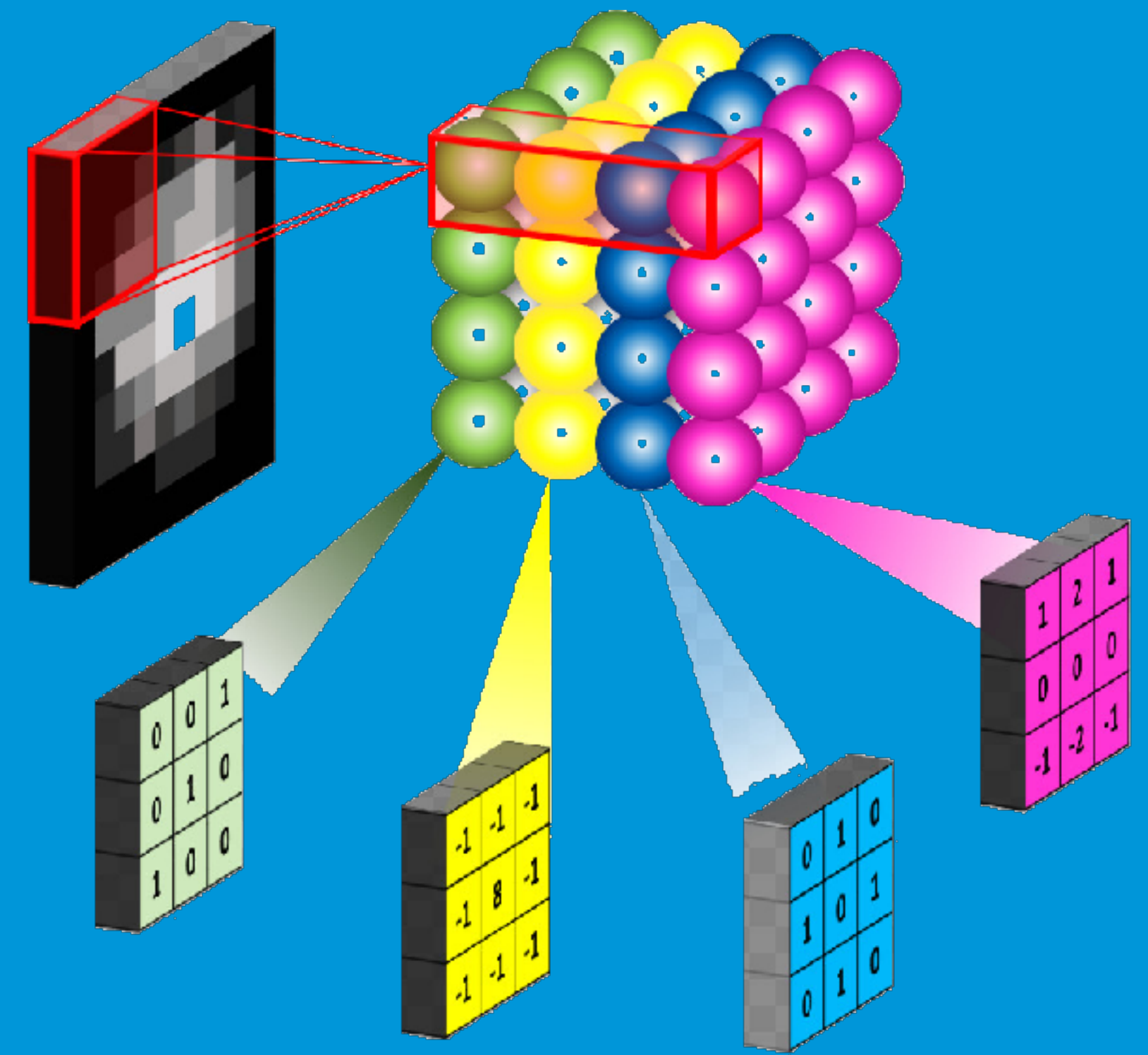
Model	Error rate
BoW (bnc) (Maas et al., 2011)	12.20 %
BoW (b Δ t'c) (Maas et al., 2011)	11.77%
LDA (Maas et al., 2011)	32.58%
Full+BoW (Maas et al., 2011)	11.67%
Full+Unlabeled+BoW (Maas et al., 2011)	11.11%
WRRBM (Dahl et al., 2012)	12.58%
WRRBM + BoW (bnc) (Dahl et al., 2012)	10.77%
MNB-uni (Wang & Manning, 2012)	16.45%
MNB-bi (Wang & Manning, 2012)	13.41%
SVM-uni (Wang & Manning, 2012)	13.05%
SVM-bi (Wang & Manning, 2012)	10.84%
NBSVM-uni (Wang & Manning, 2012)	11.71%
NBSVM-bi (Wang & Manning, 2012)	8.78%
Paragraph Vector	7.42%

Doc2Vec on Information Retrieval

Table 3. The performance of Paragraph Vector and bag-of-words models on the information retrieval task. “Weighted Bag-of-bigrams” is the method where we learn a linear matrix W on TF-IDF bigram features that maximizes the distance between the first and the third paragraph and minimizes the distance between the first and the second paragraph.

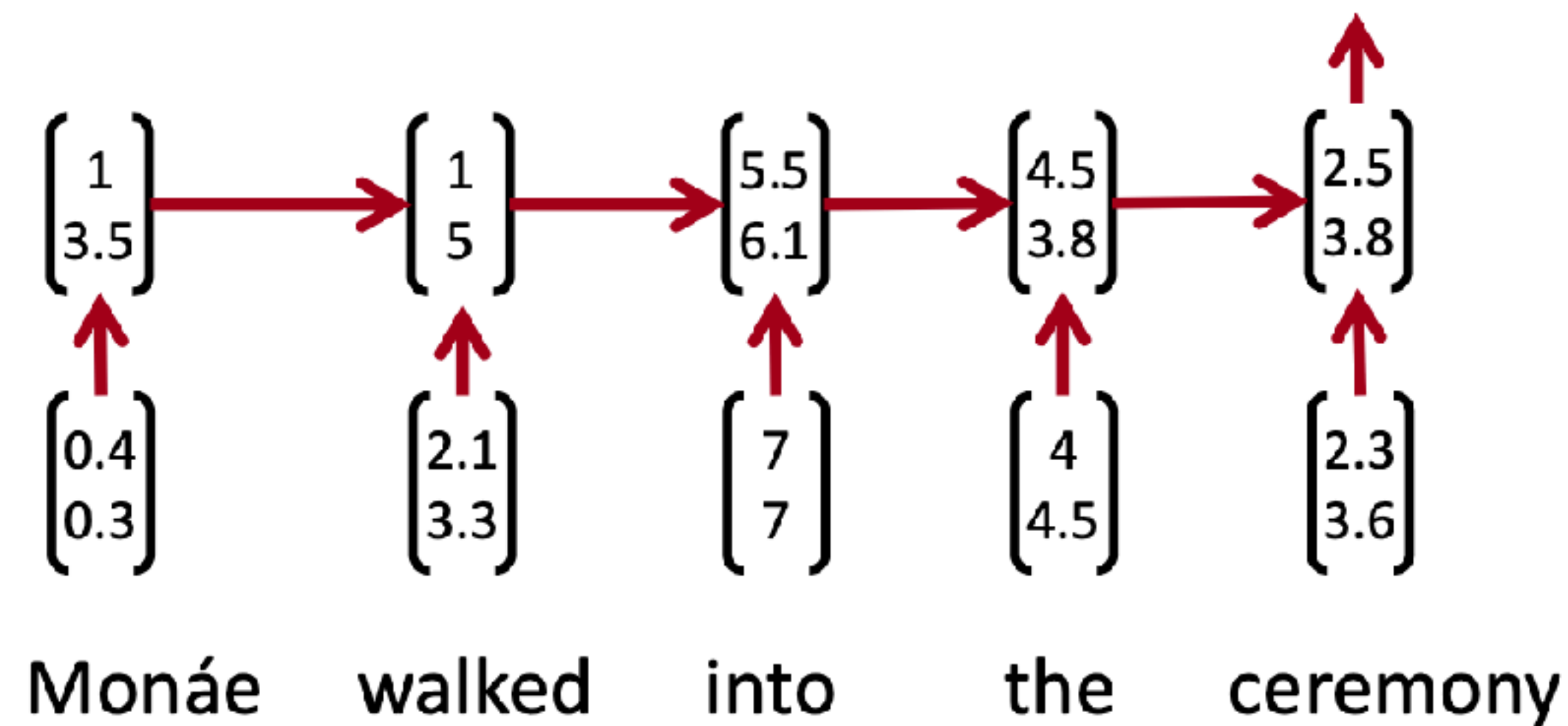
Model	Error rate
Vector Averaging	10.25%
Bag-of-words	8.10 %
Bag-of-bigrams	7.28 %
Weighted Bag-of-bigrams	5.67%
Paragraph Vector	3.82%

From RNN to CNN



22 From RNNs to Convolutional Neural Nets

- Recurrent neural nets cannot capture phrases without prefix context
- Often capture too much of last words in final vector



Softmax is often only calculated at the last step.

23 From RNNs to Convolutional Neural Nets

- ⦿ What if we compute vectors for every possible word subsequence of a certain length?
- ⦿ Regardless of whether phrase is grammatical (not very linguistically or cognitively plausible)

tentative deal reached to keep government open

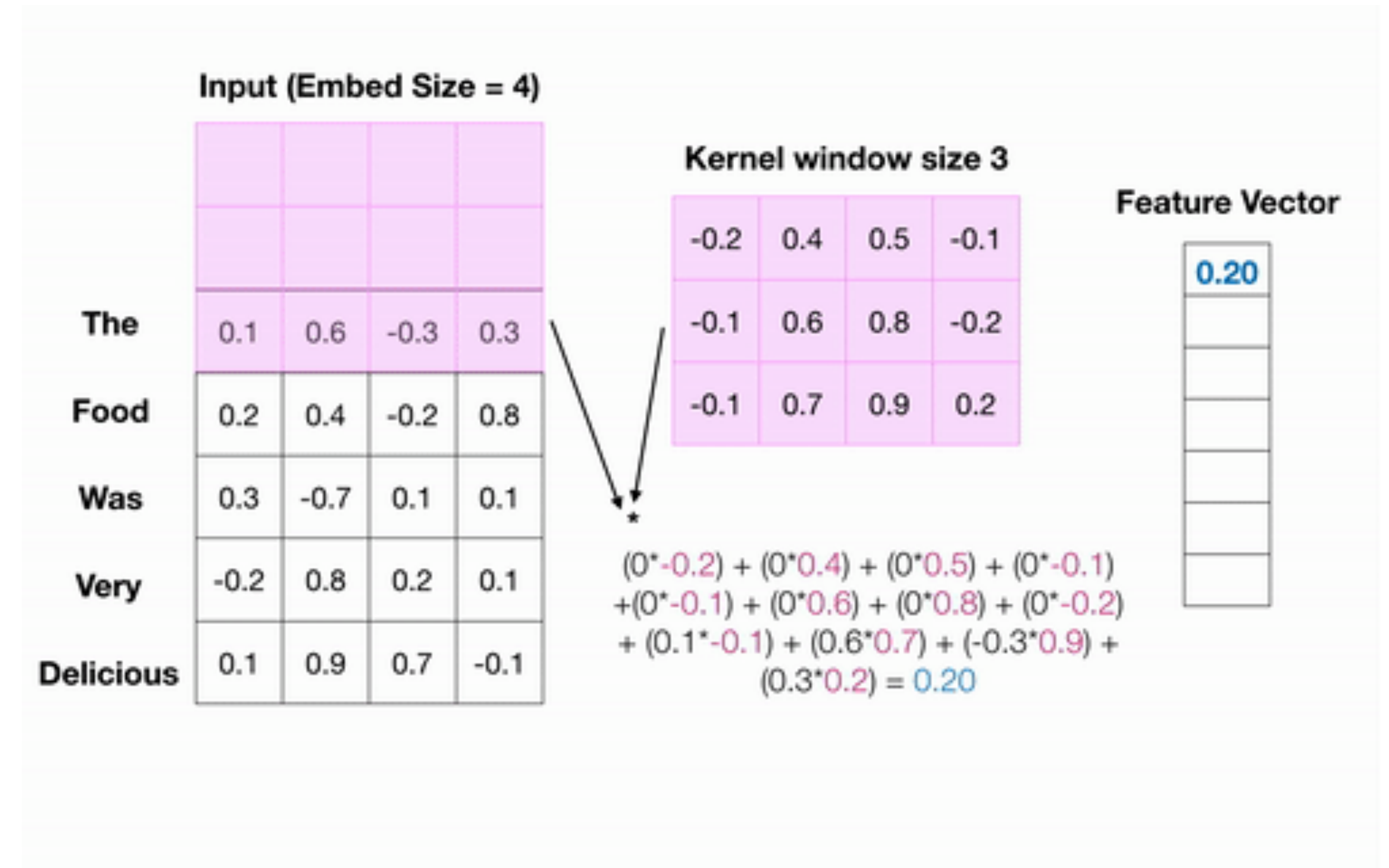
The diagram consists of several horizontal black lines of varying lengths and positions, illustrating overlapping word subsequences. The lines are arranged in a staggered, overlapping fashion, with some lines starting further to the right than others, representing different windows of a fixed length sliding across the text above.

24 1d convolution

- 1D discrete convolution

$$(f * g)[n] = \sum_{m=-M}^M f[n - m]g[m]$$

- Mostly utilized in signal processing
- Example: (on the right)
 - Kernel window size: 3
 - Number of filters: 1
 - Padding size: 2
 - Stride size: 1



2d convolution

- 2D discrete convolution

$$(f * g)[m, n] = \sum_{i=-M}^M \sum_{j=-N}^N f[i, j]g[m - i, n - j]$$

- Classically used to extract features from images

- Example: (on the right)

- Kernel window size: 3*3
- Number of filters: 1
- Padding size: 0
- Stride size: 1

1 _{x1}	1 _{x0}	1 _{x1}	0	0
0 _{x0}	1 _{x1}	1 _{x0}	1	0
0 _{x1}	0 _{x0}	1 _{x1}	1	1
0	0	1	1	0
0	1	1	0	0

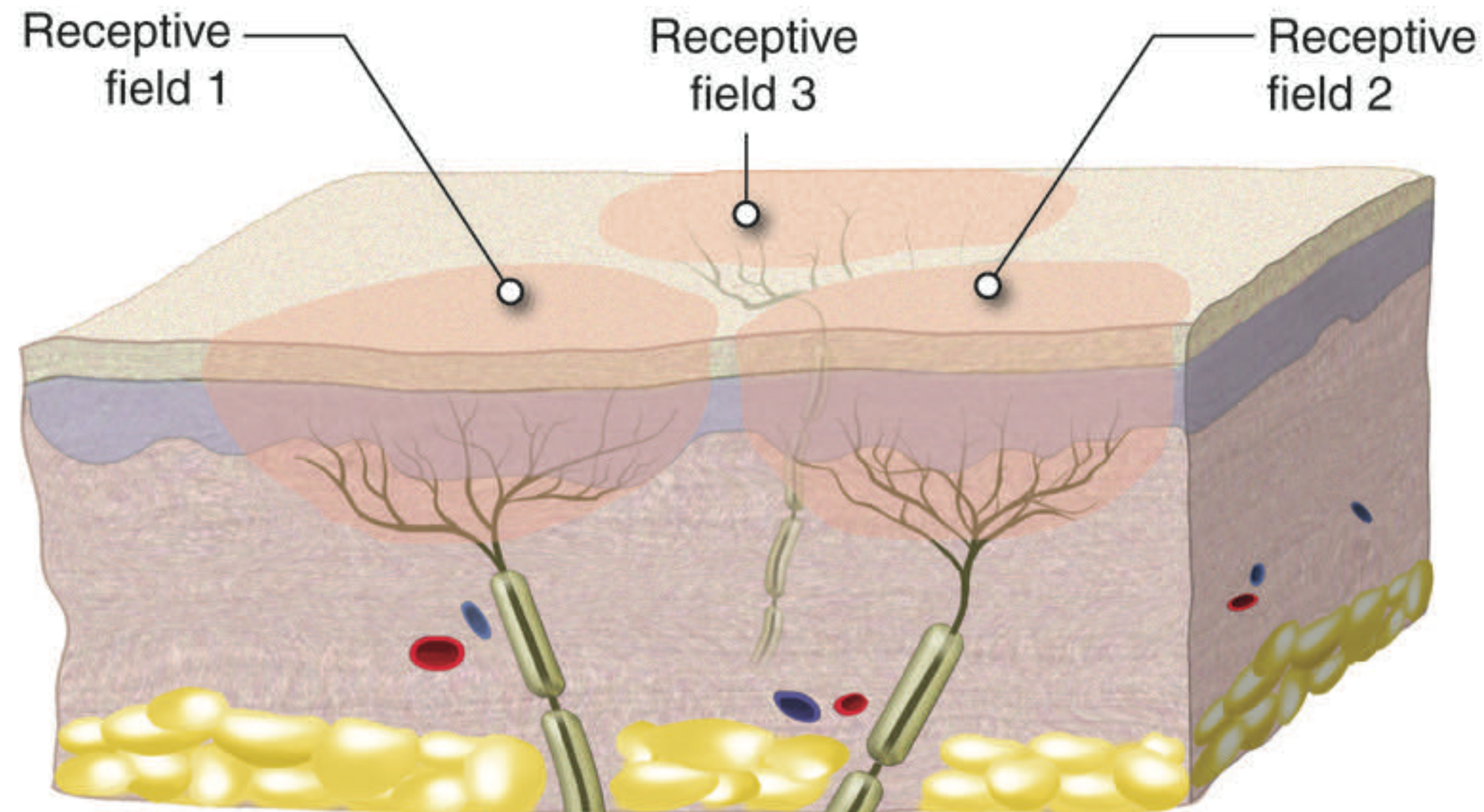
Image

4		

Convolved Feature

Receptive field and CNN

- **Convolutional Neural Networks (CNN)** is a type of feedforward neural network. It is motivated by biologically receptive fields mechanism.
- A **receptive field** is an area in which stimulation leads to response of a particular sensory neuron.



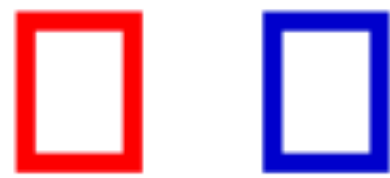
27 Receptive field and CNN



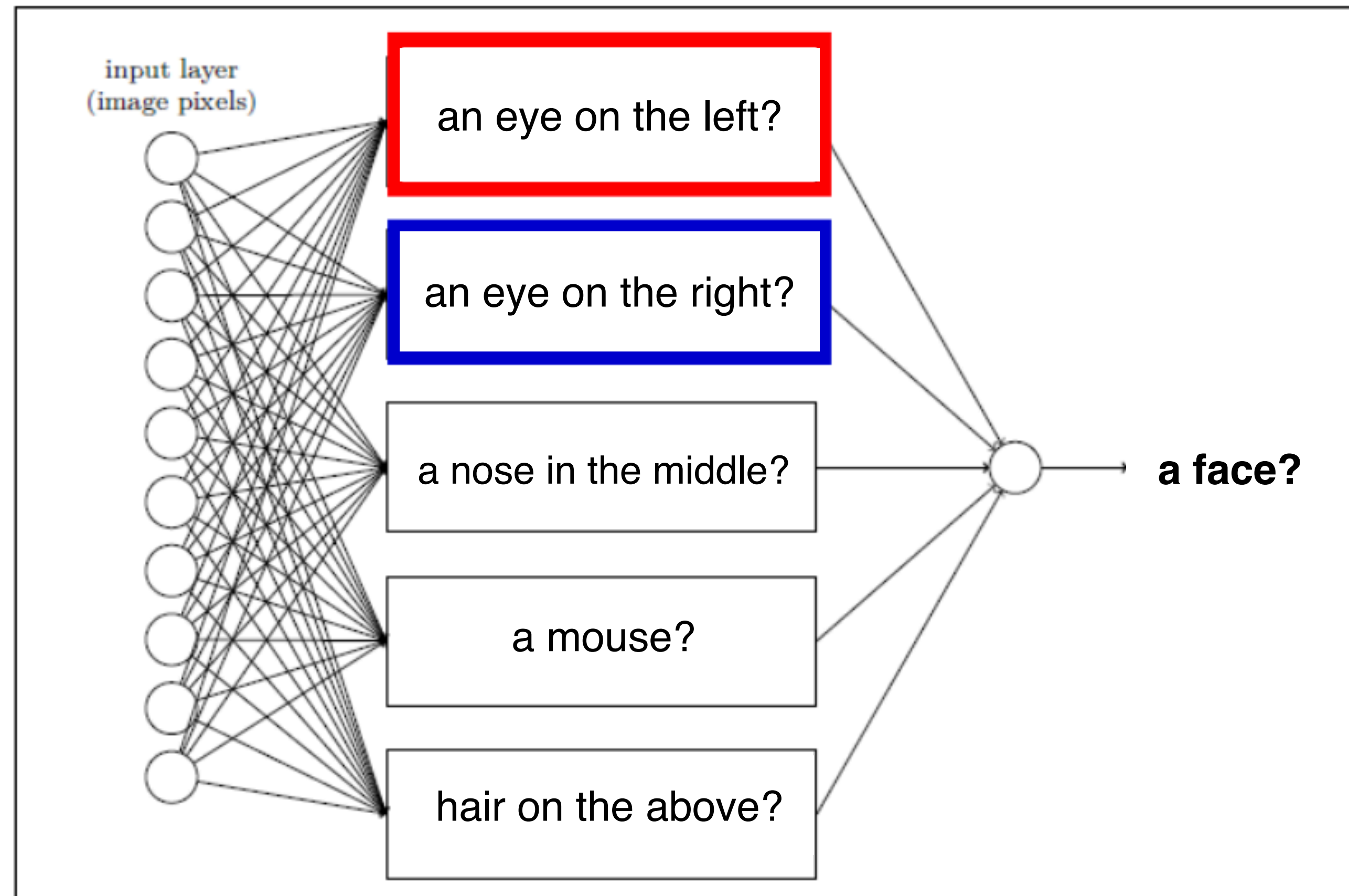
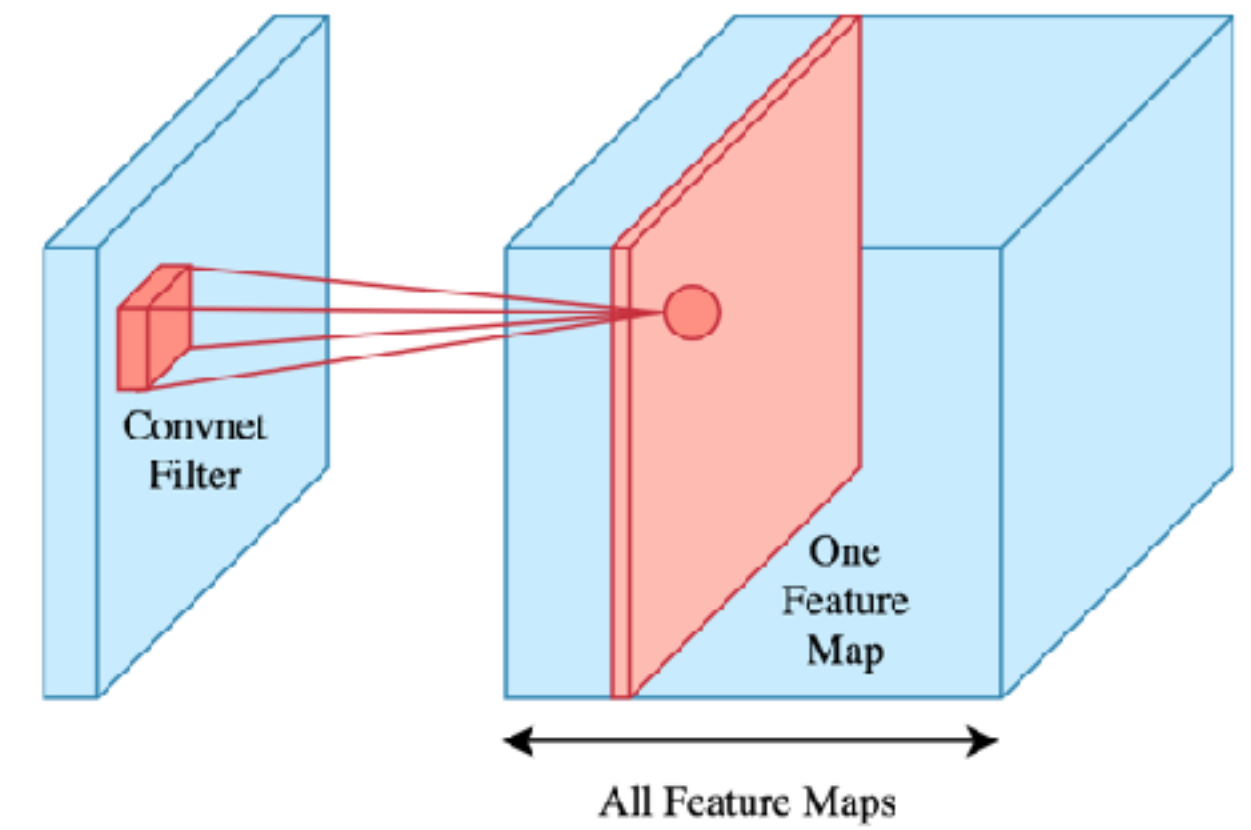
How to recognize?

Receptive field and CNN

Convolutional filters

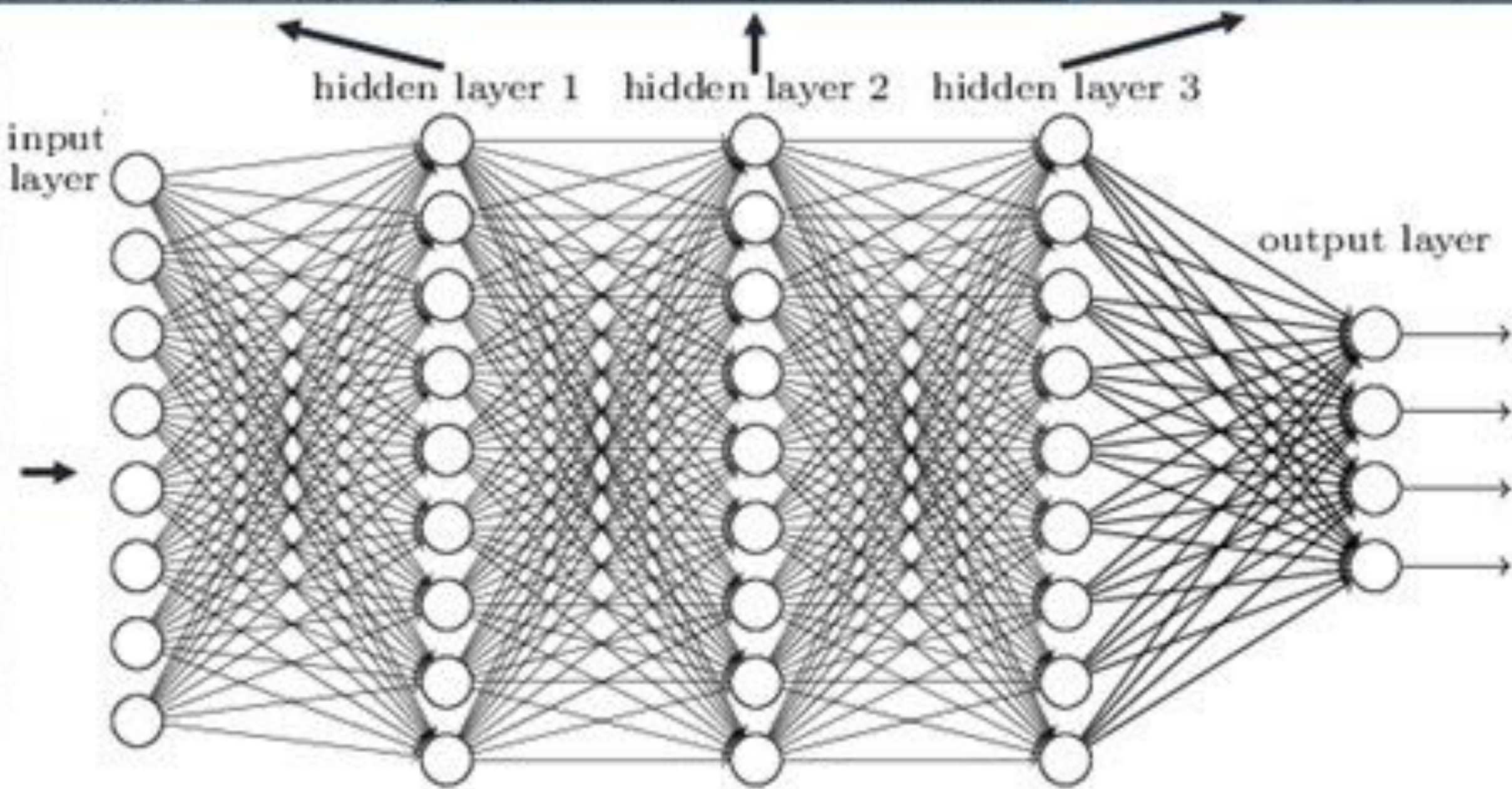
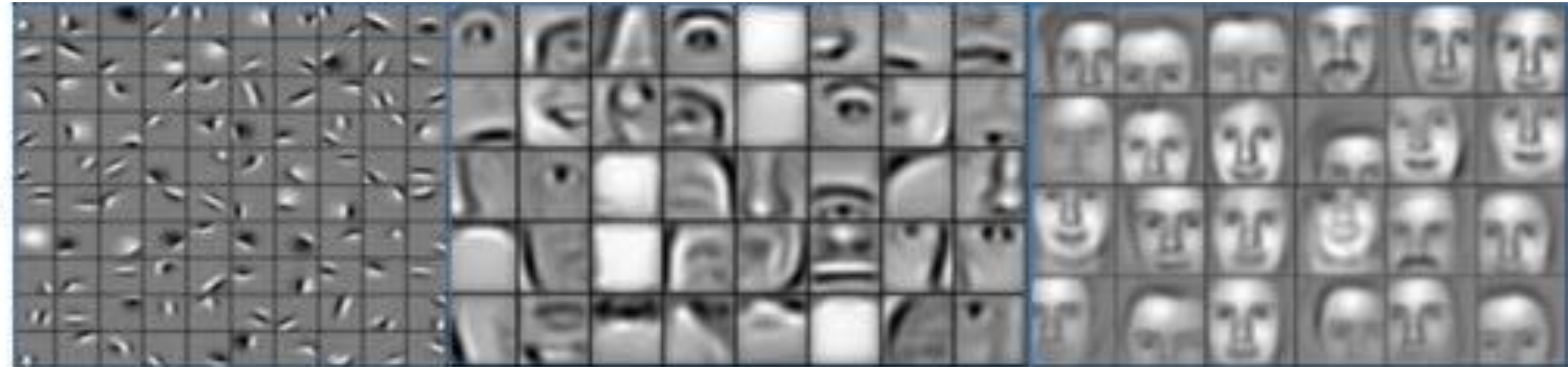


Feature maps



29 Hierarchical feature representations

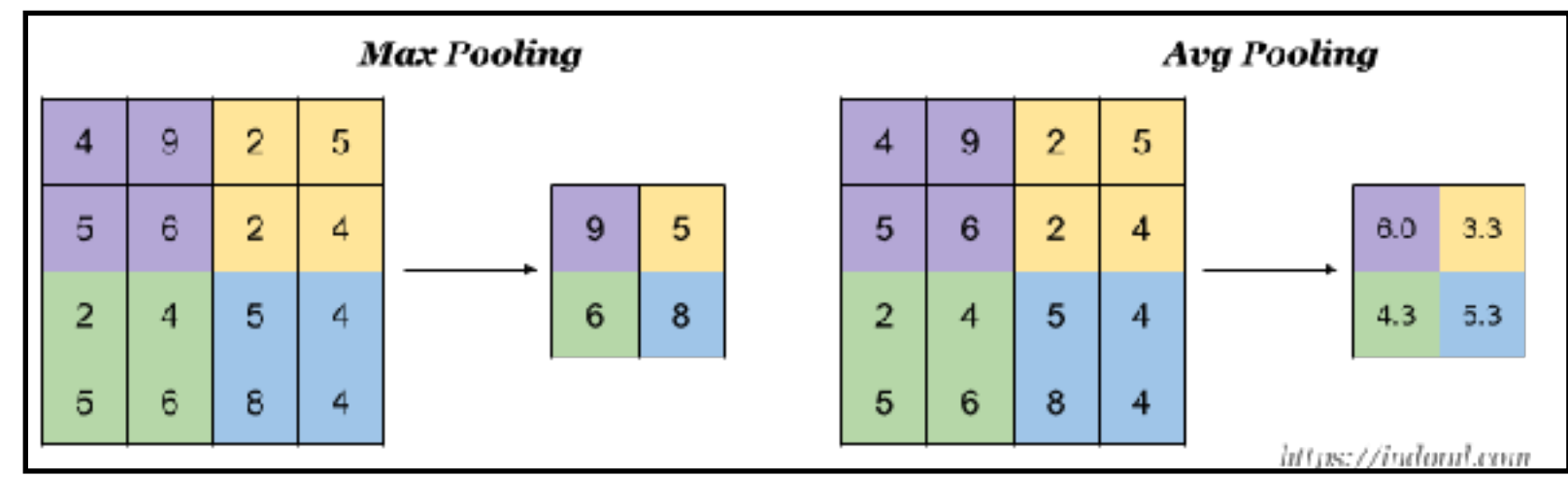
Deep neural networks learn hierarchical feature representations



CNN for sentence classification

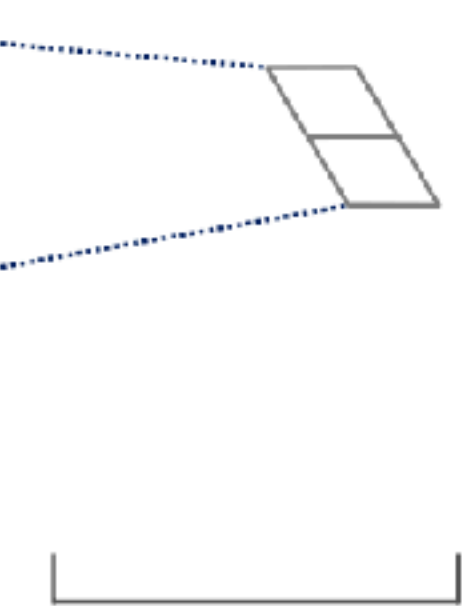
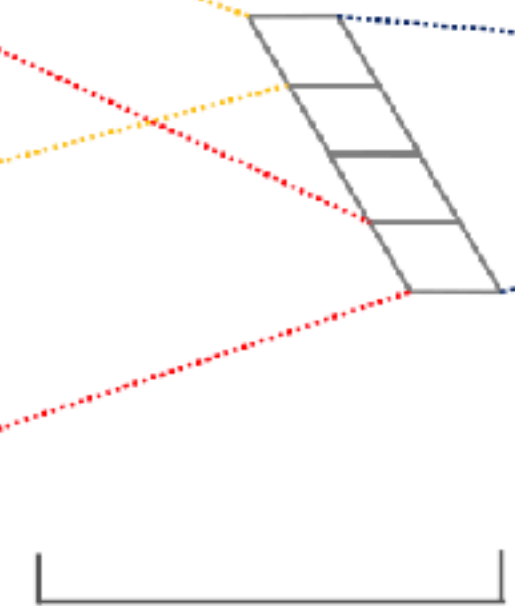
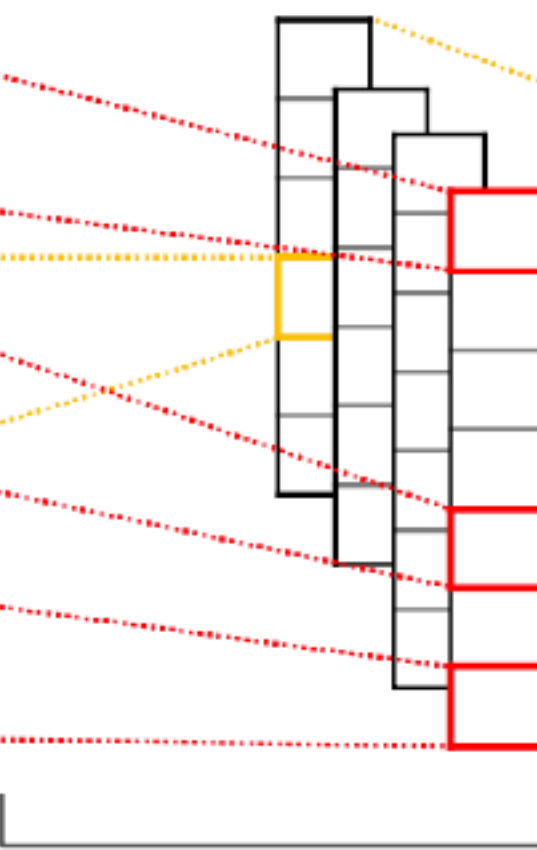
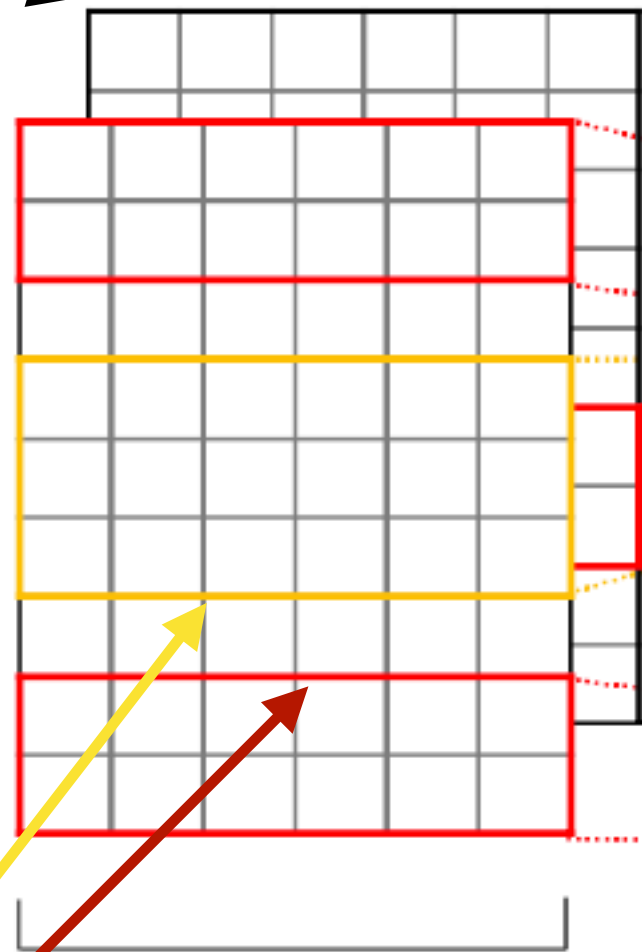
Input: 2 channels
random vectors + word2vec

Model architecture:
single layer 1d CNN



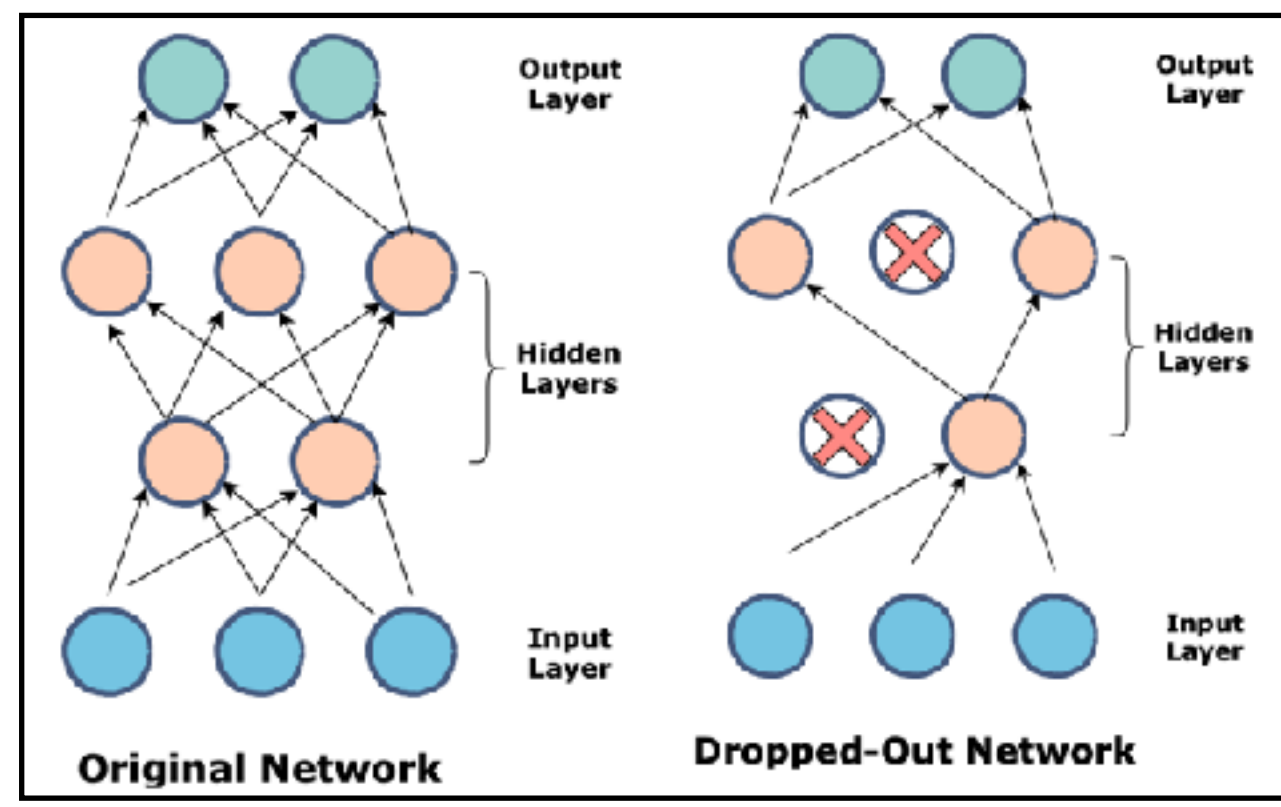
Pooling:
1. capture the most important feature for each feature map.
2. naturally deals with variable sentence lengths.

wait for the video and do n't rent it

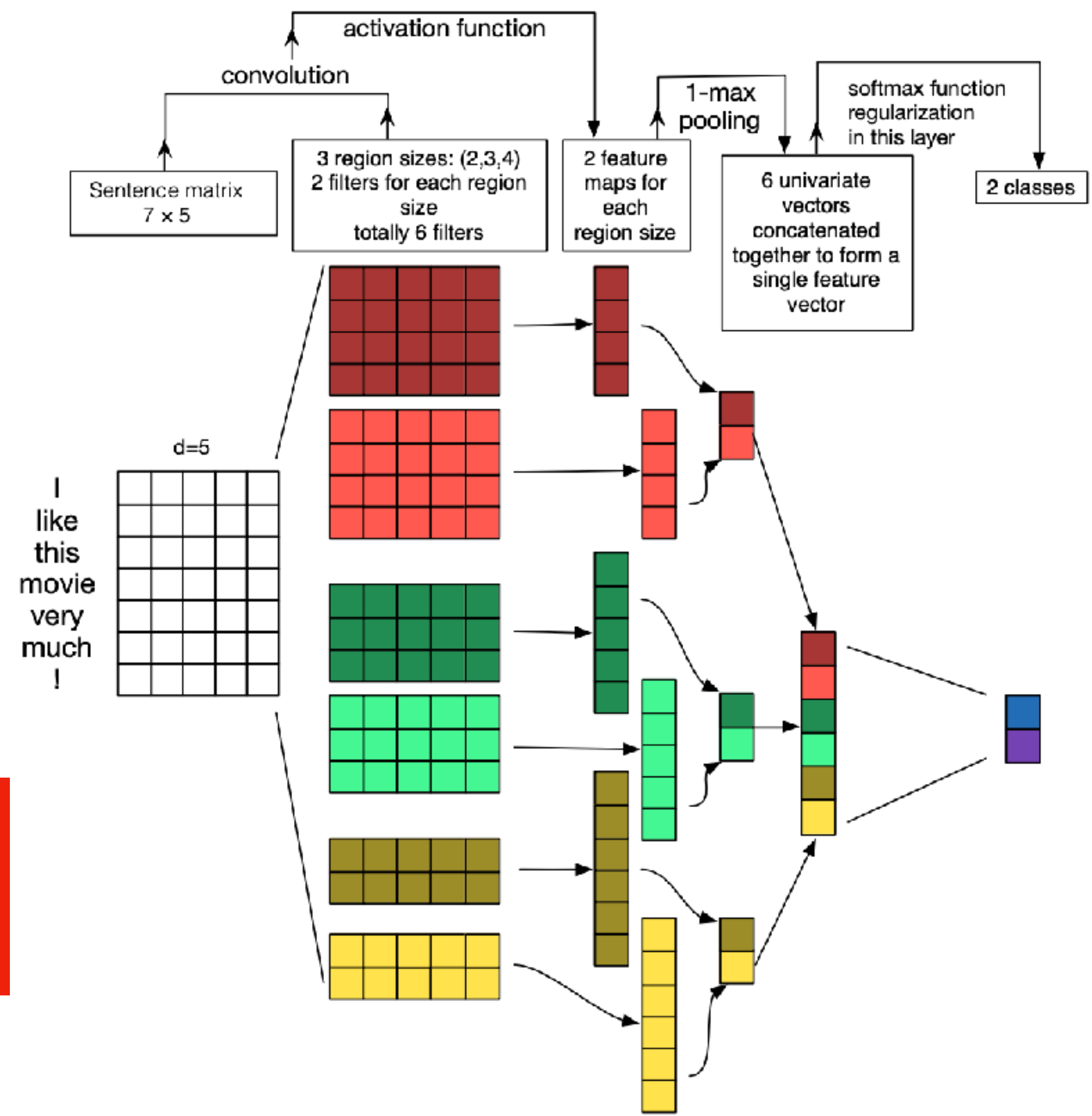


Feature extraction:
uses multiple filters (with varying window sizes) to obtain multiple features

Regularization:
dropout and constraint l2-norms of weight vectors



CNN for sentence classification



The best settings depends on the task and dataset!

**Following previous work (Kim 2014).
Test the effects of different settings.**

- Effects of different settings:
1. input word vectors: word2vec, glove, concatenate
 2. filter region size
 3. number of feature maps
 4. activation function
 5. pooling strategy
 6. regularization: dropout and l2 norm constraint

Context Matters: ELMo

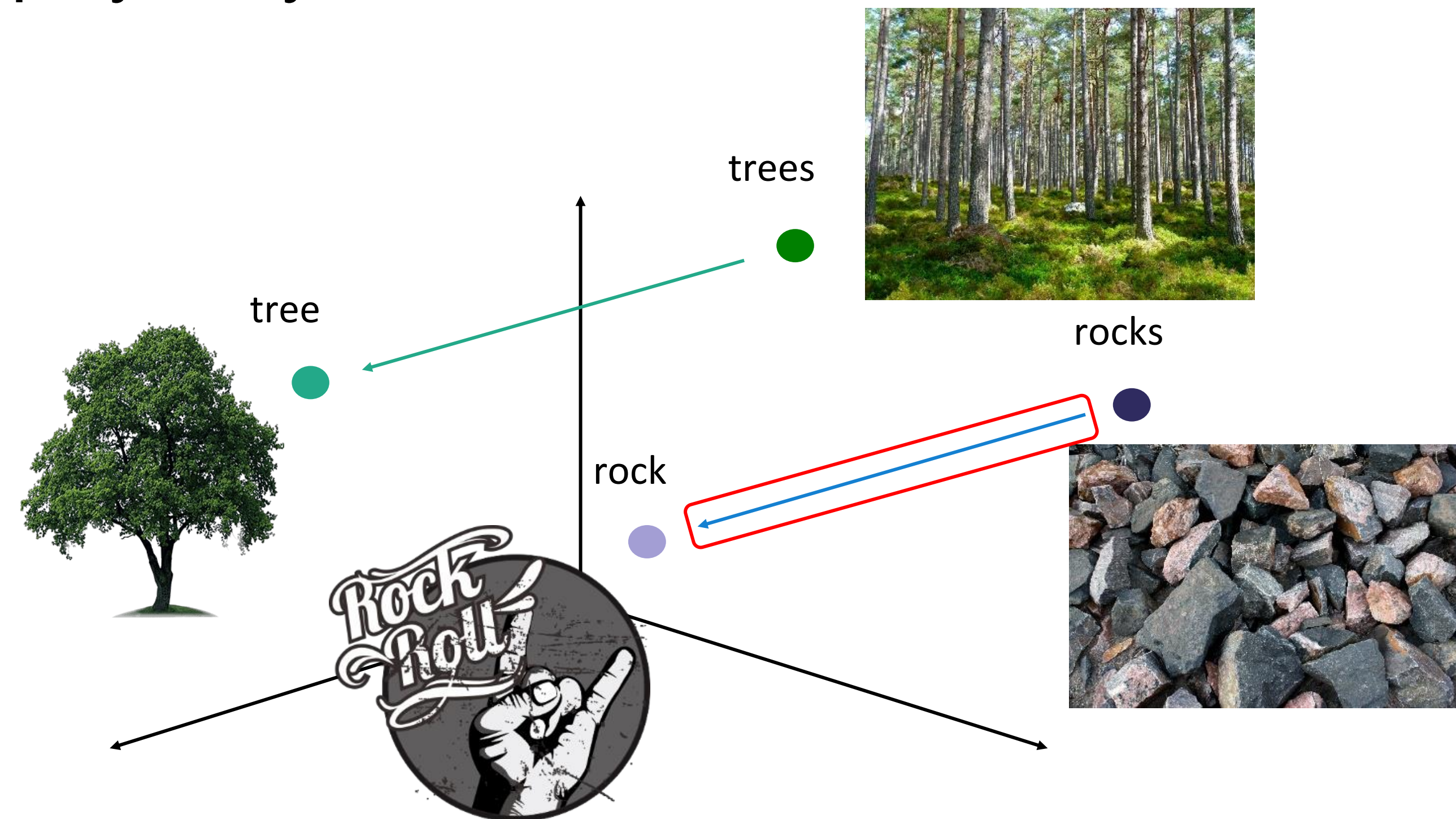


Context matters



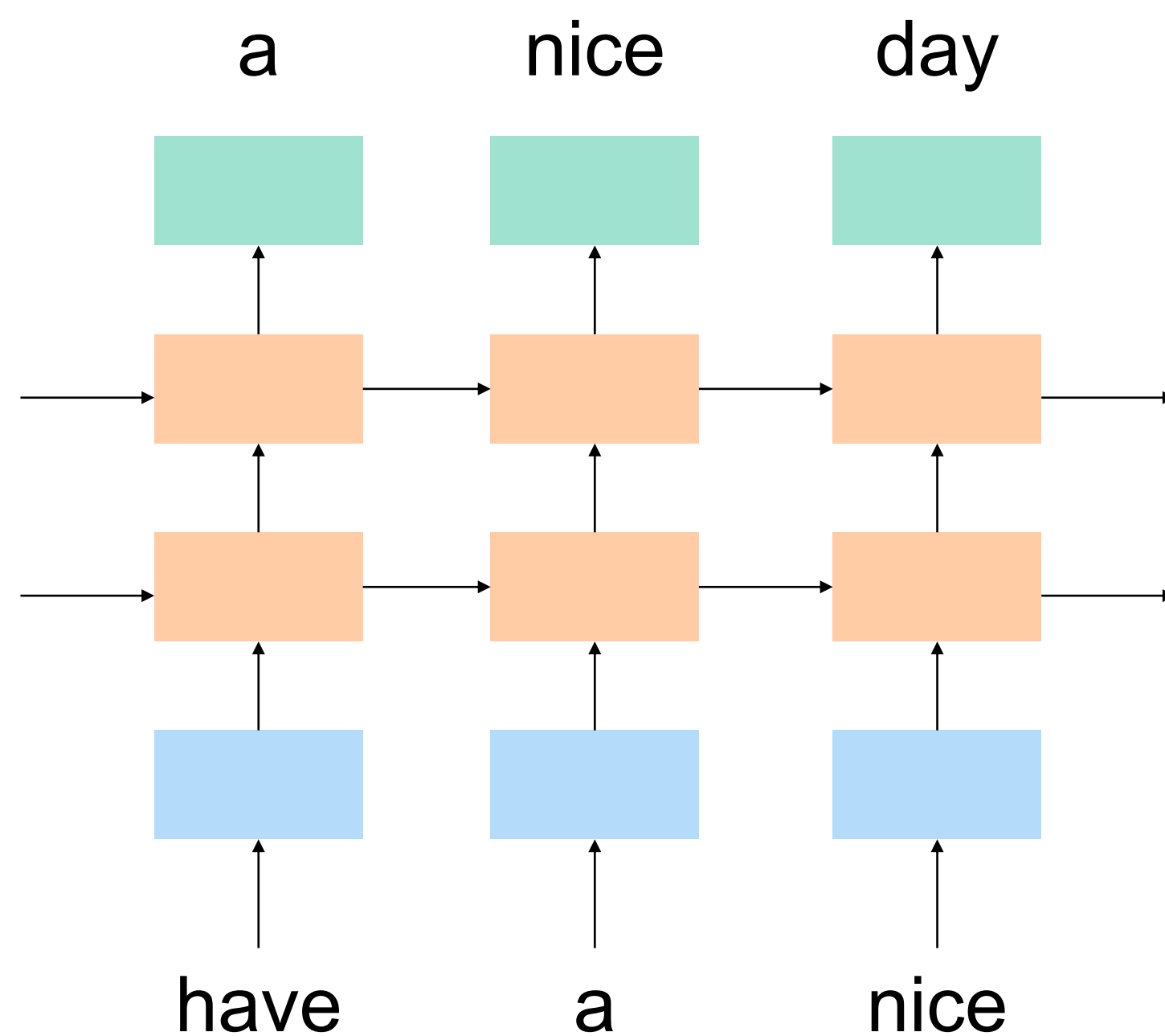
34 Word embedding polysemy issue

- Words are polysemy
 - ✓ An apple a day, keeps the doctor away.
 - ✓ Smartphone companies including apple, ...
- However, their embeddings are NOT polysemy
- Issue
 - ✓ Multi-senses (polysemy)
 - ✓ Multi-aspects (semantics, syntax)



35 ELMo: Embedding from Language Models

- Idea: contextualized word representations
- ✓ Learn word vectors using long contexts instead of a context window
- ✓ Learn a deep Bi-NLM and use all its layers in prediction

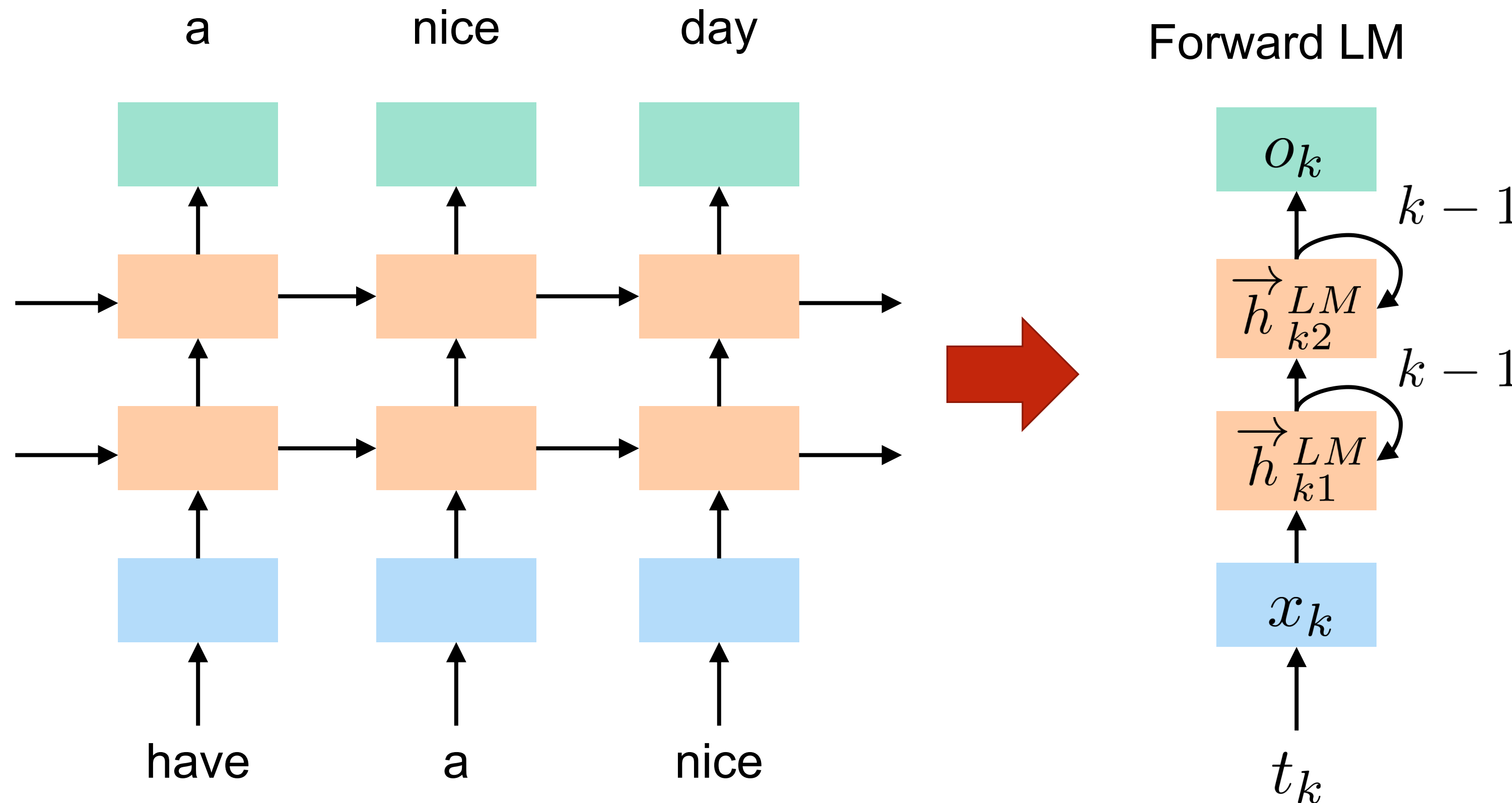


36 ELMo: Embedding from Language Models



1) Bidirectional LM

$$p(t_1, t_2, \dots, t_N) = \prod_{k=1}^N p(t_k | t_1, \dots, t_{k-1})$$



37 ELMo: Embedding from Language Models



1) Bidirectional LM

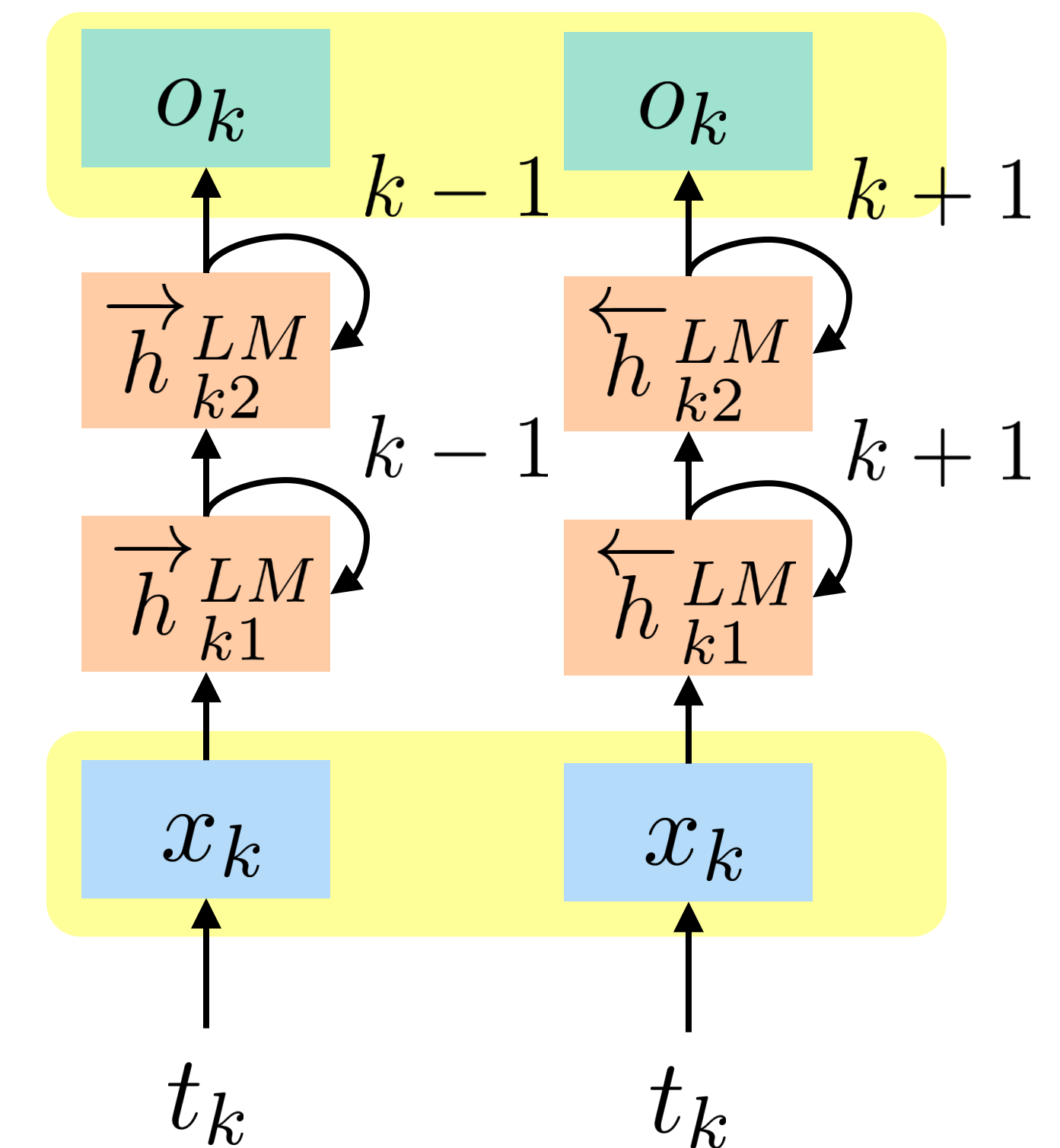
$$p(t_1, t_2, \dots, t_N) = \prod_{k=1}^N p(t_k | t_1, \dots, t_{k-1})$$

$$p(t_1, t_2, \dots, t_N) = \prod_{k=1}^N p(t_k | t_{k+1}, \dots, t_N)$$

- Character CNN for initial word embeddings
2048 n-gram filters, 2 highway layers, 512 dim projection
- 2 BLSTM layers
- Parameter tying for input/output layers

$$O = \sum_{k=1}^N \left(\log p(t_k | t_1, \dots, t_{k-1}; \Theta_x, \vec{\Theta}_{LSTM}, \Theta_s) \right. \\ \left. + \log p(t_k | t_{k+1}, \dots, t_N; \Theta_x, \overleftarrow{\Theta}_{LSTM}, \Theta_s) \right)$$

Forward LM Backward LM



38 ELMo: Embedding from Language Models



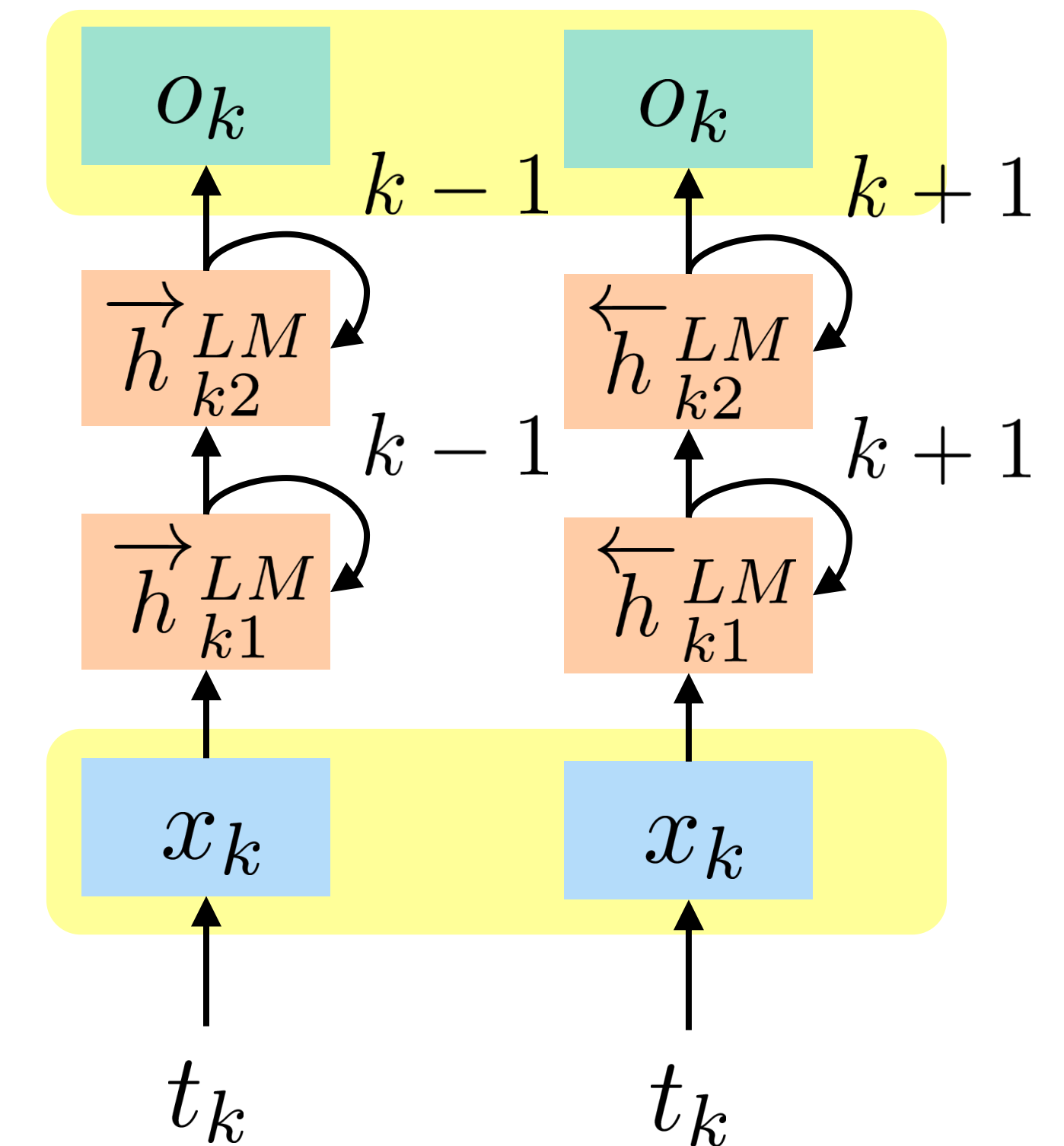
2) ELMo

- Learn task-specific linear combination of LM embeddings
- Use multiple layers in LSTM instead of top one

$$\text{ELMo}_k^{\text{task}} = \gamma^{\text{task}} \times \sum \begin{cases} s_2^{\text{task}} \times h_{k2}^{LM} & \begin{matrix} \vec{h}_{k2}^{LM} & \overleftarrow{h}_{k2}^{LM} \end{matrix} \\ s_1^{\text{task}} \times h_{k1}^{LM} & \begin{matrix} \vec{h}_{k1}^{LM} & \overleftarrow{h}_{k1}^{LM} \end{matrix} \\ s_0^{\text{task}} \times h_{k0}^{LM} & \begin{matrix} x_k & x_k \end{matrix} \end{cases}$$

- γ^{task} scales overall usefulness of ELMo to task
- s^{task} are softmax-normalized weights
- optional layer normalization

Forward LM Backward LM



A task-specific embedding with combining weights learned from a downstream task

39 ELMo: Embedding from Language Models



3) Use ELMo in Supervised NLP Tasks

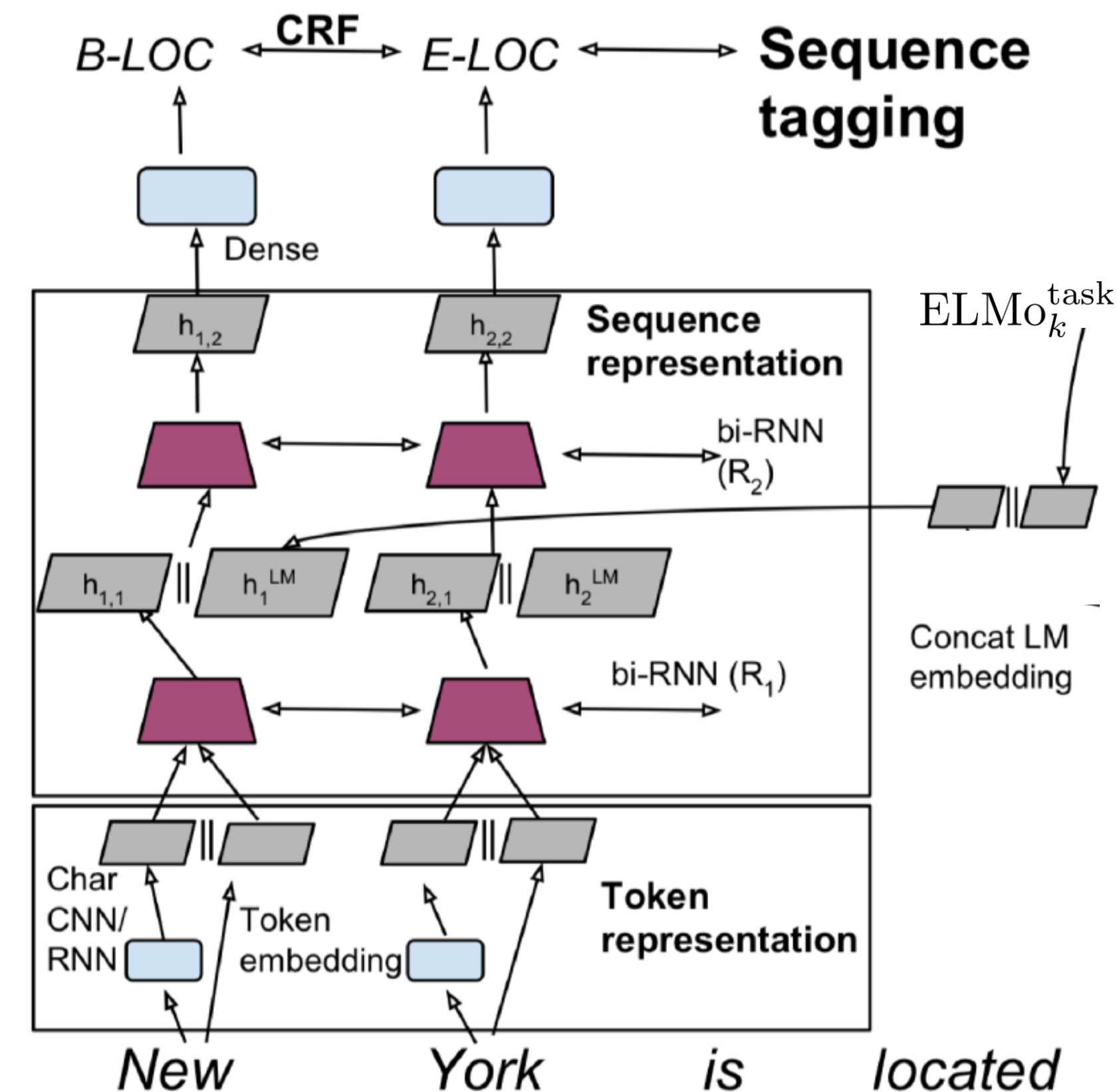
- Get LM embedding for each word
- Freeze LM weights to form ELMo enhanced embeddings

$[h_k; \text{ELMo}_k^{\text{task}}]$: concatenate ELMo into the intermediate layer

$[x_k; \text{ELMo}_k^{\text{task}}]$: concatenate ELMo into the input layer

- Tricks: dropout, regularization

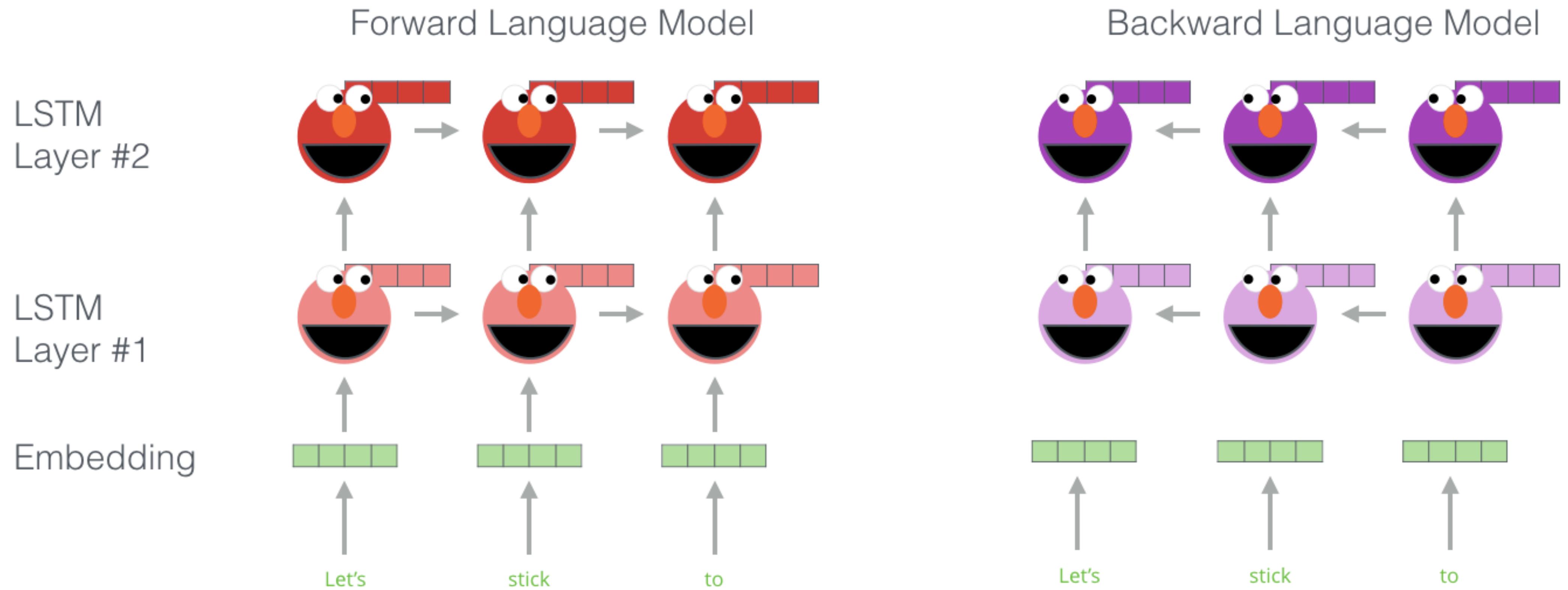
The way for concatenation depends on the task



ELMo illustration



Embedding of “stick” in “Let’s stick to” - Step #1



ELMo illustration



Embedding of "stick" in "Let's stick to" - Step #2

1- Concatenate hidden layers



2- Multiply each vector by a weight based on the task

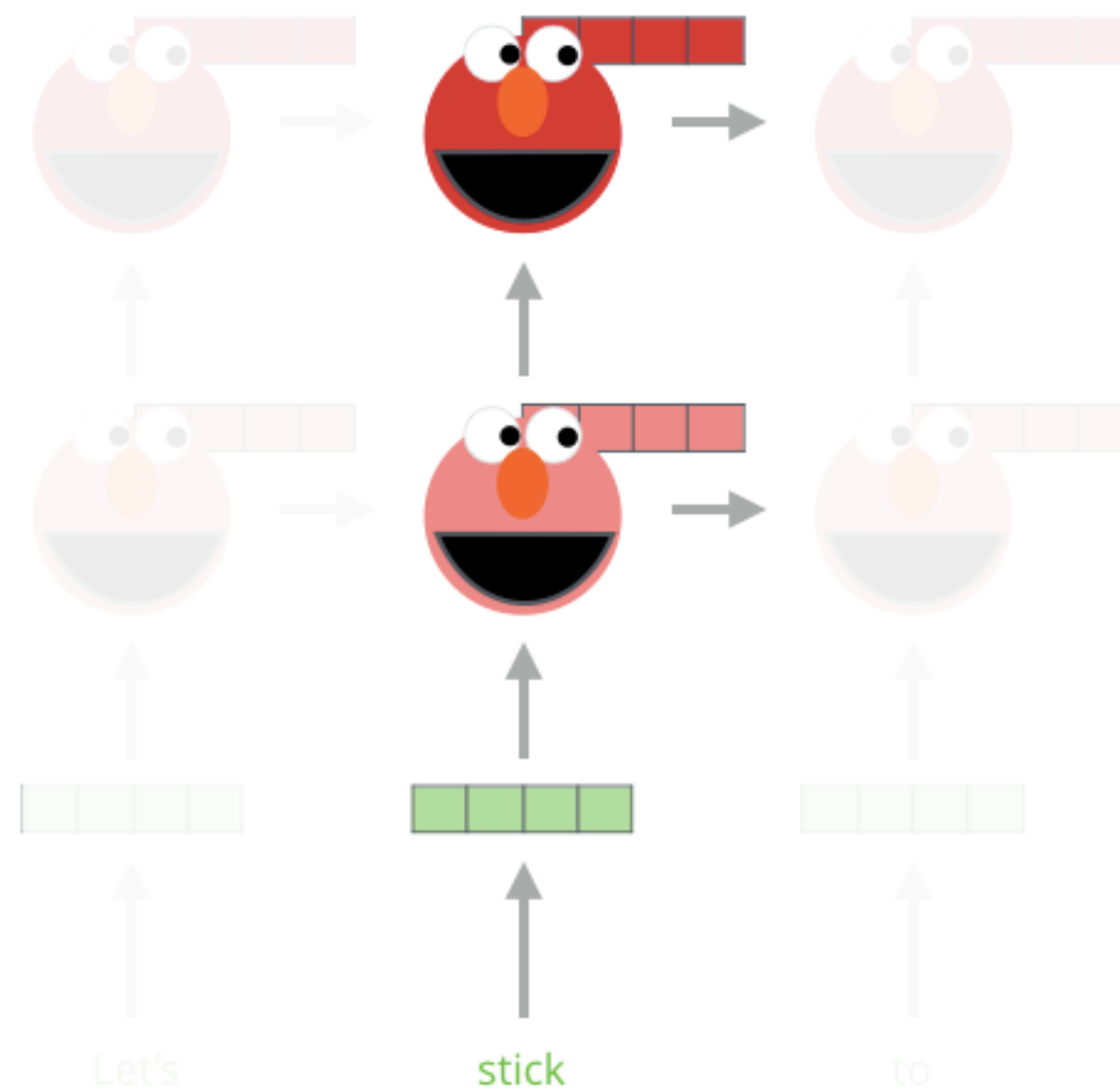


3- Sum the (now weighted) vectors

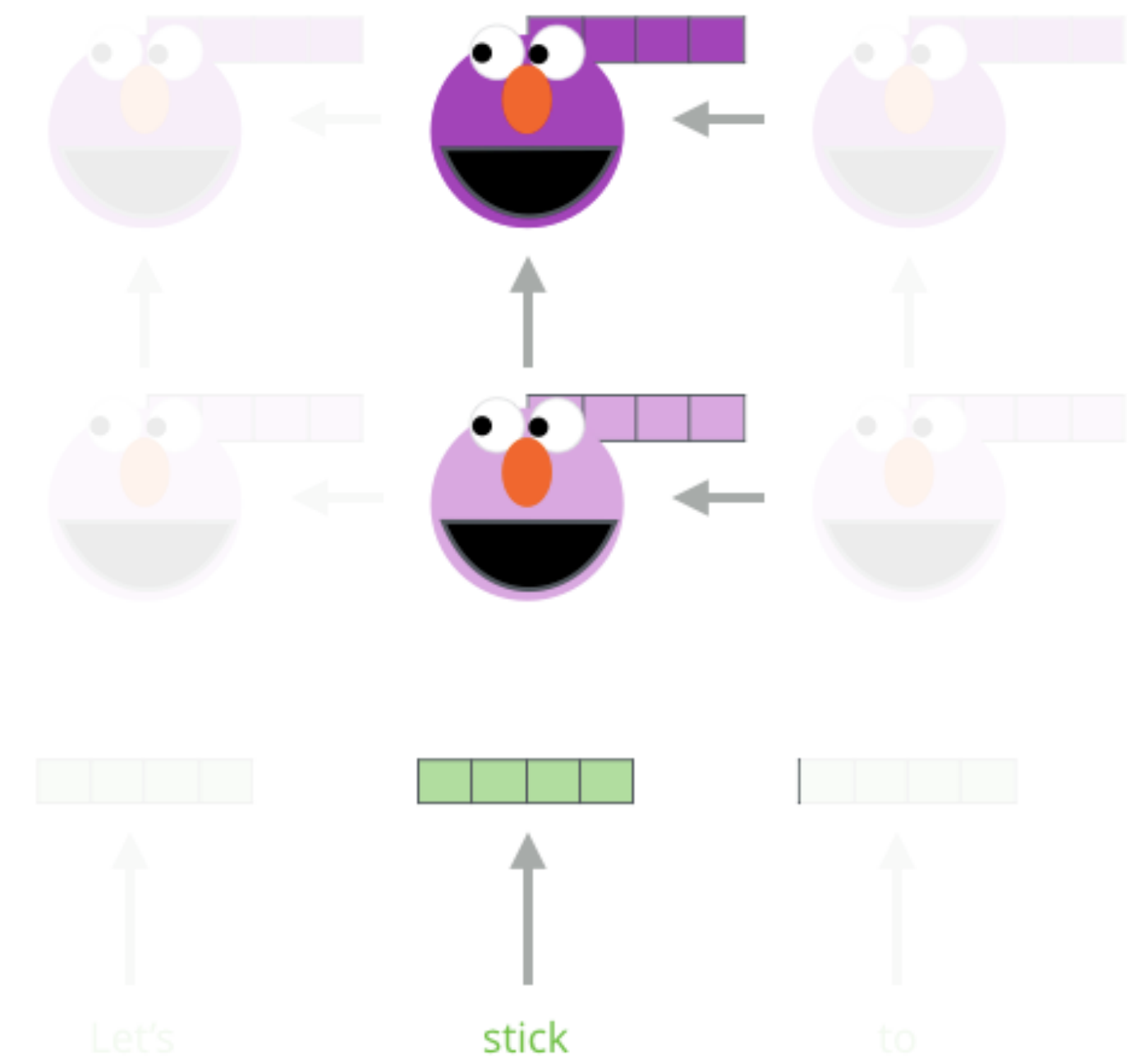


ELMo embedding of "stick" for this task in this context

Forward Language Model



Backward Language Model





Model	Description	CONLL 2003 F1
Klein+, 2003	MEMM softmax markov model	86.07
Florian+, 2003	Linear/softmax/TBL/HMM	88.76
Finkel+, 2005	Categorical feature CRF	86.86
Ratinov and Roth, 2009	CRF+Wiki+Word cls	90.80
Peters+, 2017	BLSTM + char CNN + CRF	90.87
Ma and Hovy, 2016	BLSTM + char CNN + CRF	91.21
TagLM (Peters+, 2017)	LSTM BiLM in BLSTM Tagger	91.93
ELMo (Peters+, 2018)	ELMo in BLSTM	92.22

43 ELMo results



Improvement on various NLP tasks

	TASK	PREVIOUS SOTA		OUR BASELINE	ELMo + BASELINE	INCREASE (ABSOLUTE/ RELATIVE)
Machine Comprehension	SQuAD	Liu et al. (2017)	84.4	81.1	85.8	4.7 / 24.9%
Textual Entailment	SNLI	Chen et al. (2017)	88.6	88.0	88.7 ± 0.17	0.7 / 5.8%
Semantic Role Labeling	SRL	He et al. (2017)	81.7	81.4	84.6	3.2 / 17.2%
Coreference Resolution	Coref	Lee et al. (2017)	67.2	67.2	70.4	3.2 / 9.8%
Name Entity Recognition	NER	Peters et al. (2017)	91.93 ± 0.19	90.15	92.22 ± 0.10	2.06 / 21%
Sentiment Analysis	SST-5	McCann et al. (2017)	53.7	51.4	54.7 ± 0.5	3.3 / 6.8%

Good transfer learning in NLP (similar to computer vision)

44 ELMo analysis



Word embeddings v.s. contextualized embeddings

	Source	Nearest Neighbors
GloVe	play	playing, game, games, played, players, plays, player, Play, football, multiplayer
biLM	Chico Ruiz made a spectacular <u>play</u> on Alusik 's grounder {...}	Kieffer , the only junior in the group , was commended for his ability to hit in the clutch , as well as his all-round excellent <u>play</u> .
	Olivia De Havilland signed to do a Broadway <u>play</u> for Garson {...}	{...} they were actors who had been handed fat roles in a successful <u>play</u> , and had talent enough to fill the roles competently , with nice understatement .

The biLM is able to disambiguate both the PoS and word sense in the source sentence



- The two NLM layers have differentiated uses/meanings
 - ✓ Lower layer is better for lower-level **syntax**, etc. (e.g. Part-of-speech tagging, syntactic dependencies, NER)
 - ✓ Higher layer is better for higher-level **semantics** (e.g. sentiment, semantic role labeling, question answering, SNLI)

PoS Tagging

Model	Acc.
Collobert et al. (2011)	97.3
Ma and Hovy (2016)	97.6
Ling et al. (2015)	97.8
CoVe, First Layer	93.3
CoVe, Second Layer	92.8
biLM, First Layer	97.3
biLM, Second Layer	96.8

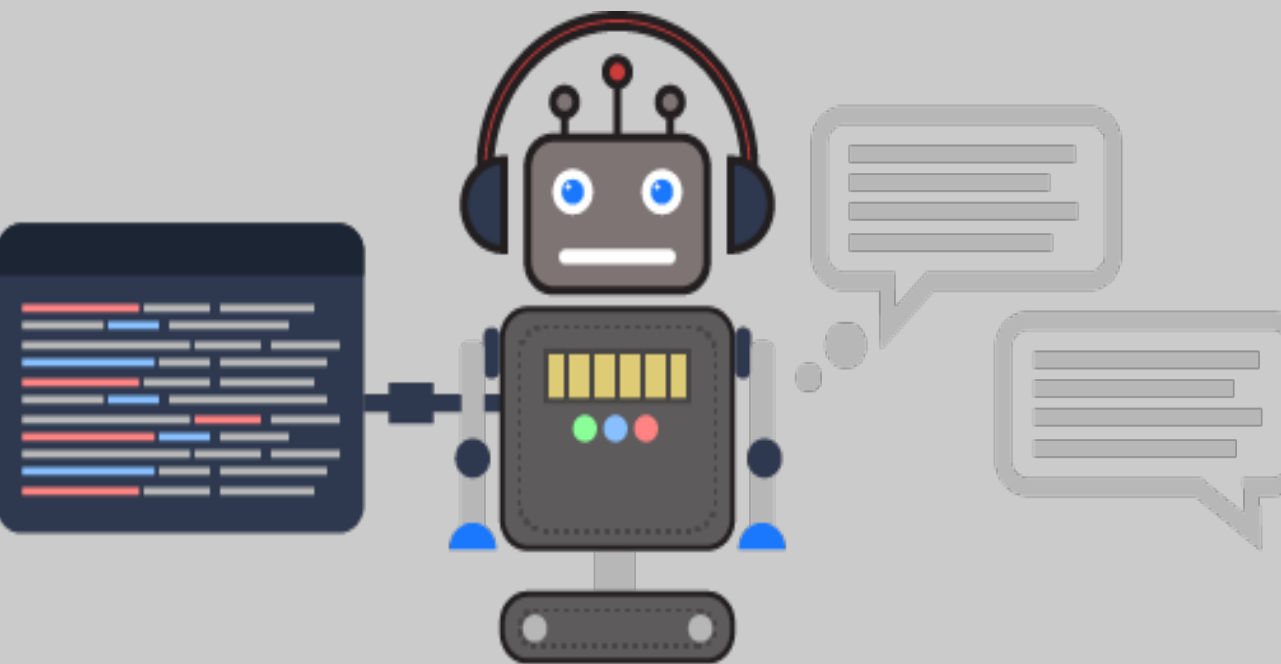
Word Sense Disambiguation

Model	F ₁
WordNet 1st Sense Baseline	65.9
Raganato et al. (2017a)	69.9
Iacobacci et al. (2016)	70.1
CoVe, First Layer	59.4
CoVe, Second Layer	64.7
biLM, First layer	67.4
biLM, Second layer	69.0

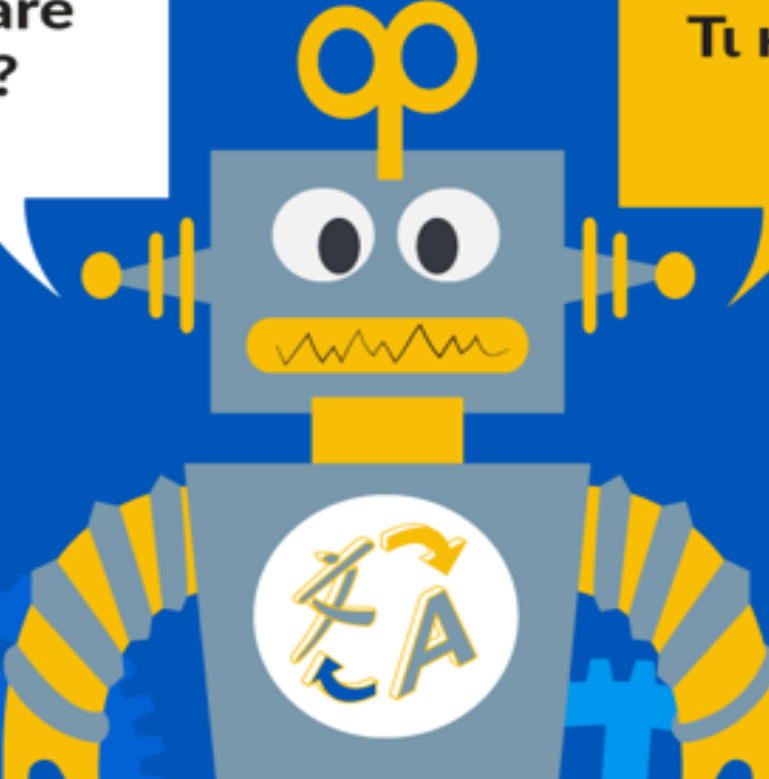
Structure Matters: Hierarchical Sentence Factorization



Semantic matching



Chatbot query response



How are you?
Τι κάνετε;


Translation source target

I just need the main ideas



long short

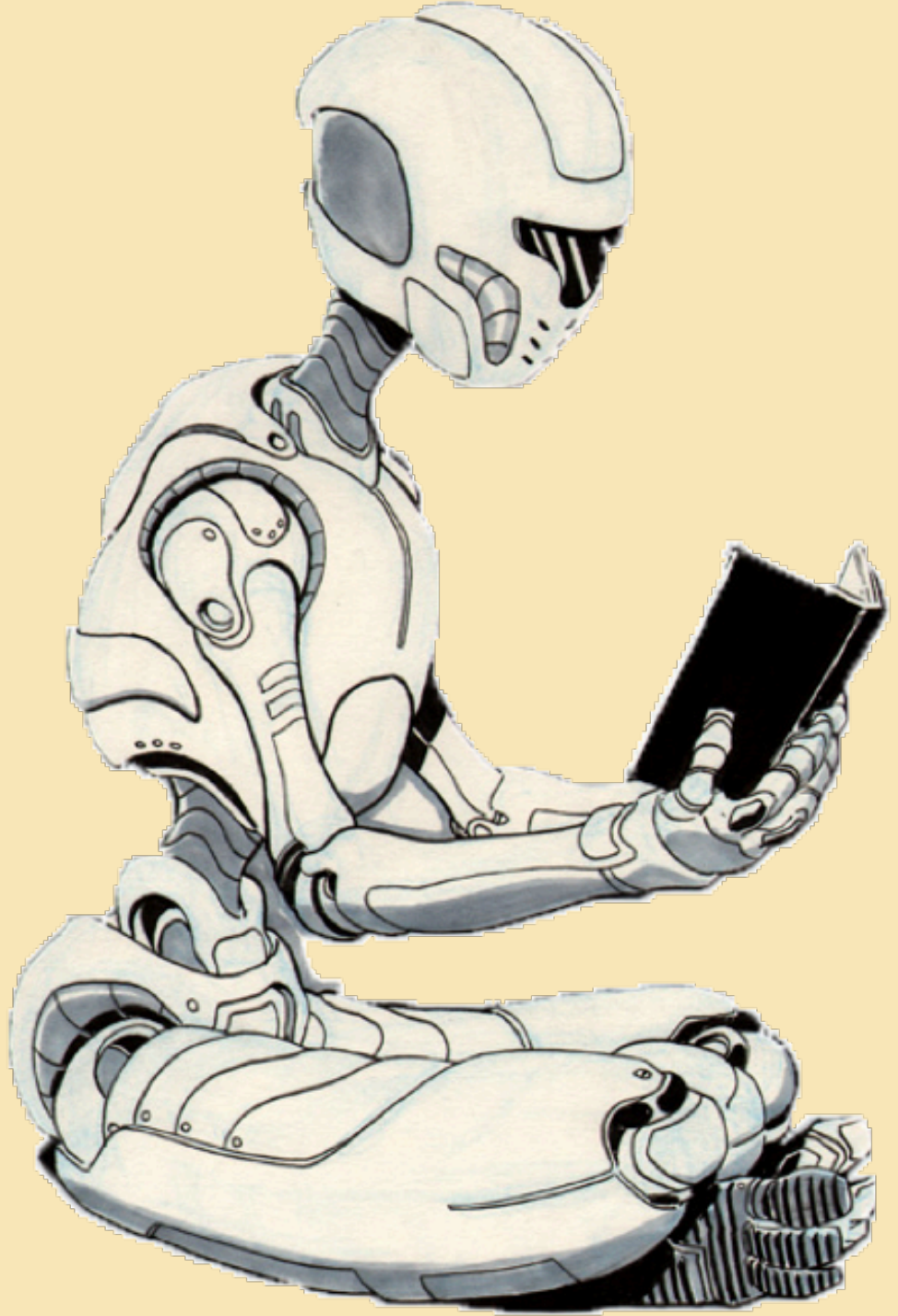
Summarization



question answer

Question Answering

passages + question answer



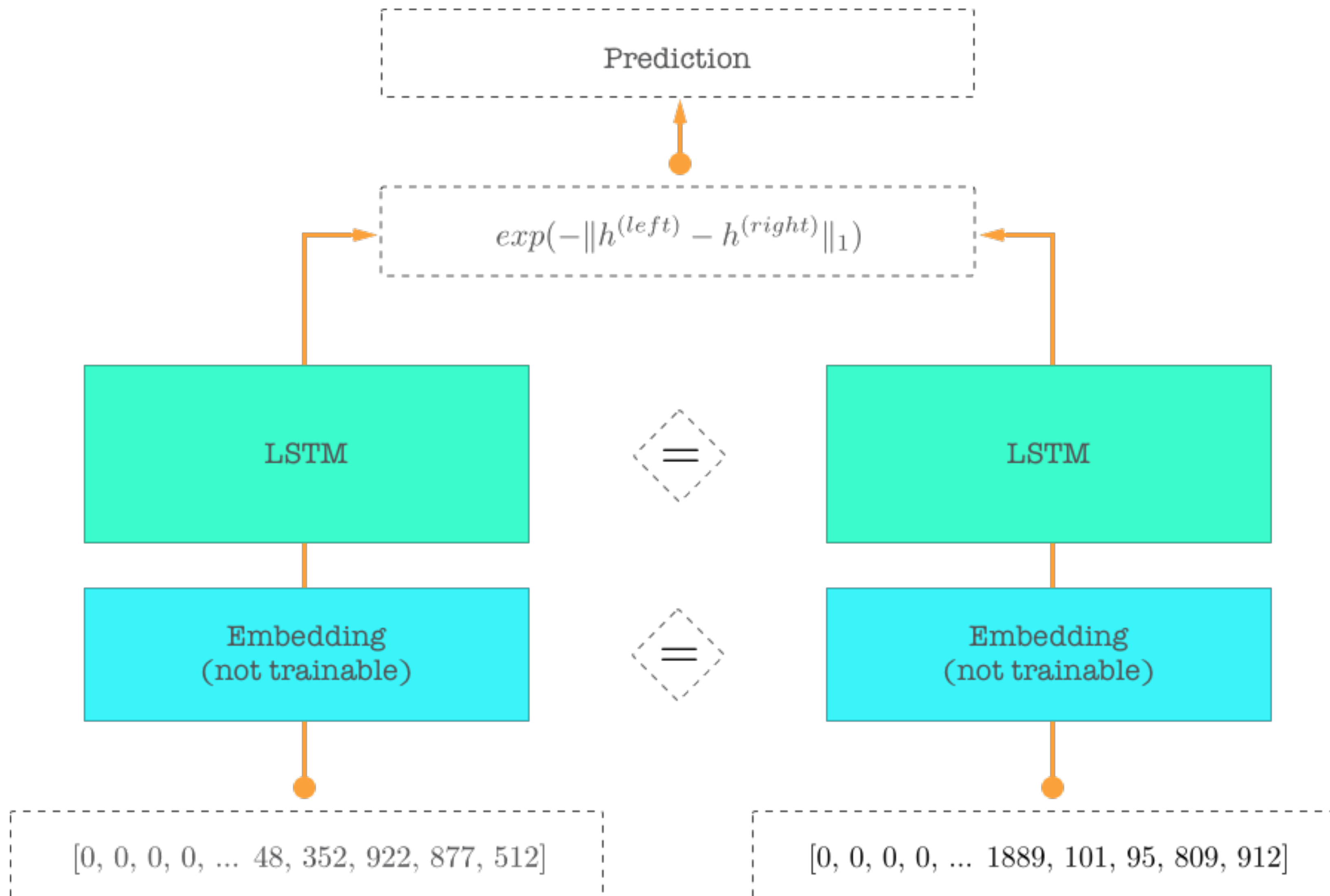
Reading Comprehension

Semantic similarity estimation

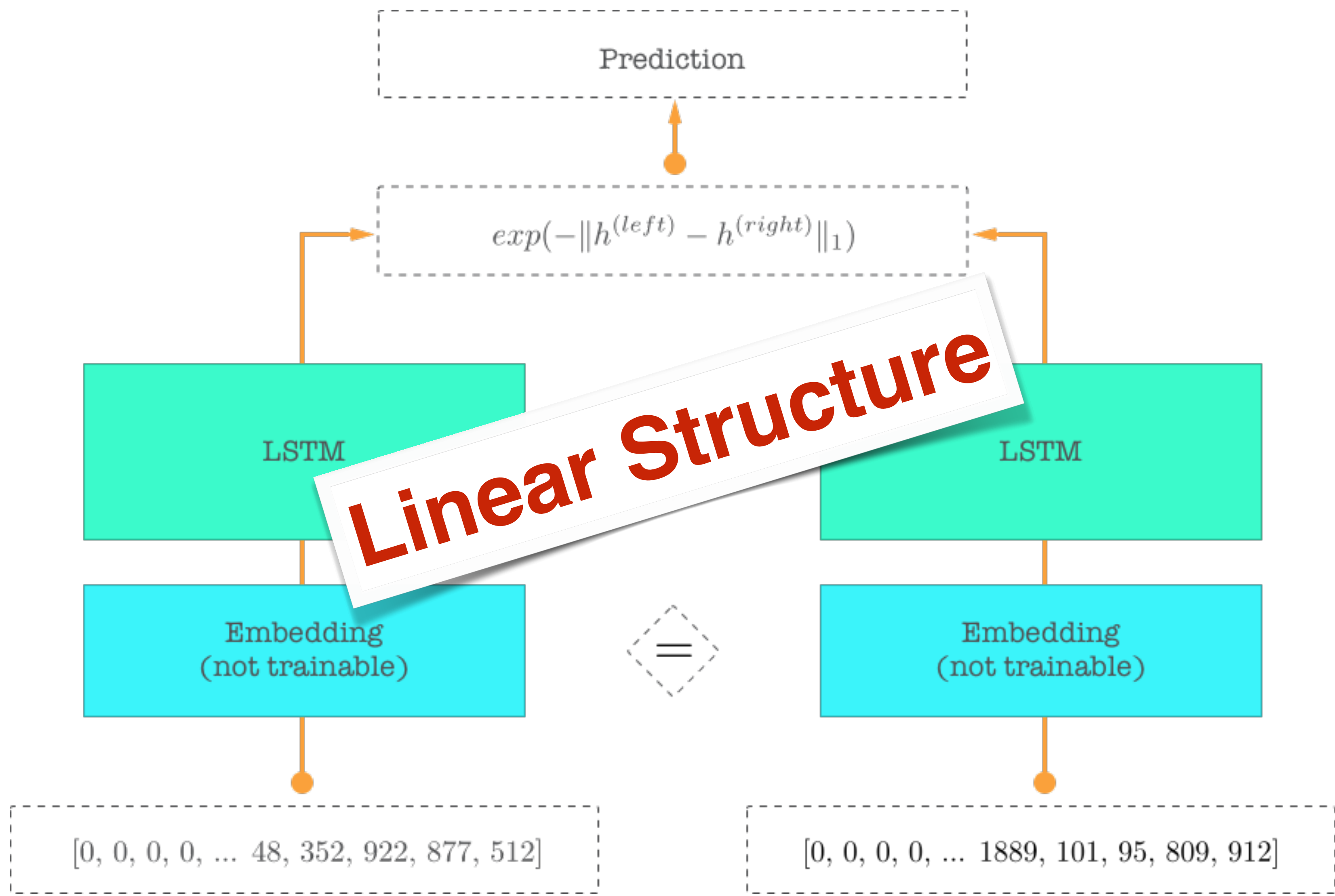
Degree of semantic similarity between two sentences

- (5) Completely equivalent:** they mean the same thing
- (4) Mostly equivalent:** some unimportant details differ.
- (3) Roughly equivalent:** some important information differs/missing.
- (2) Not equivalent:** share some details.
- (1) Not equivalent:** on the same topic.
- (0) On different topics.**

Siamese neural network



Siamese neural network



Natural Language is **Flexible**

Sentence A:

The blue cat is chasing the brown mouse.

Sentence B:

The brown mouse is being chased by the blue cat.

Natural Language is **Flexible**

Sentence A:

The blue cat is chasing the brown mouse.

Sentence B:

The brown mouse is being chased by the blue cat.

Normalized sentence:

chase **blue cat** **brown mouse.**

Predicate

Argument 0

Argument 1

Natural Language is **Compositional**

Sentence A:

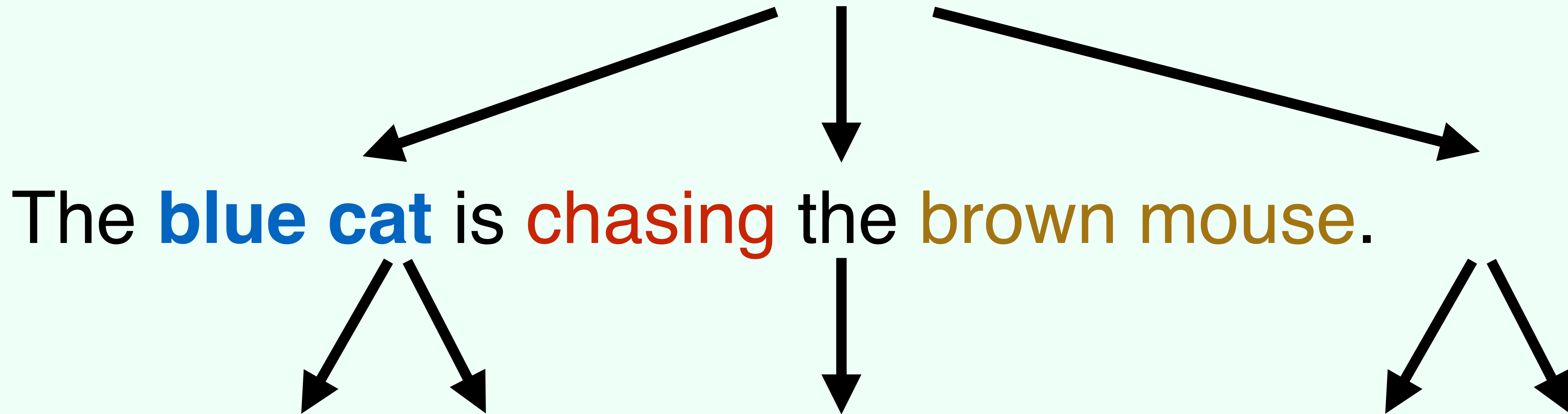
The **blue cat** is **chasing** the **brown mouse**.

Semantic Units: word, phrase, sentence

Natural Language is **Hierarchical**

Sentence A:

The **blue cat** is chasing the brown mouse.



The **blue cat** is chasing the brown mouse.

55

Input sentence pair

Sentence A: The **little Jerry** is **being chased** by **Tom** in the **big yard**.

Sentence B: The **blue cat** is **catching** the **brown mouse** in the **forecourt**.

Flexible: *normalize order*

Compositional: *factorize semantic units*

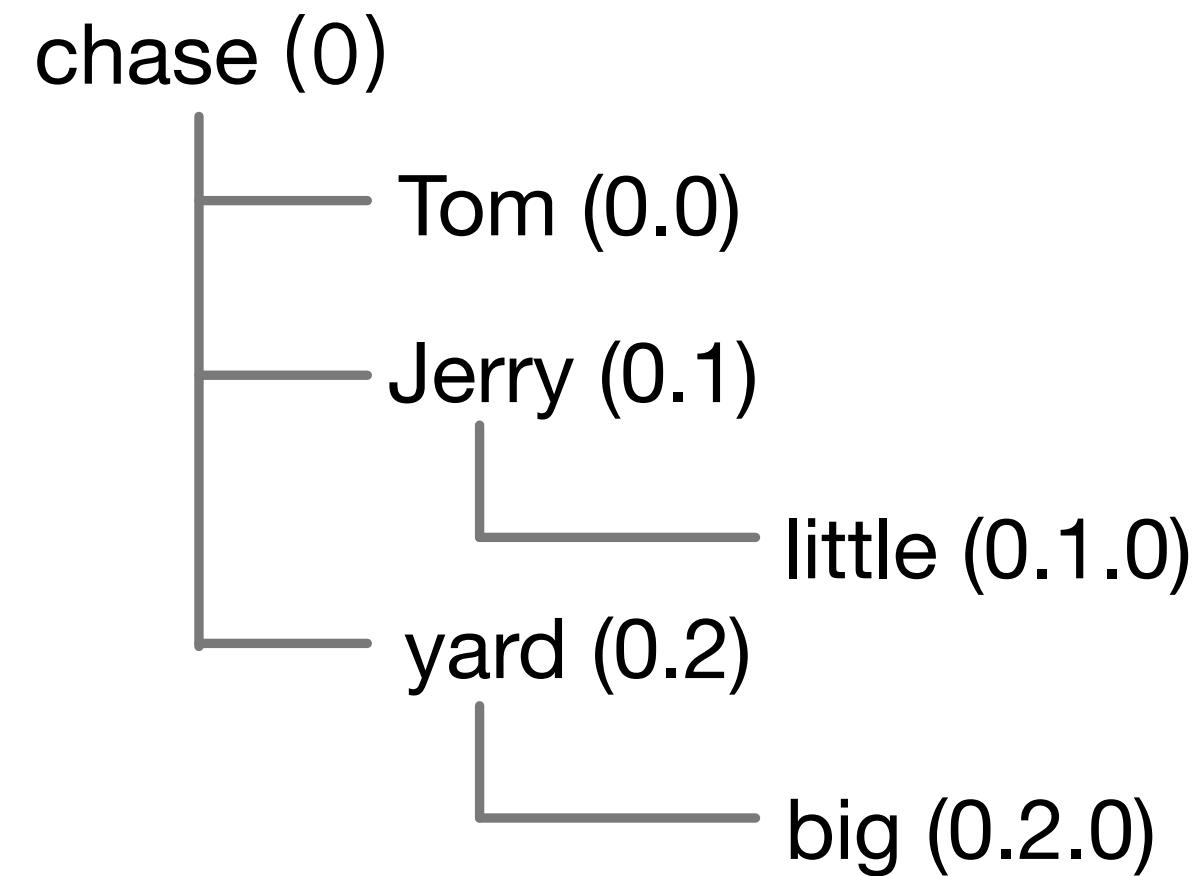
Hierarchical: *multi-layer factorization*

56 Hierarchical sentence factorization

Sentence A: The **little Jerry** is **being chased** by **Tom** in the **big yard**.

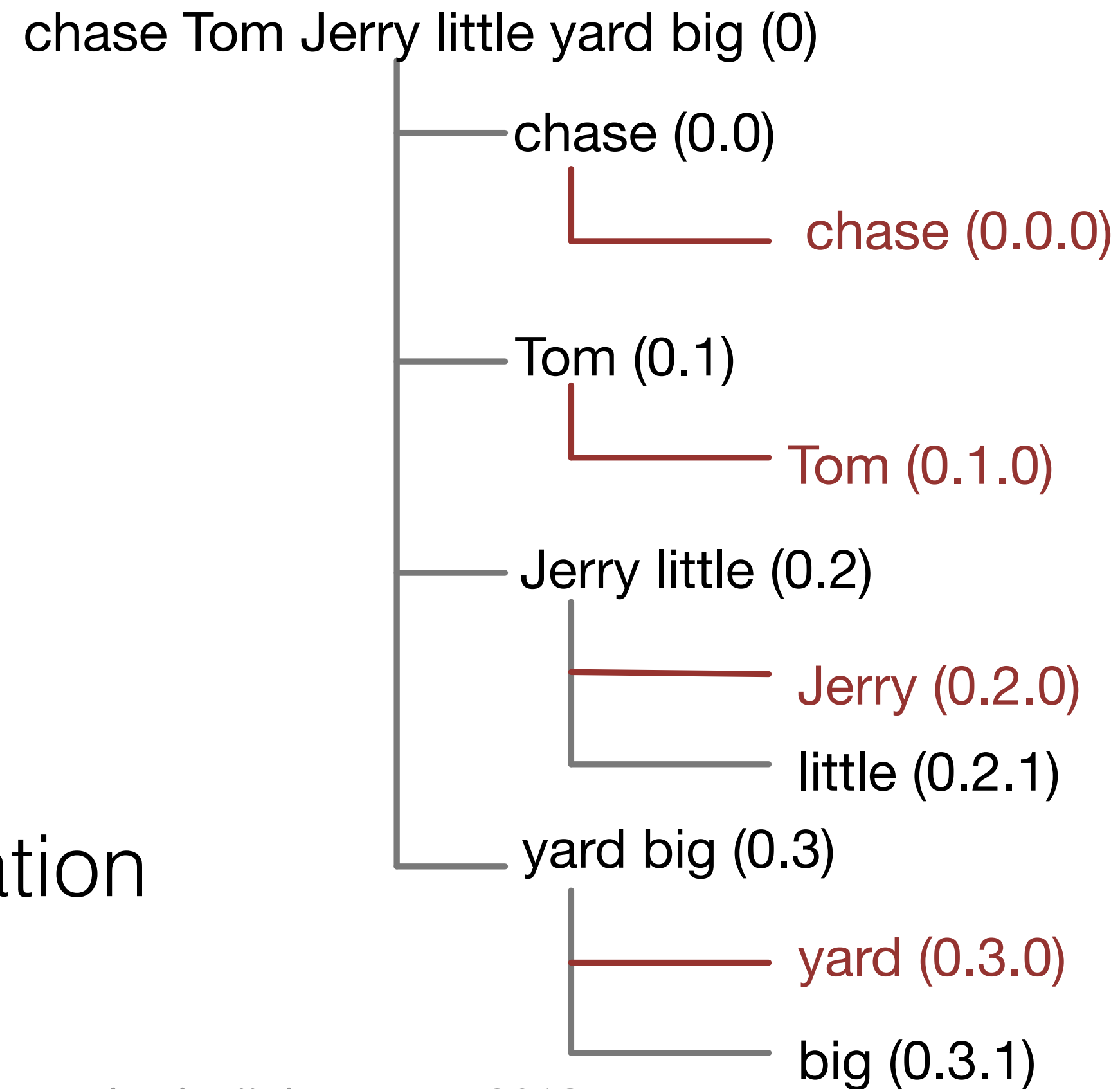
AMR Purification

(a1)



Node Traversal

(a4)



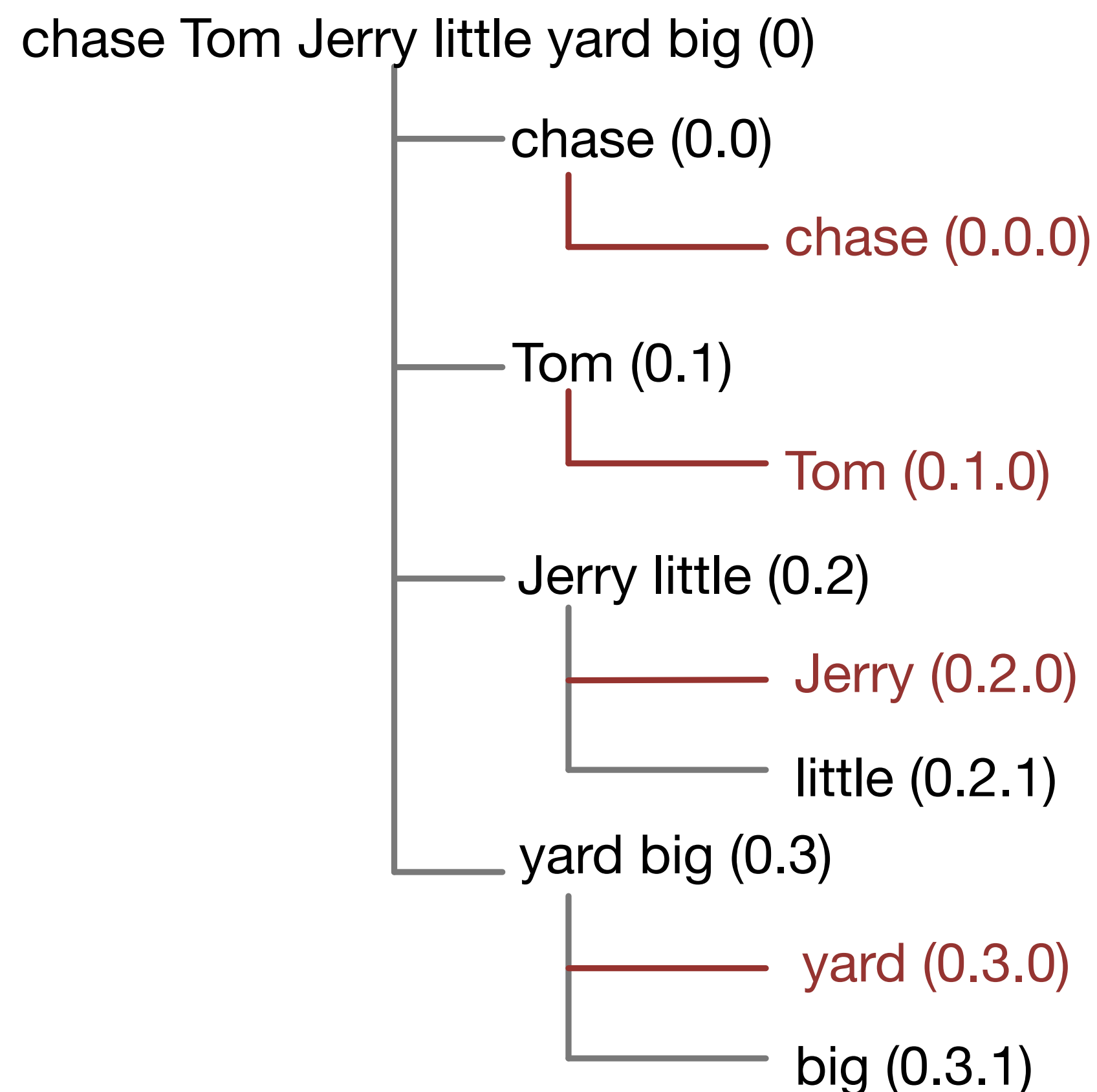
AMR:

Abstract Meaning Representation

57 Alignment

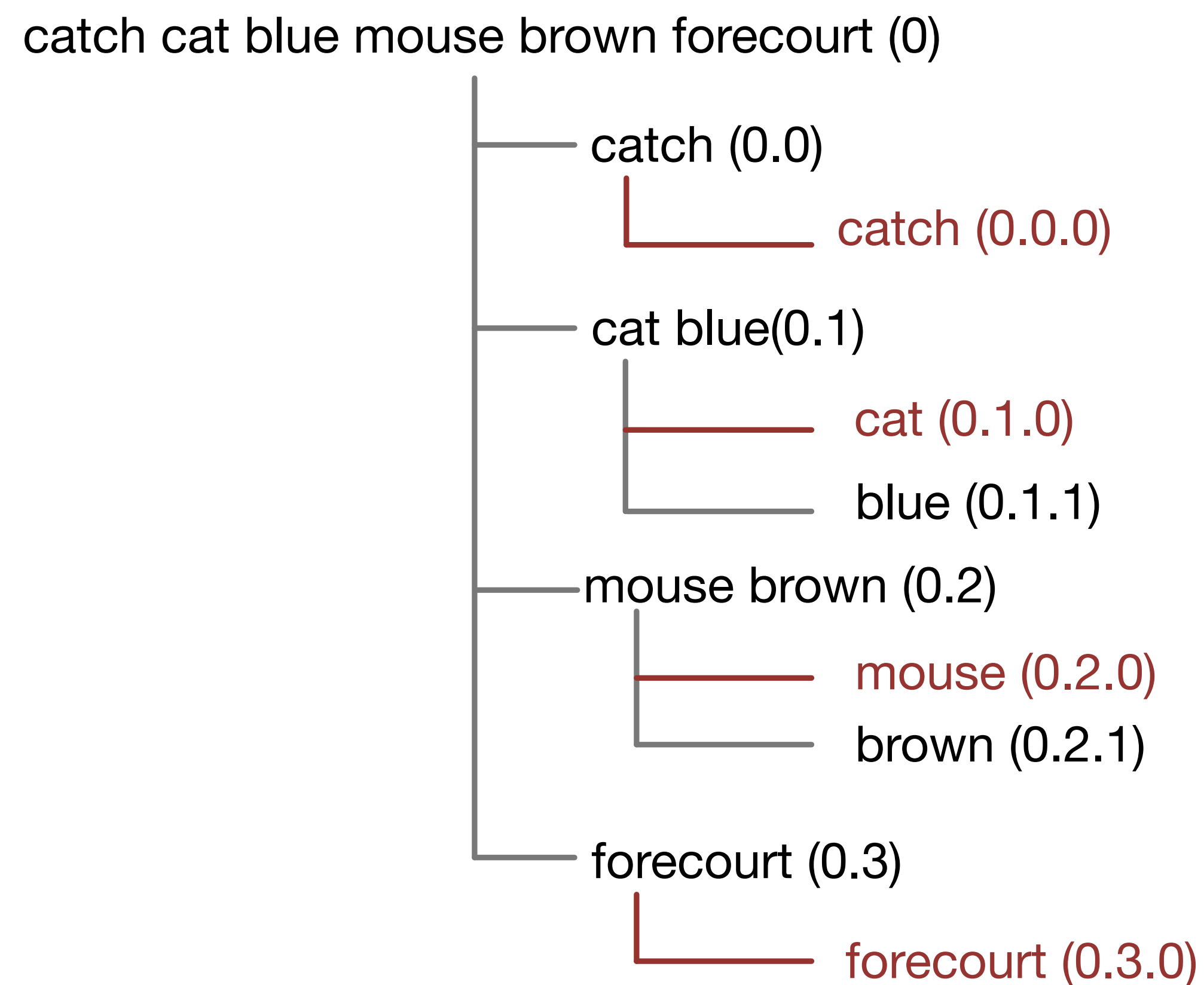
Sentence A: The **little Jerry** is **being chased** by **Tom** in the **big yard**.

(a4)

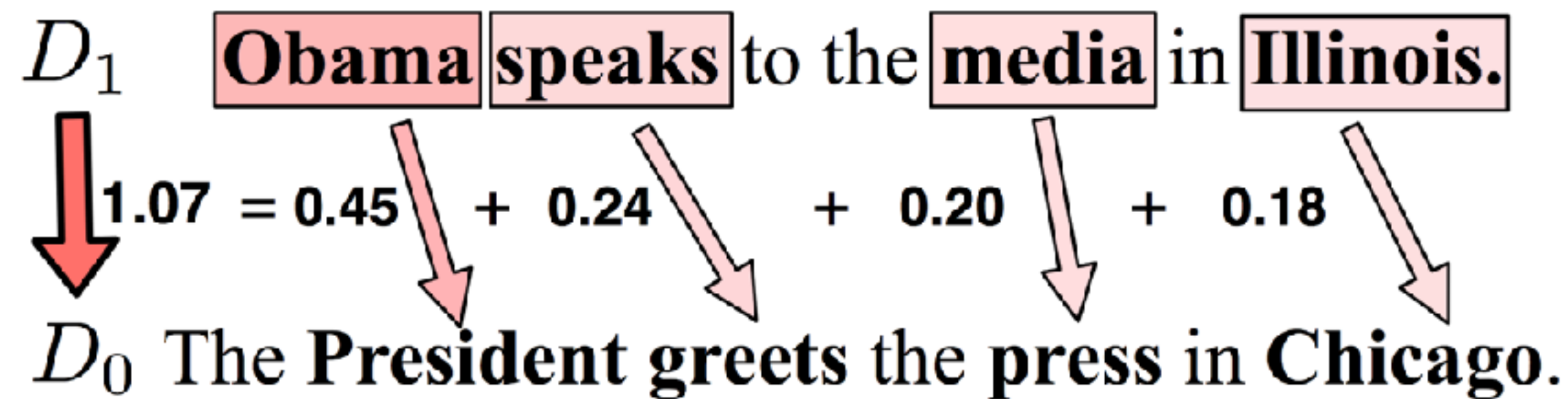


Sentence B: The **blue cat** is **catching** the **brown mouse** in the **forecourt**.

(b4)



Word Mover's Distance



$$\begin{aligned}
 &\text{minimize}_{T \in \mathbb{R}_+^{M \times N}} \sum_{i,j} T_{ij} D_{ij} \\
 &\text{subject to} \quad \sum_{i=1}^M T_{ij} = \beta_j \quad 1 \leq j \leq N, \\
 &\quad \quad \quad \sum_{j=1}^N T_{ij} = \alpha_i \quad 1 \leq i \leq M.
 \end{aligned}$$

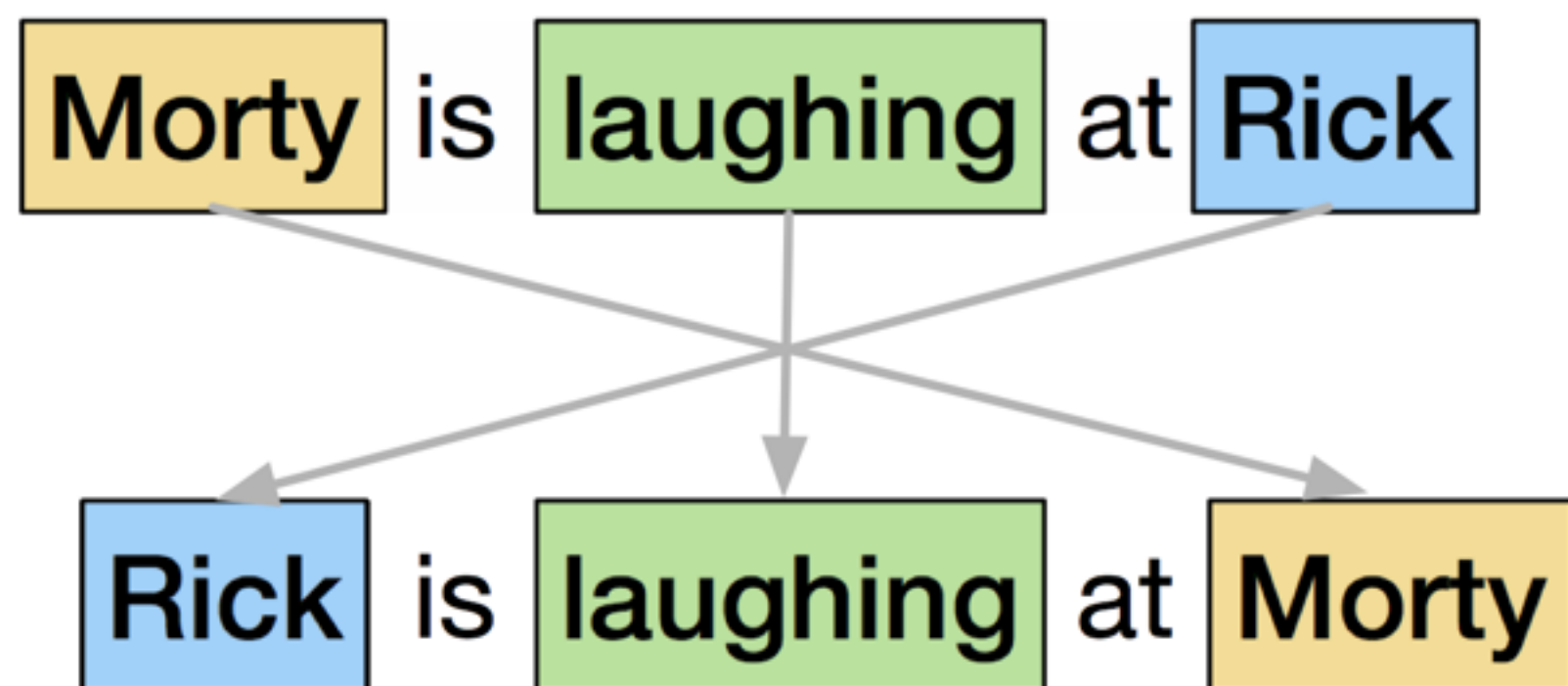
$\alpha = \{\alpha_1, \dots, \alpha_M\}$: normalized bag-of-words vector of S_1

$\beta = \{\beta_1, \dots, \beta_N\}$: normalized bag-of-words vector of S_2

D_{ij} : distance between word $i \in S_1$ and word $j \in S_2$

T_{ij} : the portion of word $i \in S_1$ that transports to word $j \in S_2$

Word Mover's Distance



$$\begin{aligned}
 & \text{minimize}_{T \in \mathbb{R}_+^{M \times N}} \sum_{i,j} T_{ij} D_{ij} \\
 & \text{subject to} \quad \sum_{i=1}^M T_{ij} = \beta_j \quad 1 \leq j \leq N, \\
 & \quad \quad \quad \sum_{j=1}^N T_{ij} = \alpha_i \quad 1 \leq i \leq M.
 \end{aligned}$$

$\alpha = \{\alpha_1, \dots, \alpha_M\}$: normalized bag-of-words vector of S_1

$\beta = \{\beta_1, \dots, \beta_N\}$: normalized bag-of-words vector of S_2

D_{ij} : distance between word $i \in S_1$ and word $j \in S_2$

T_{ij} : the portion of word $i \in S_1$ that transports to word $j \in S_2$

60 Ordered Word Mover's Distance

Morty is laughing at Rick

WMD Matching

Rick is laughing at Morty

OWMD Matching

Morty is laughing at Rick

Inverse difference moment:

prior distribution for values in T:

Distance from point (i, j) to diagonal line:

$$\begin{aligned} & \text{minimize}_{T \in \mathbb{R}_+^{M \times N}} \sum_{i,j} T_{ij} D_{ij} - \lambda_1 I(T) + \lambda_2 KL(T||P) \\ & \text{subject to} \sum_{i=1}^M T_{ij} = \beta'_j \quad 1 \leq j \leq N', \\ & \sum_{j=1}^N T_{ij} = \alpha'_i \quad 1 \leq i \leq M' \end{aligned}$$

$$I(T) = \sum_{i=1}^{M'} \sum_{j=1}^{N'} \frac{T_{ij}}{\left(\frac{i}{M'} - \frac{j}{N'}\right)^2 + 1}$$

$$P_{ij} = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{l^2(i,j)}{2\sigma^2}}$$

$$l(i, j) = \frac{|i/M' - j/N'|}{\sqrt{1/M'^2 + 1/N'^2}}$$

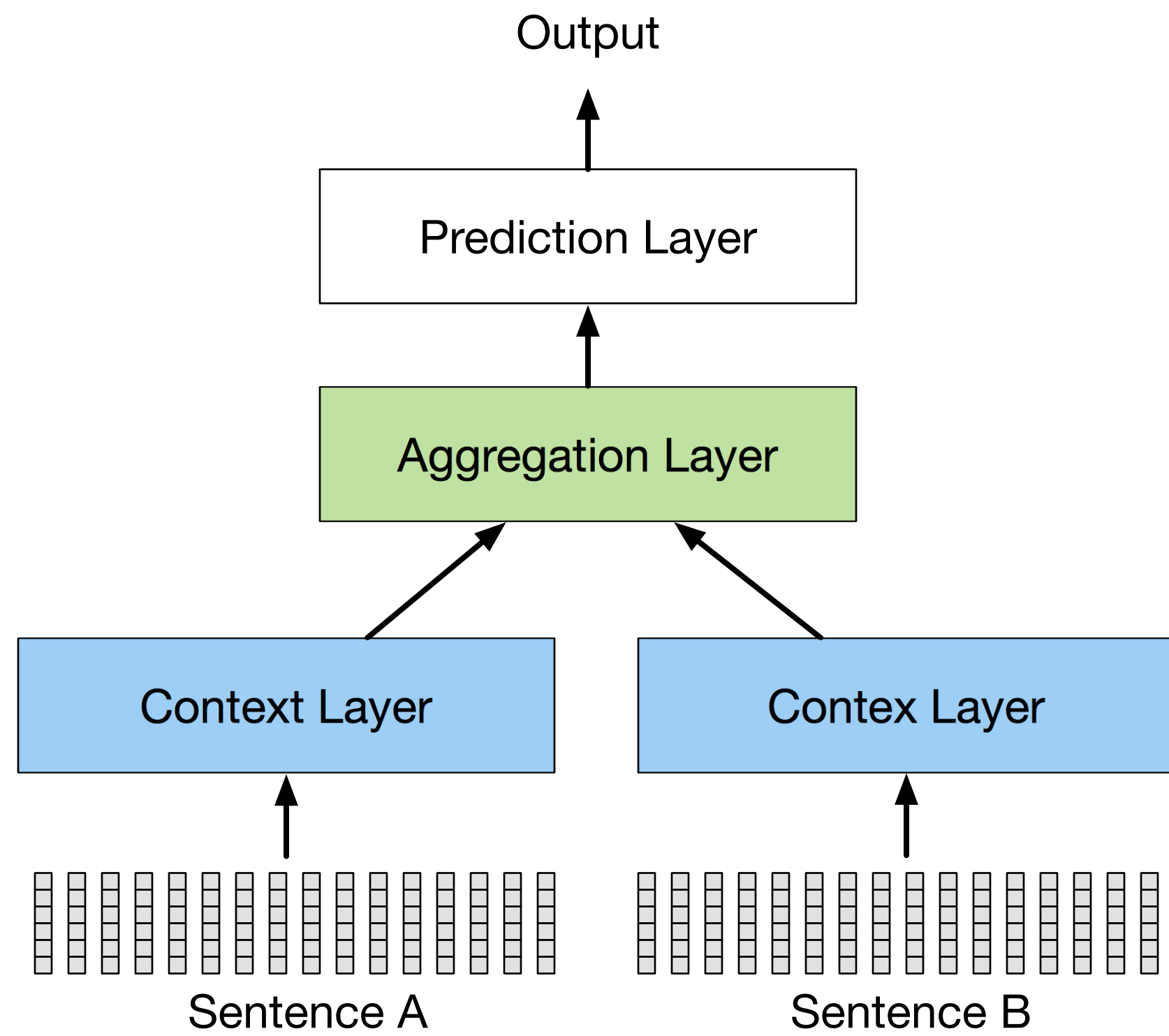
Table 2: Pearson Correlation results on different distance metrics.

Algorithm	STSbenchmark		SICK		MSRvid
	Test	Dev	Test	Dev	Test
BoW	0.5705	0.6561	0.6114	0.6087	0.5044
LexVec	0.5759	0.6852	0.6948	0.6811	0.7318
GloVe	0.4064	0.5207	0.6297	0.5892	0.5481
Fastext	0.5079	0.6247	0.6517	0.6421	0.5517
Word2vec	0.5550	0.6911	0.7021	0.6730	0.7209
WMD	0.4241	0.5679	0.5962	0.5953	0.3430
OWMD	0.6144	0.7240	0.6797	0.6772	0.7519

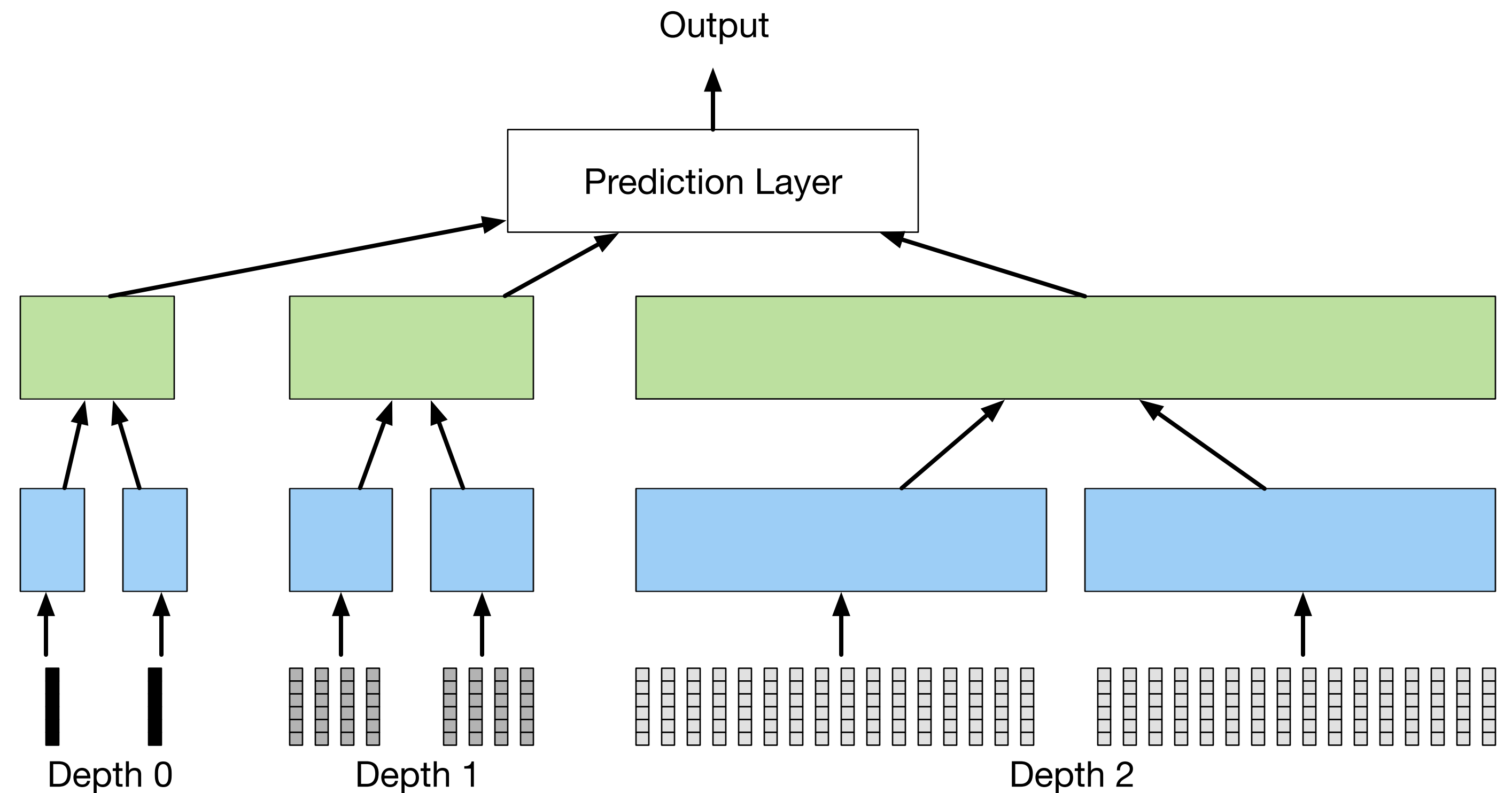
Table 3: Spearman’s Rank Correlation results on different distance metrics.

Algorithm	STSbenchmark		SICK		MSRvid
	Test	Dev	Test	Dev	Test
BoW	0.5592	0.6572	0.5727	0.5894	0.5233
LexVec	0.5472	0.7032	0.5872	0.5879	0.7311
GloVe	0.4268	0.5862	0.5505	0.5490	0.5828
Fastext	0.4874	0.6424	0.5739	0.5941	0.5634
Word2vec	0.5184	0.7021	0.6082	0.6056	0.7175
WMD	0.4270	0.5781	0.5488	0.5612	0.3699
OWMD	0.5855	0.7253	0.6133	0.6188	0.7543

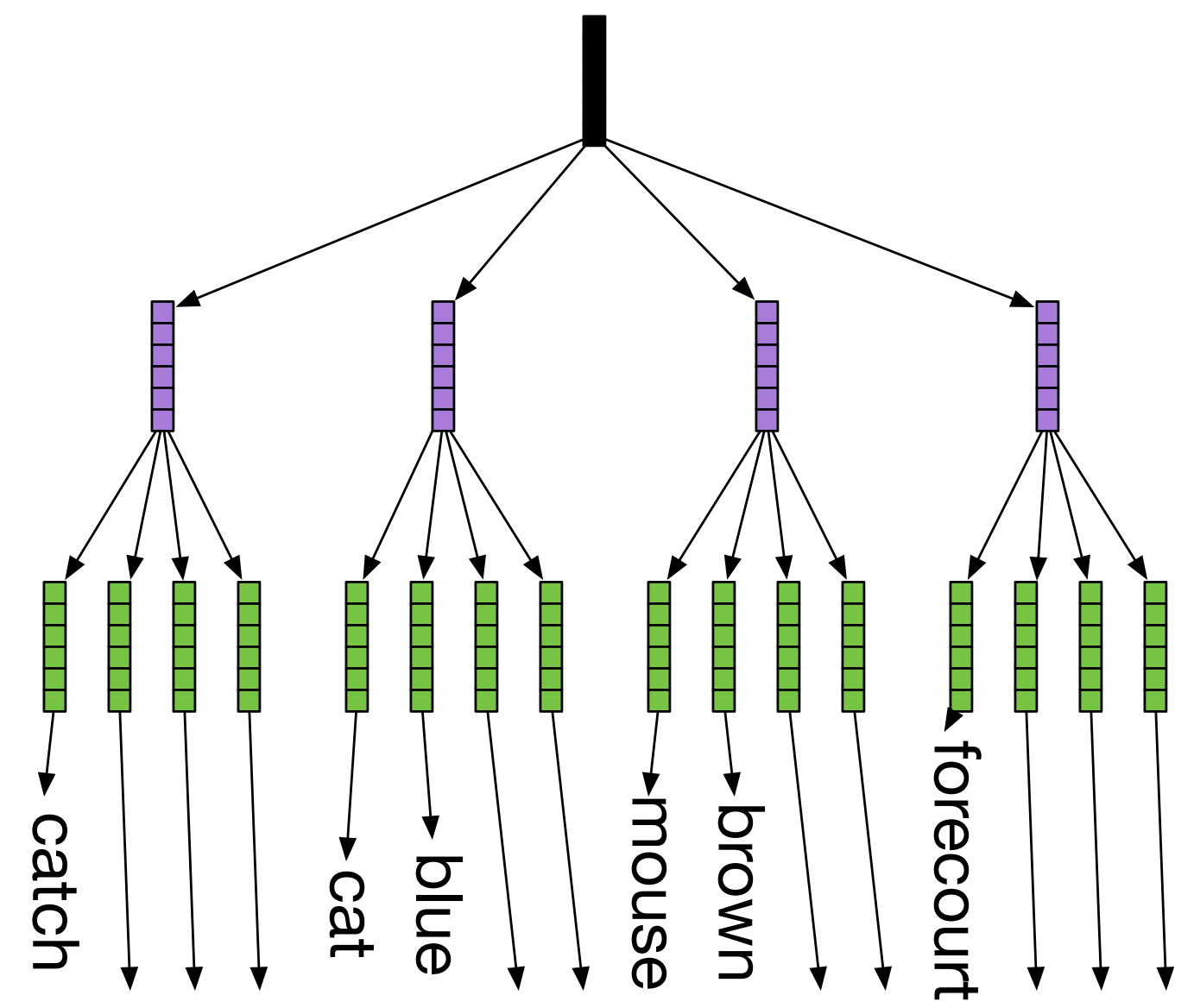
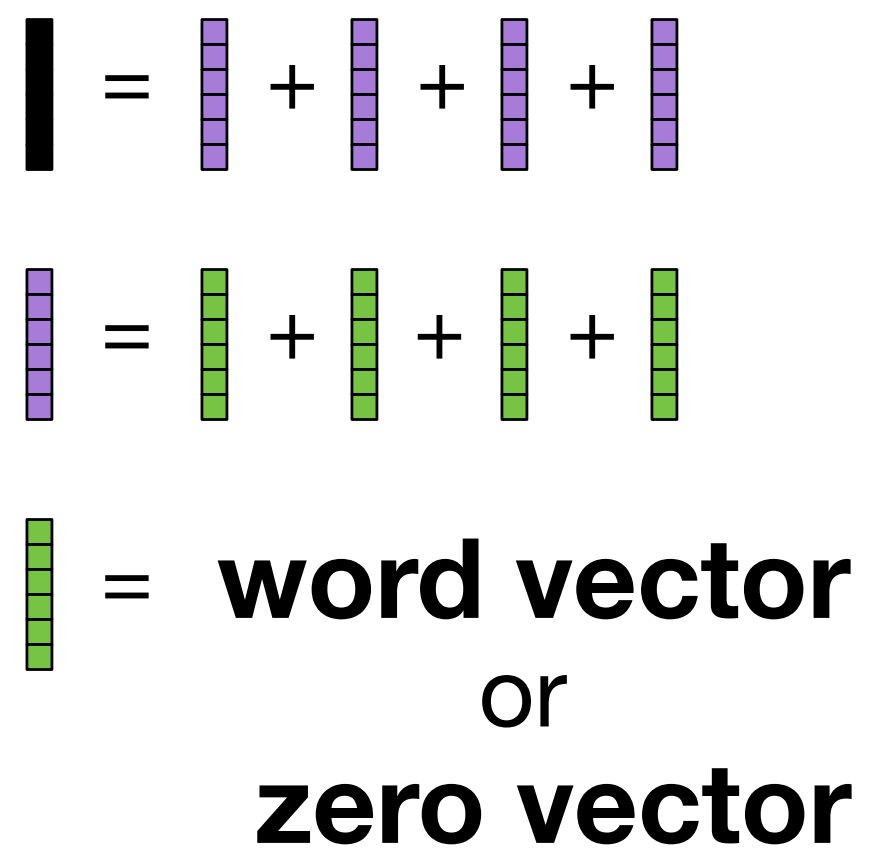
62 Multi-scale sentence matching



(a) Siamese Architecture for Sentence Matching

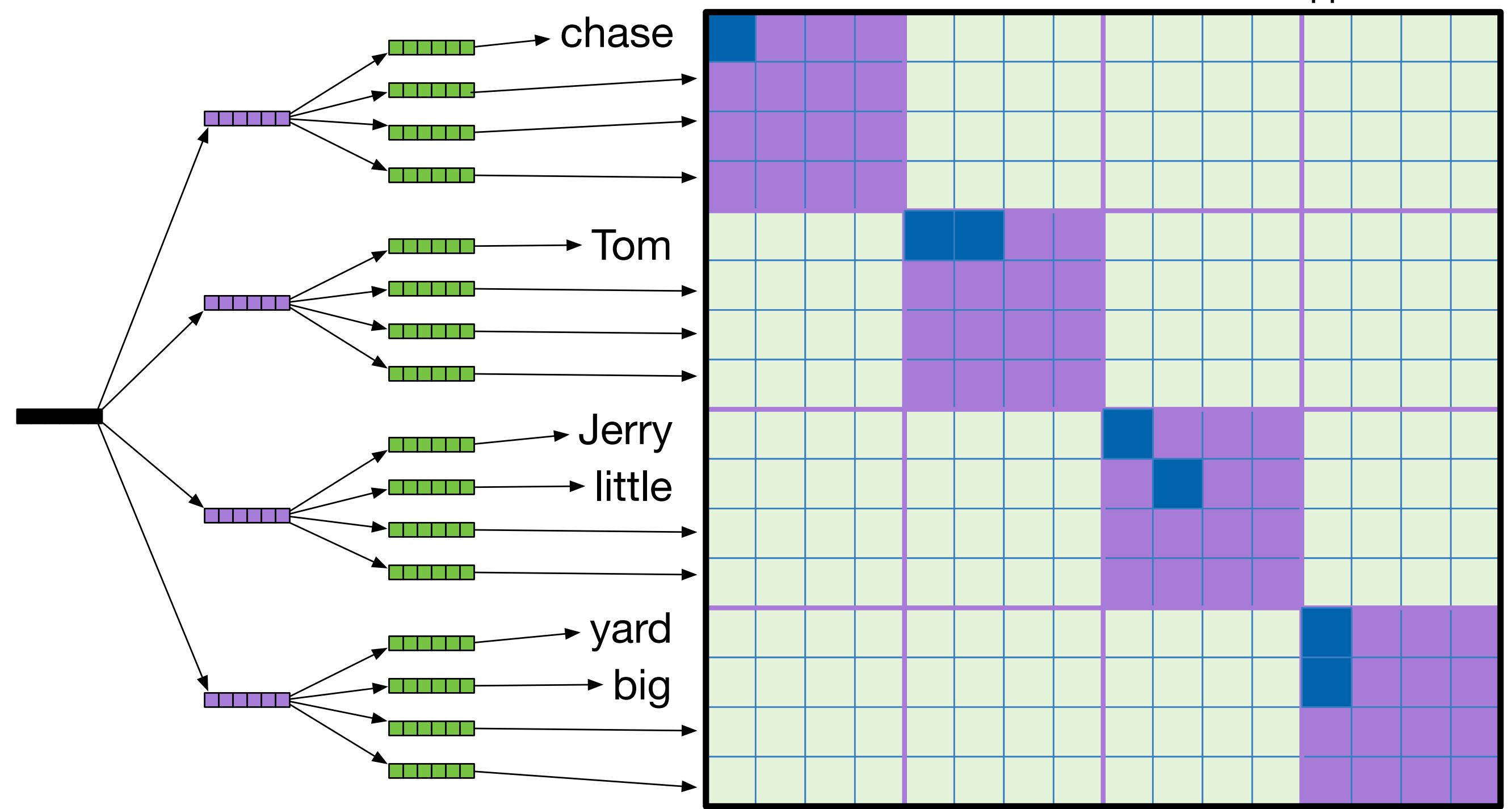


(b) Siamese Architecture with Factorized Multi-scale Sentence Representation

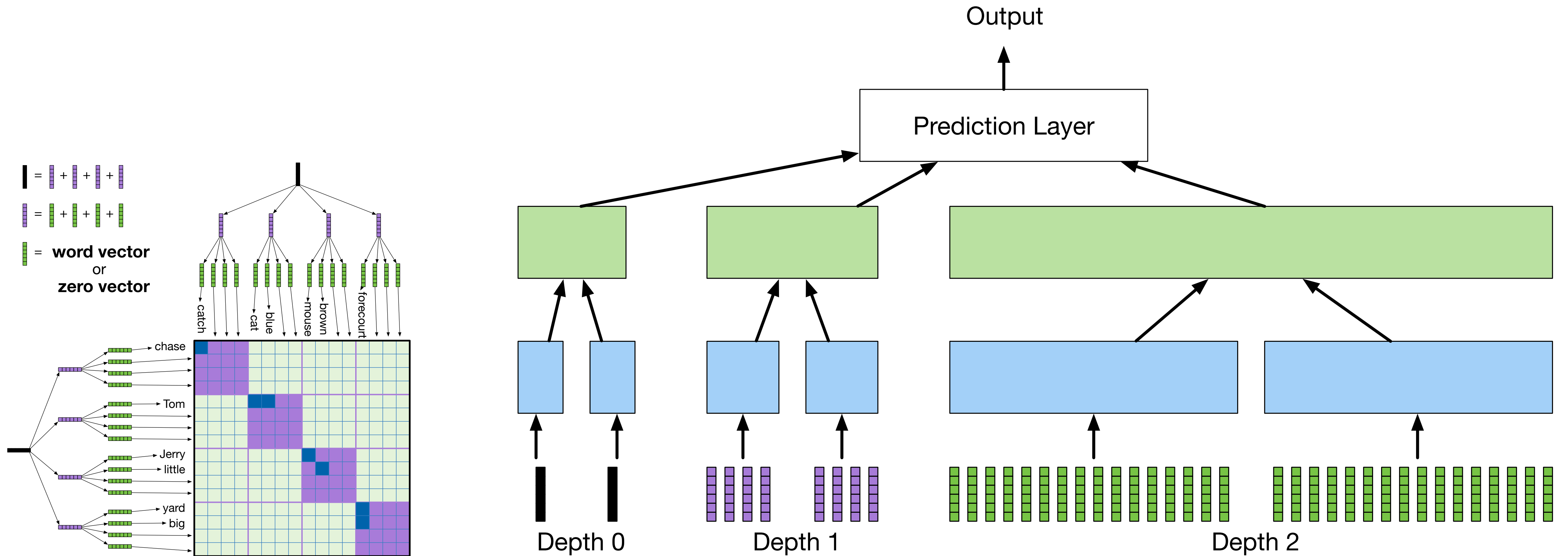


The **blue cat** is **catching** the **brown mouse** in the **forecourt**.

The **little Jerry** is **being chased** by **Tom** in the **big yard**.



64 Multi-scale sentence matching



65 Comparison to existing methods

Model	MSRP		SICK		MSRvid		STSbenchmark	
	Acc.(%)	F1(%)	r	ρ	r	ρ	r	ρ
MaLSTM	66.95	73.95	0.7824	0.71843	0.7325	0.7193	0.5739	0.5558
Multi-scale MaLSTM	74.09	82.18	0.8168	0.74226	0.8236	0.8188	0.6839	0.6575
HCTI	73.80	80.85	0.8408	0.7698	0.8848	0.8763	0.7697	0.7549
Multi-scale HCTI	74.03	81.76	0.8437	0.7729	0.8763	0.8686	0.7269	0.7033

(Accuracy, F1, Pearson's r , Spearman's ρ)

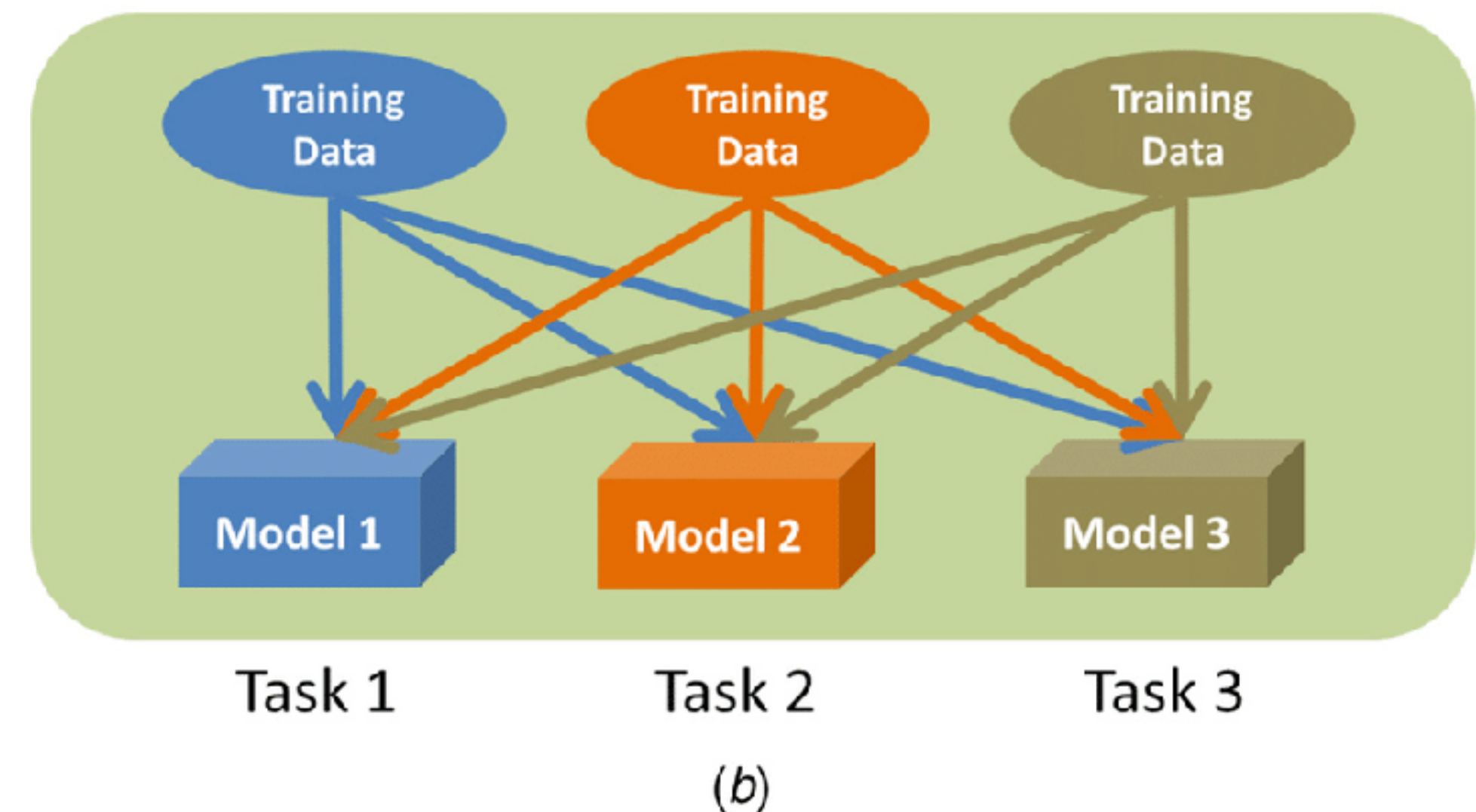
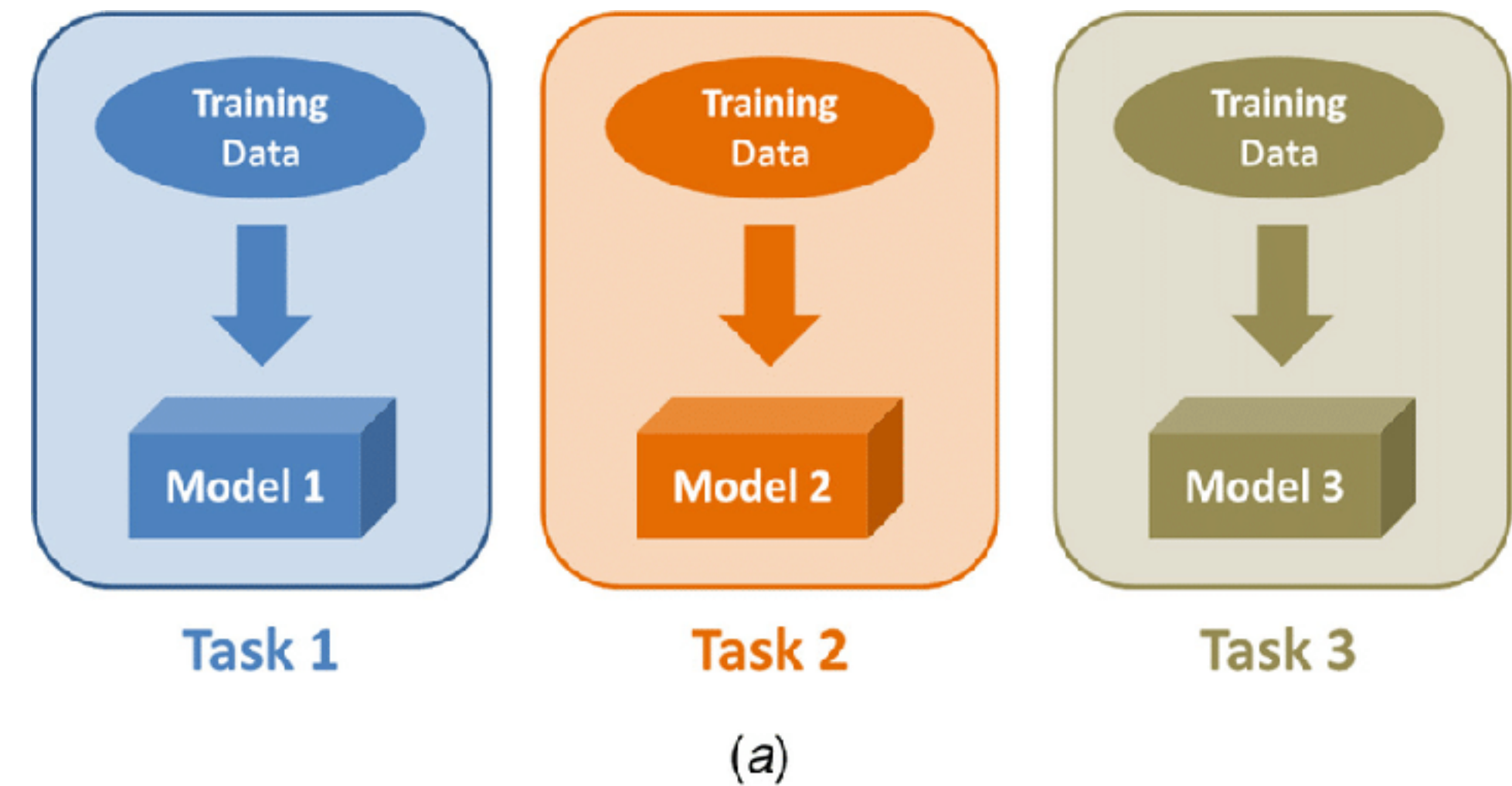
Open source: <https://github.com/BangLiu/SentenceMatching>

Multitask Learning for Sentence Embedding



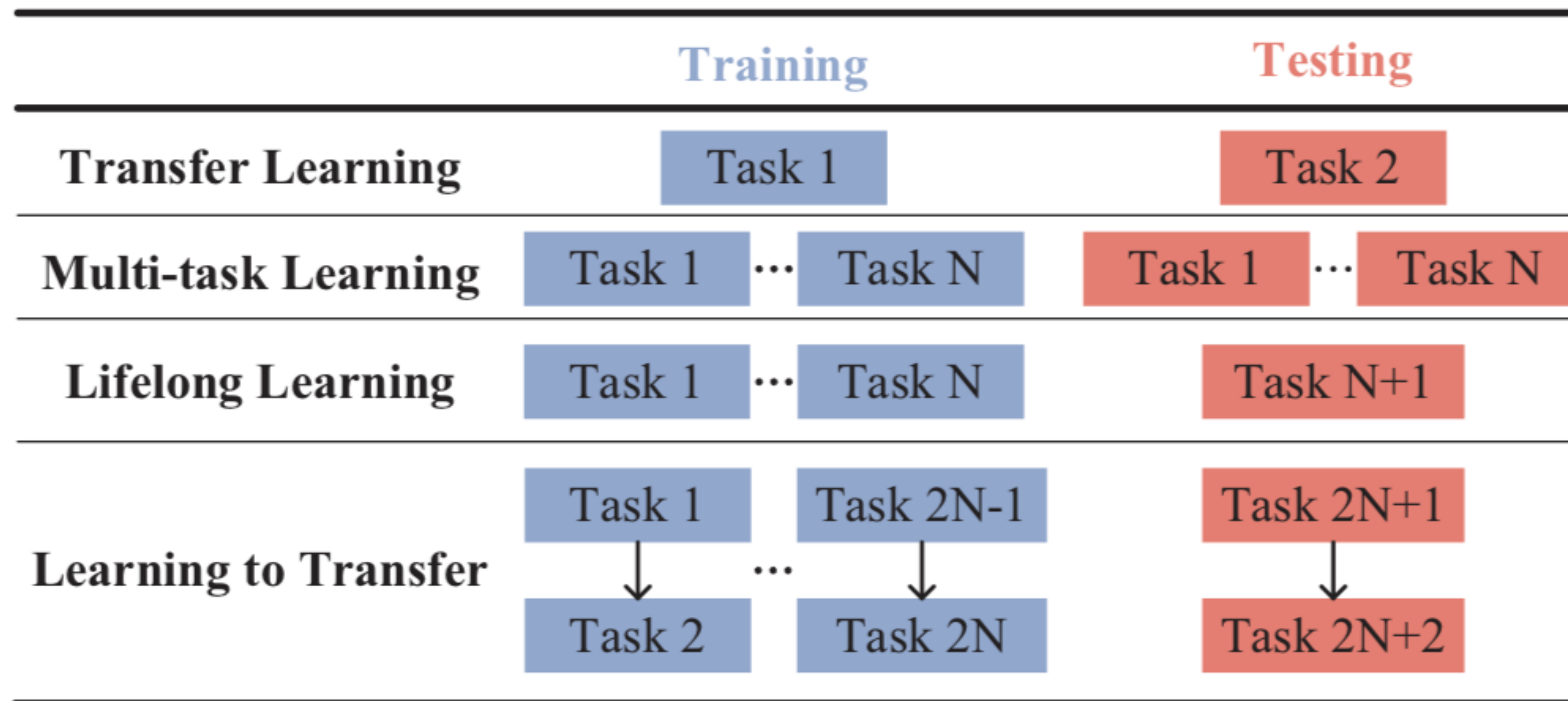
67 Motivation: limited training data

- Limited amounts of training data are available for many NLP tasks.
- Multitask to Increase Data: perform multitasking when one of your two tasks has many fewer data
- Multitask to Increase Data: perform multitasking when your tasks are related



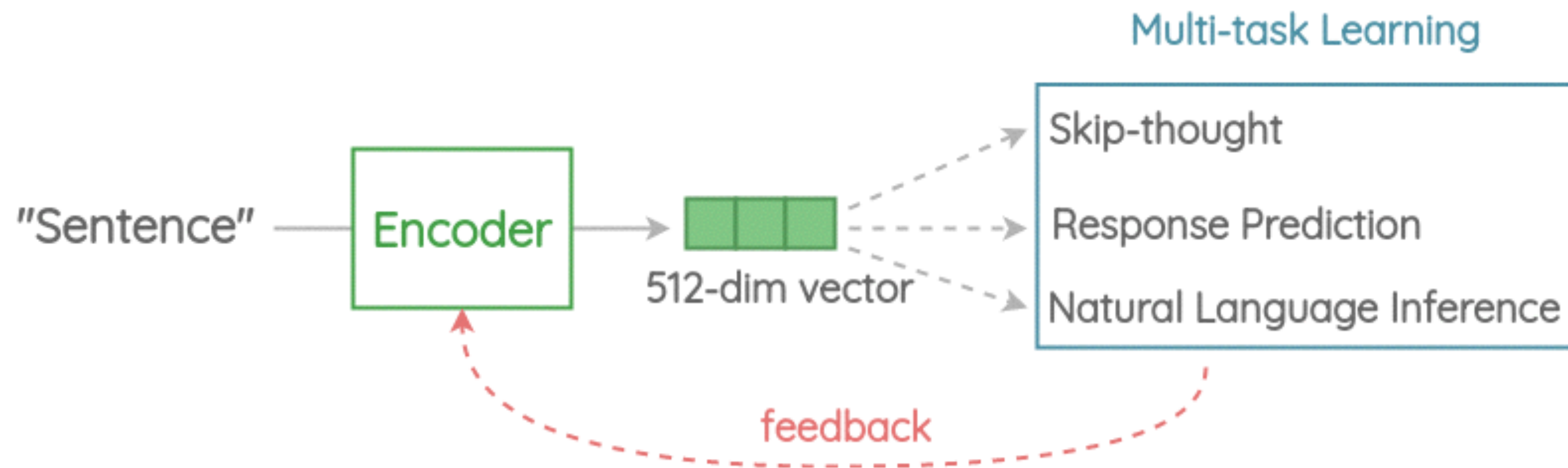
Types of learning

- Multi-task learning is a general term for training on multiple tasks
- Transfer learning is a type of multi-task learning where we only really care about one of the tasks



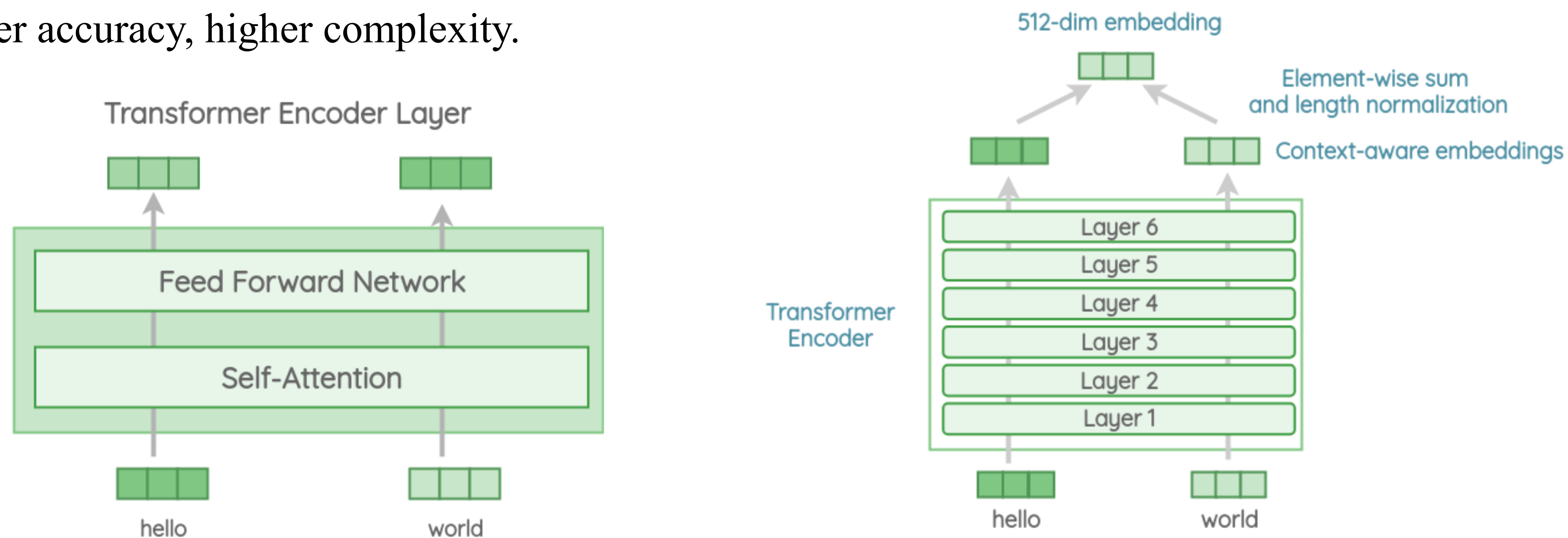
69 Universal Sentence Encoder (USE)

- Design an encoder that summarizes any given sentence to a 512-dimensional sentence embedding.
- Use this same embedding to solve multiple tasks and based on the mistakes it makes on those, we update the sentence embedding.



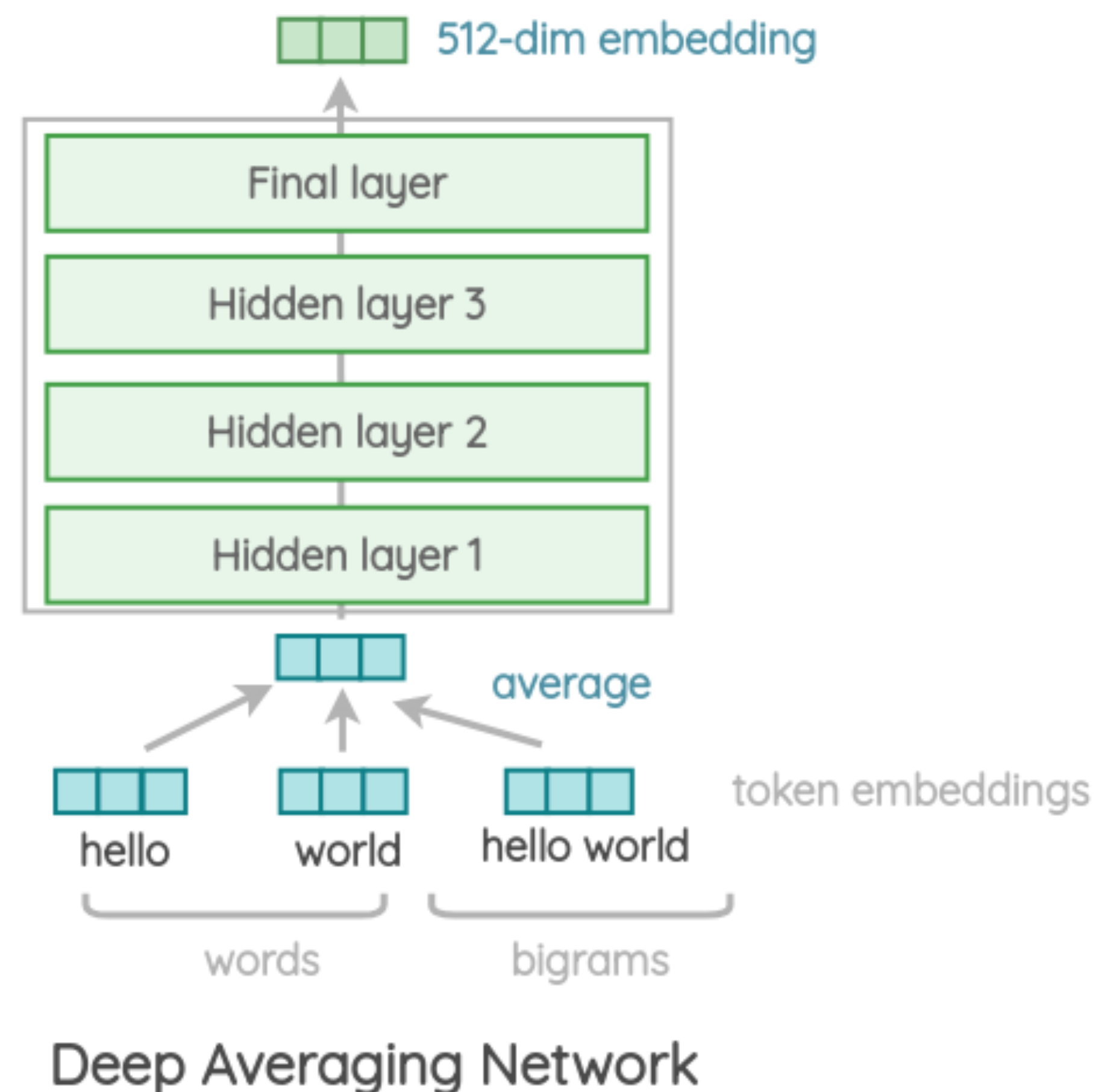
70 USE Encoder: Transformer

- 6 stacked transformer layers. Each layer has a self-attention module followed by a feed-forward network.
- The output context-aware word embeddings are added element-wise and divided by the square root of the length of the sentence to account for the sentence-length difference.
- Better accuracy, higher complexity.



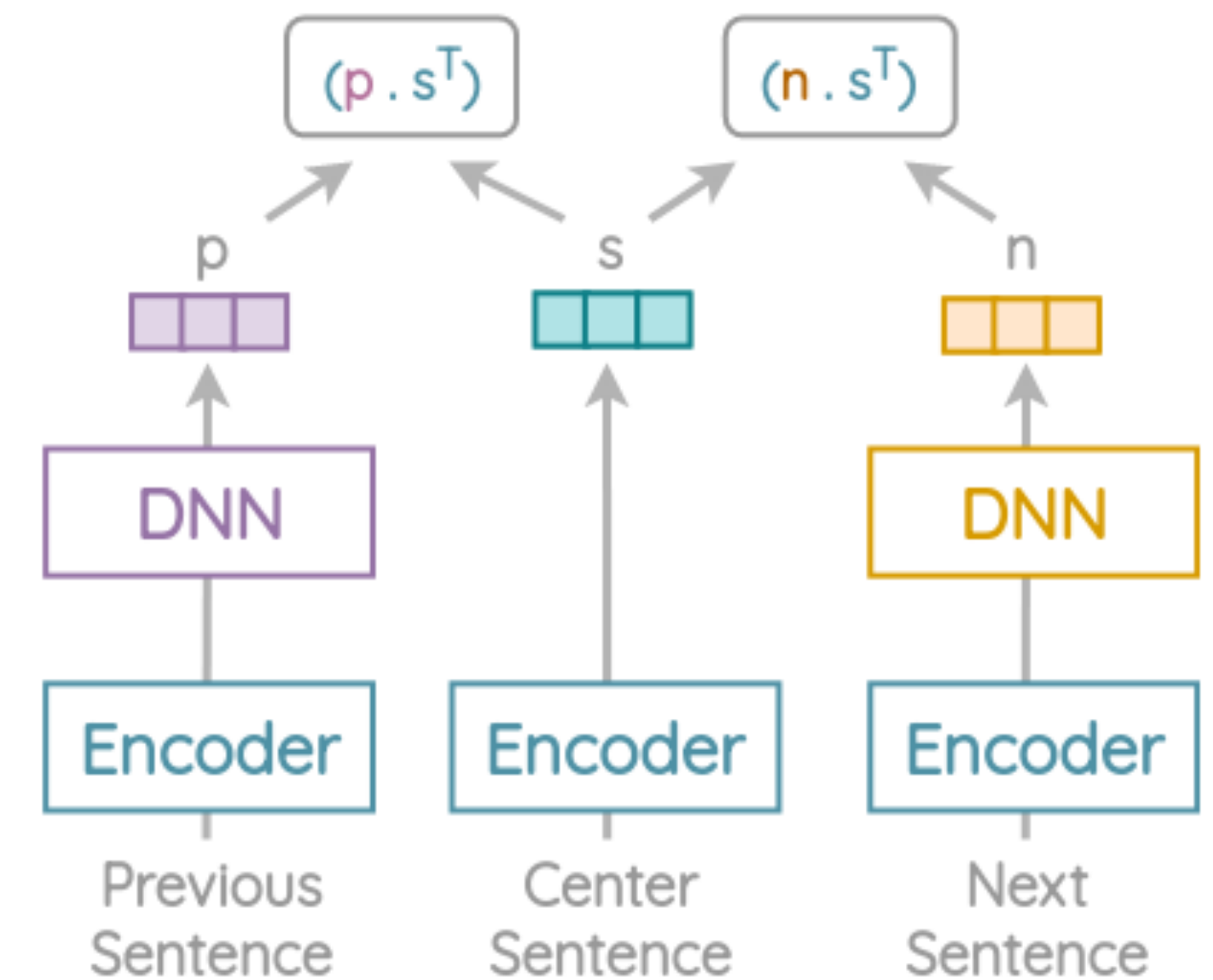
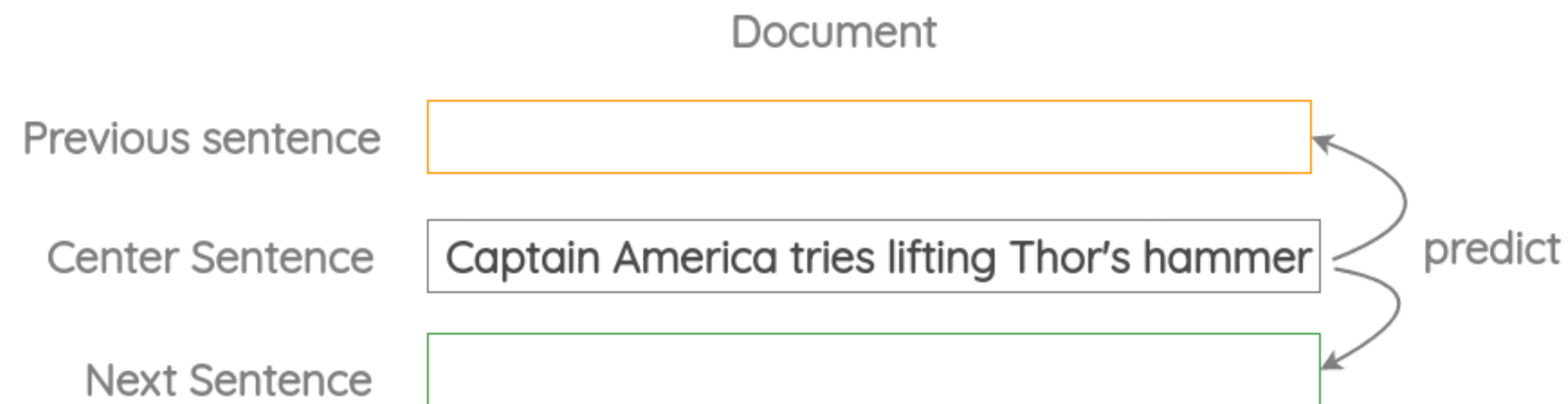
71 USE Encoder: Deep Average Network

- First, the embeddings for word and bi-grams present in a sentence are averaged together.
- Then, they are passed through 4-layer feed-forward deep DNN to get 512-dimensional sentence embedding.
- Slightly reduced accuracy, more efficient inference



Task: modified skip-thought

- The idea with original skip-thought paper from Kiros et al. was to use the current sentence to predict the previous and next sentence.
- In USE, the same core idea is used. But instead of LSTM encoder-decoder architecture, transformer or DAN is used.



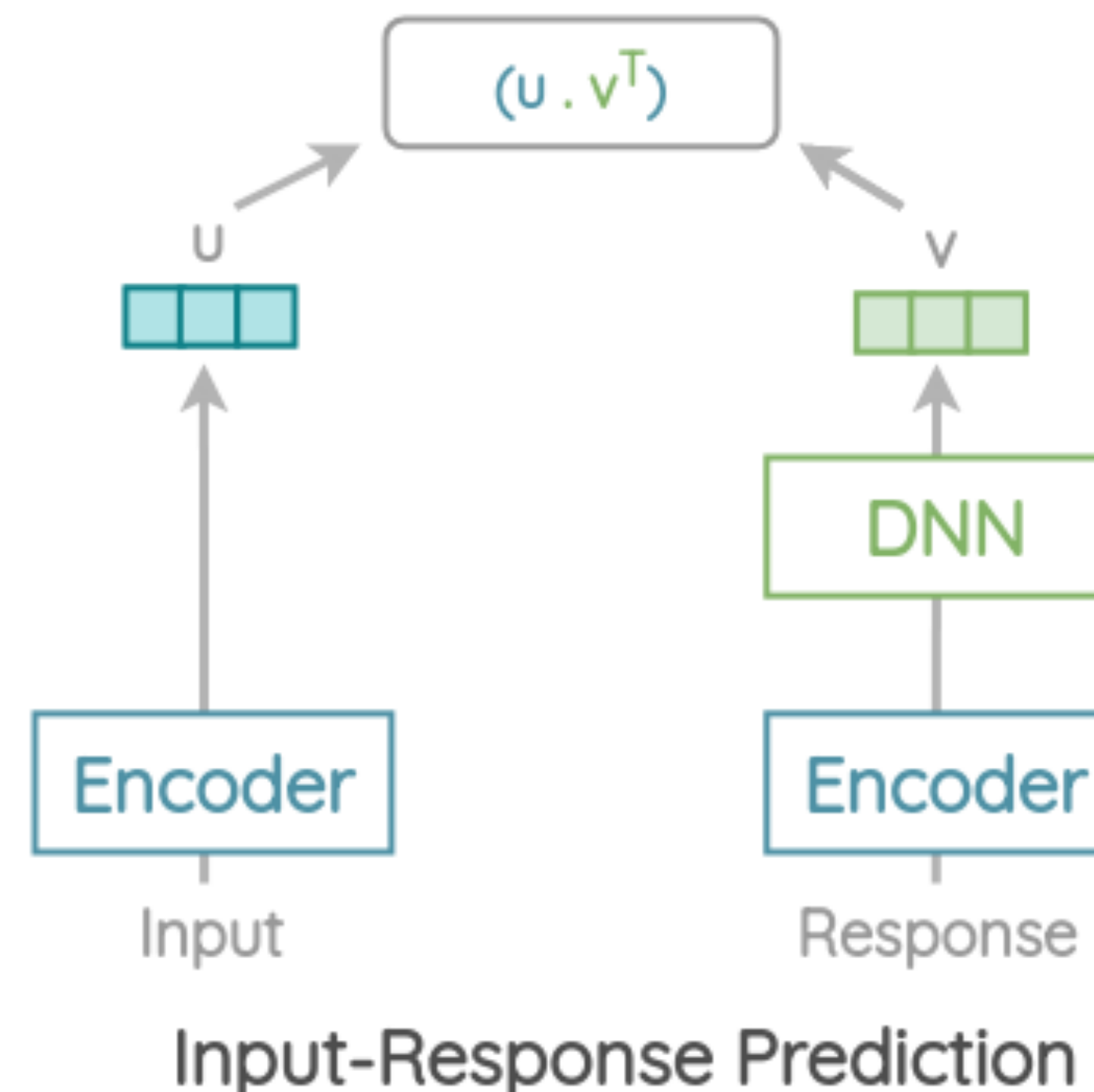
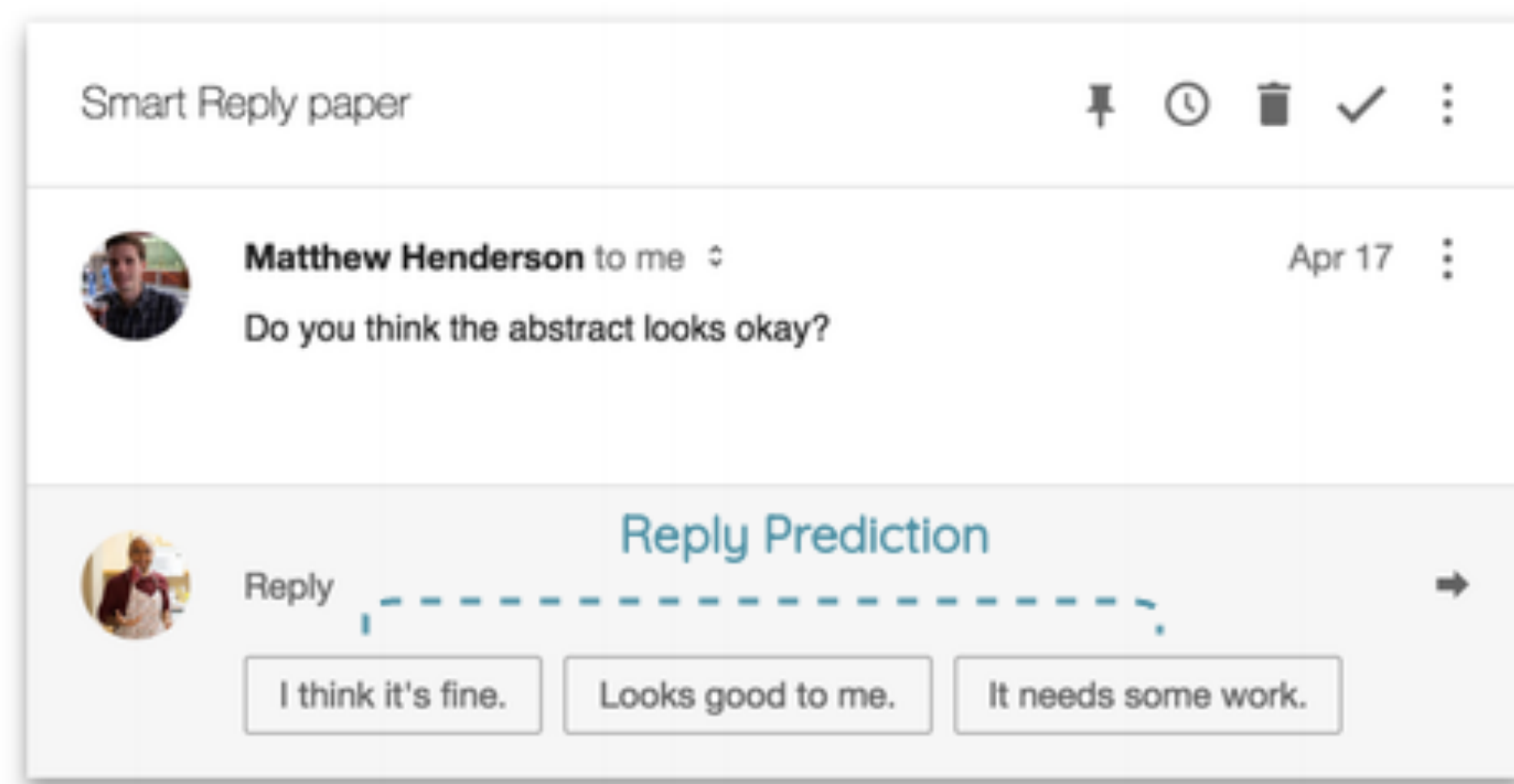
Skip-Thought Vectors: <https://arxiv.org/pdf/1506.06726.pdf>

Cer et al., “Universal Sentence Encoder”

<https://amitness.com/2020/06/universal-sentence-encoder/>

73 Task: conversational input-response prediction

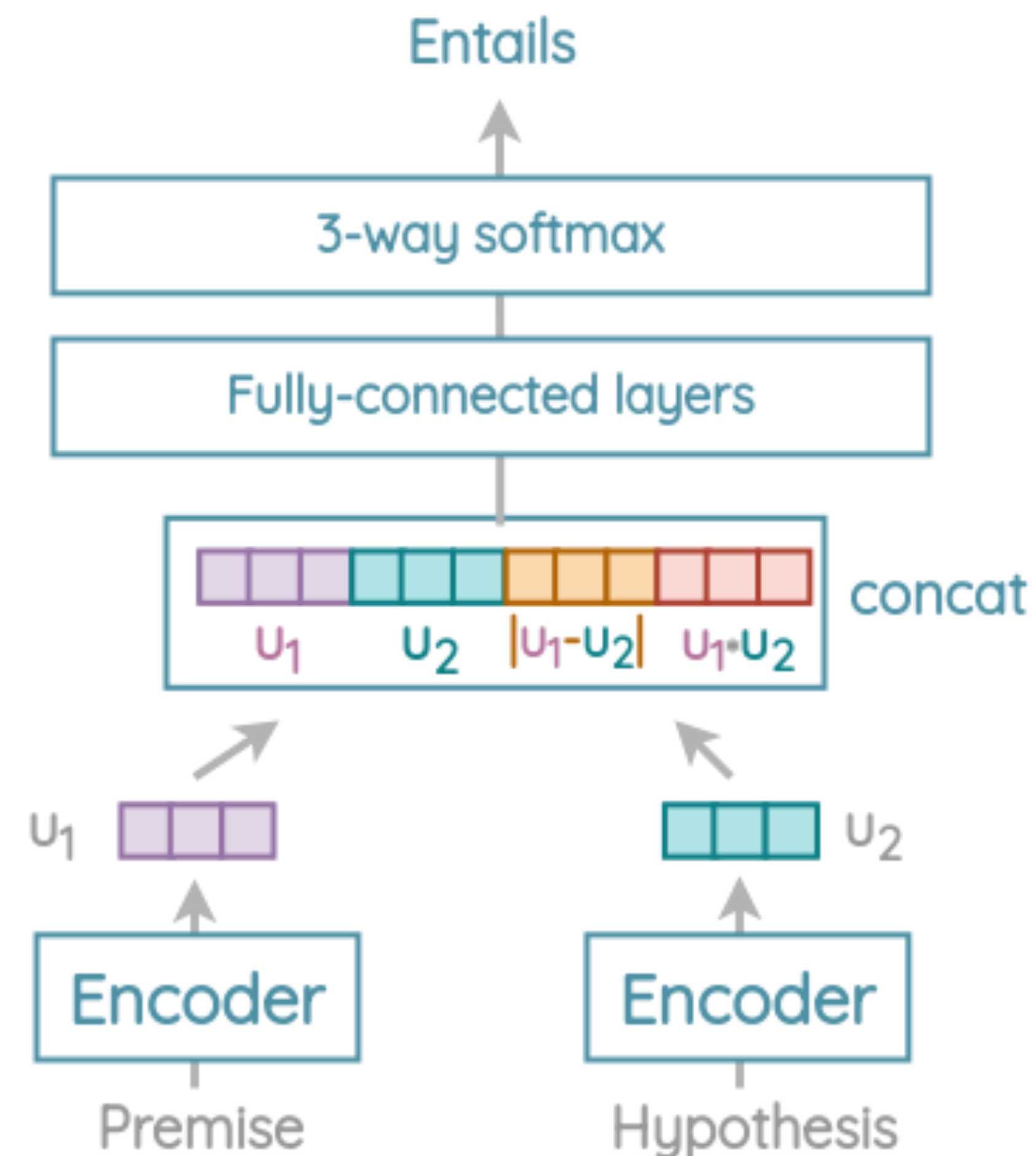
- Predict the correct response for a given input among a list of correct responses and other randomly sampled responses.
- The dot product of this two vectors (u for input and v for response) gives the relevance of an input to response.



74 Task: natural language inference

- Predict if a hypothesis entails, contradicts, or is neutral to a premise

Premise	Hypothesis	Judgement
A soccer game with multiple males playing	Some men are playing a sport	entailment
I love Marvel movies	I hate Marvel movies	contradiction
I love Marvel movies	A ship arrived	neutral



**Which method is
better?**

Which model?

- Not very extensive comparison...
- Wieting et al. (2015) find that simple word averaging is more robust out-of-domain
- Devlin et al. (2018) compare unidirectional and bi-directional transformer, but no comparison to LSTM like ELMo (for performance reasons?)
- Yang et al. (2019) have ablation where similar data to BERT is used and improvements are shown

77 Which training objective?

- Not very extensive comparison...
- Zhang and Bowman (2018) control for training data, and find that bi-directional LM seems better than MT encoder
- Devlin et al. (2018) find next-sentence prediction objective good compliment to LM objective

78 Which data?

- Not very extensive comparison...
- Zhang and Bowman (2018) find that more data is probably better, but results preliminary.
- Yang et al. (2019) show some improvements by adding much more data from web, but not 100% consistent.
- Data with context is probably essential.

79 Todo

- **Reading Assignment:** A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification: <https://www.aclweb.org/anthology/I17-1026.pdf>
(Due: Feb 4th, 2022 23:59pm, EST timezone)
- Suggested Readings:
 - Doc2Vec: https://cs.stanford.edu/~quocle/paragraph_vector.pdf
 - Universal Sentence Encoder: <https://arxiv.org/pdf/1803.11175.pdf>

Next lecture: Seq2Seq, Attention, Machine Translation

Thanks! Q&A

Bang Liu

Email: bang.liu@umontreal.ca

Homepage: <http://www-labs.iro.umontreal.ca/~liubang/>