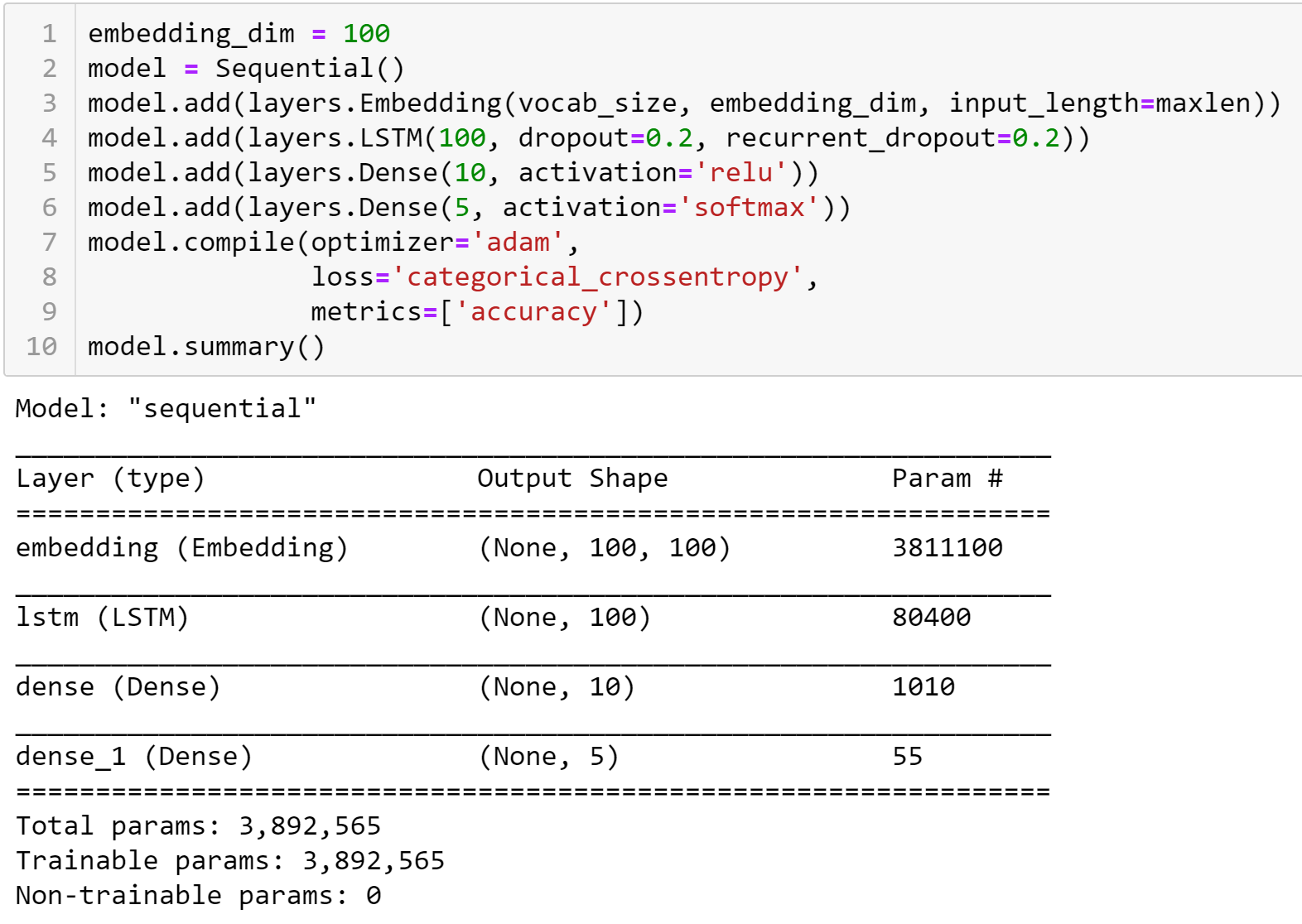
Deep Learning for Text 2

Applied Text Mining

Ayoub Bagheri

## RNN: Recap



## Lecture’s plan

1. Convolutional Neural Networks
2. Transformers
3. BERT

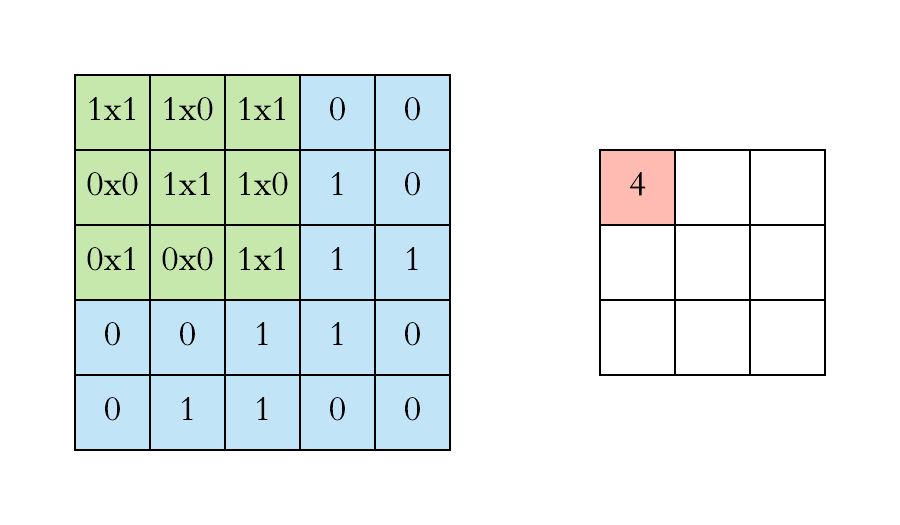
## Convolutional Neural Network (CNN)

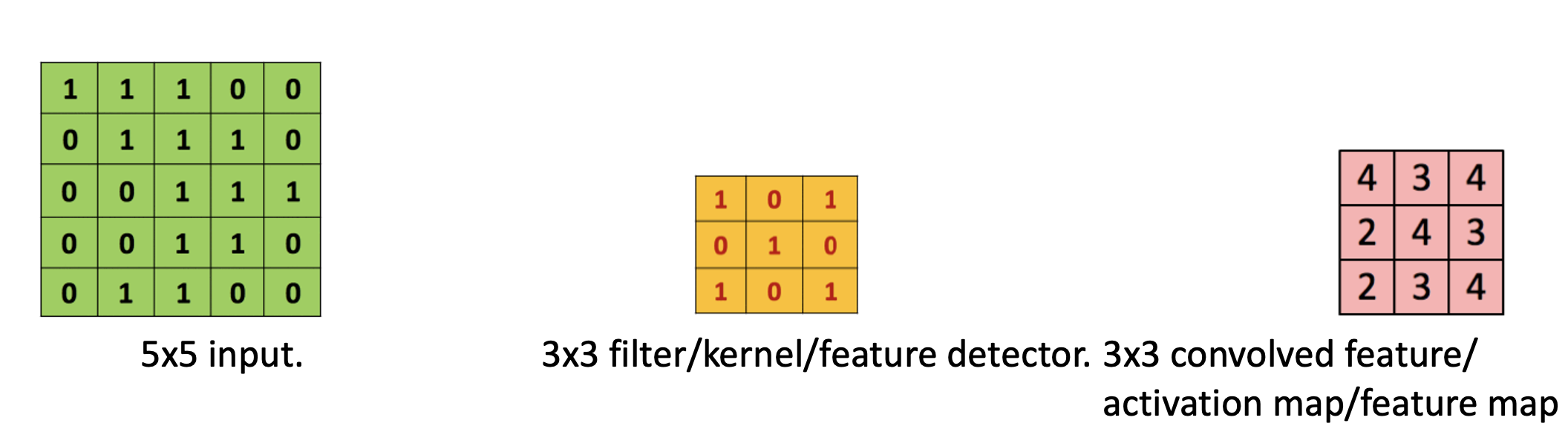
* Intuition: Neural network with specialized connectivity structure
  + Stacking multiple layers of feature extractors, low-level layers extract local features, and high-level layers extract learn global patterns.
* There are a few distinct types of layers:
  + **Convolution Layer**: detecting local features through filters (discrete convolution)
  + **Pooling Layer**: merging similar features

## Convolution layer

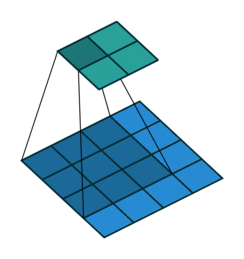
* The core layer of CNNs
* Convolutional layer consists of a set of filters
* Each filter covers a spatially small portion of the input data
* Each filter is convolved across the dimensions of the input data, producing a multidimensional **feature map**.
* As we convolve the filter, we are computing the dot product between the parameters of the filter and the input.
* **Deep Learning algorithm**: During training, the network corrects errors and filters are **learned**, e.g., in Keras, by adjusting weights based on **Stochastic Gradient Descent**, **SGD**.
* The key architectural characteristics of the convolutional layer is **local connectivity** and **shared weights**.

## Convolution without padding

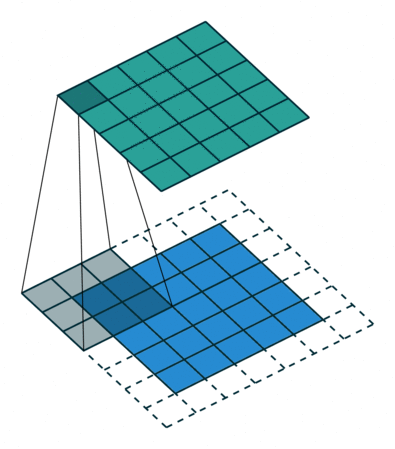




## Convolution with padding



4x4 input. 3x3 filter. Stride = 1. 2x2 output.

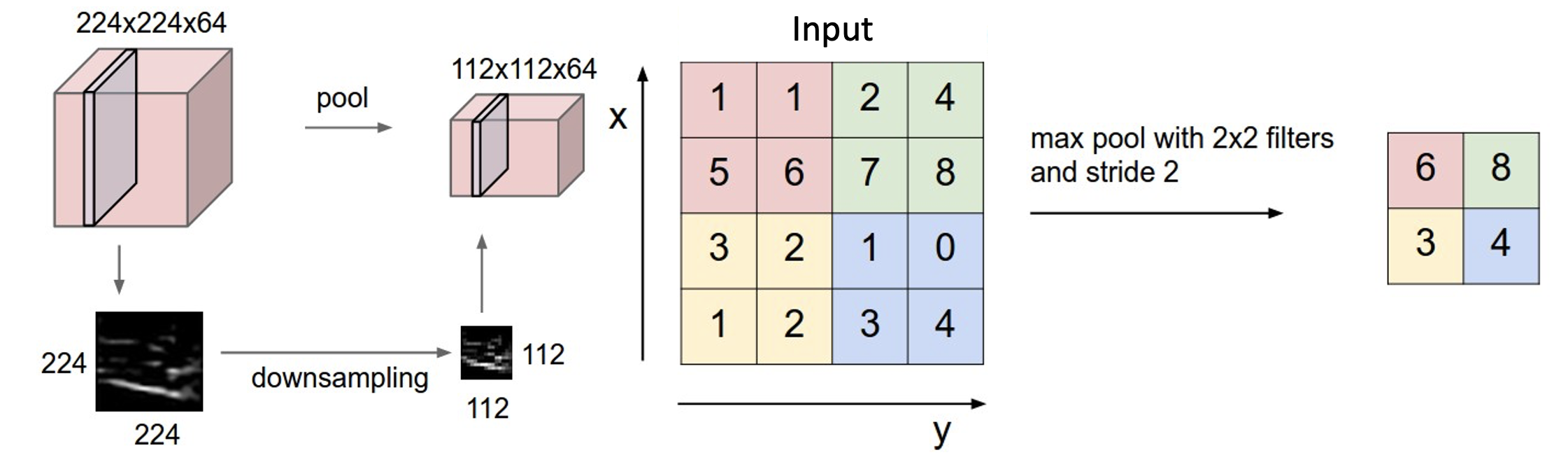


5x5 input. 3x3 filter. Stride = 1. 5x5 output.

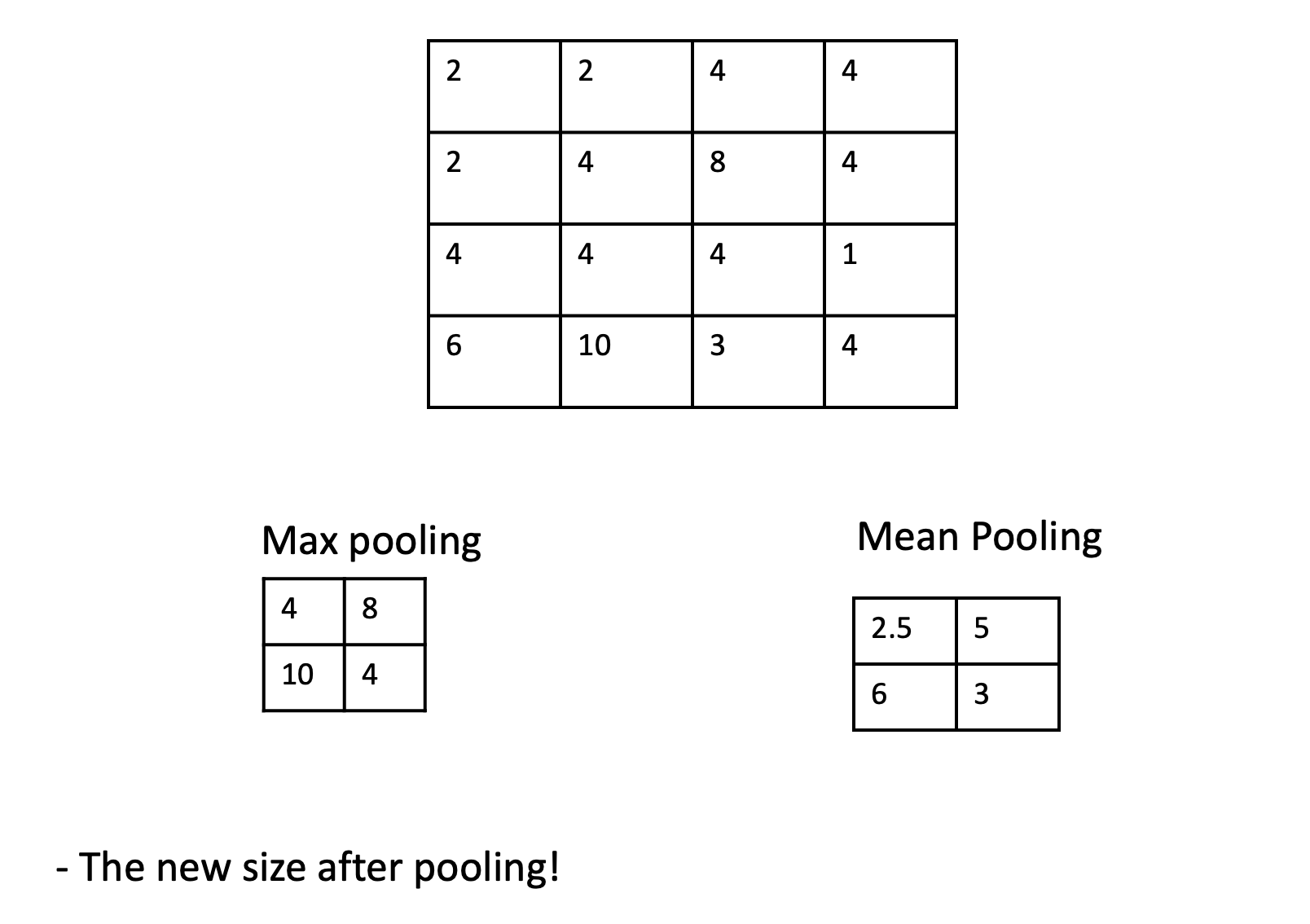
<https://github.com/vdumoulin/conv_arithmetic>

## Pooling layer

* Intuition: to progressively reduce the spatial size of the representation to reduce the amount of parameters and computation in the network, and hence to also control overfitting
* Pooling partitions the input image (or documents) into a set of non-overlapping rectangles (n-grams) and, for each such sub-region, outputs the maximum value of the features in that region.



## Pooling (down sampling)



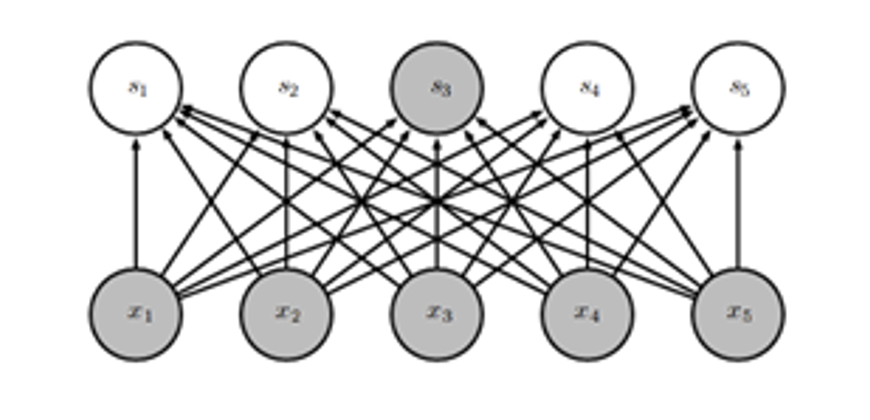
## Convolutional neural network

For processing data with a **grid-like** or array topology:

* 1-D convolution: time-series data, sensor signal data
* 2-D convolution: image data
* 3-D convolution: video data

## Other layers

* The convolution, and pooling layers are typically used as a set. Multiple sets of the above layers can appear in a CNN design.
* After a few sets, the output is typically sent to one or two **fully connected layers**.
  + A fully connected layer is a ordinary neural network layer as in other neural networks.
  + Typical activation function is the sigmoid function.
  + Output is typically class (classification) or real number (regression).



## Other layers

* The final layer of a CNN is determined by the research task.
* Classification: Softmax Layer
  + The outputs are the probabilities of belonging to each class.
* Regression: Linear Layer
  + The output is a real number.

What hyperparameters do we have in a CNN model?

# CNN for Text

## CNN

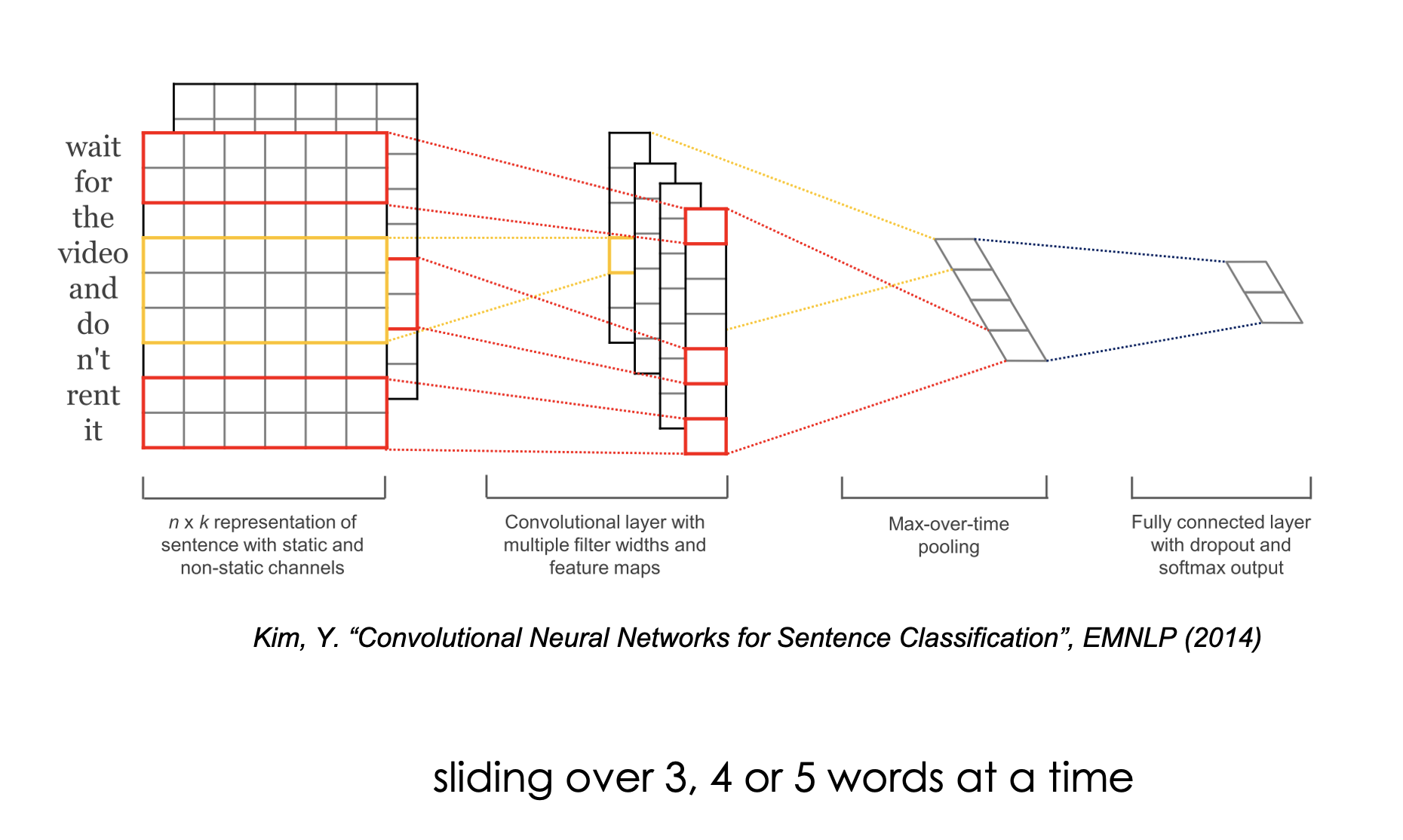
Main CNN idea for text:

Compute vectors for n-grams and group them afterwards

Example: “this takes too long” compute vectors for:

This takes, takes too, too long, this takes too, takes too long, this takes too long

## CNNs for sentence classification



<https://arxiv.org/pdf/1408.5882.pdf>

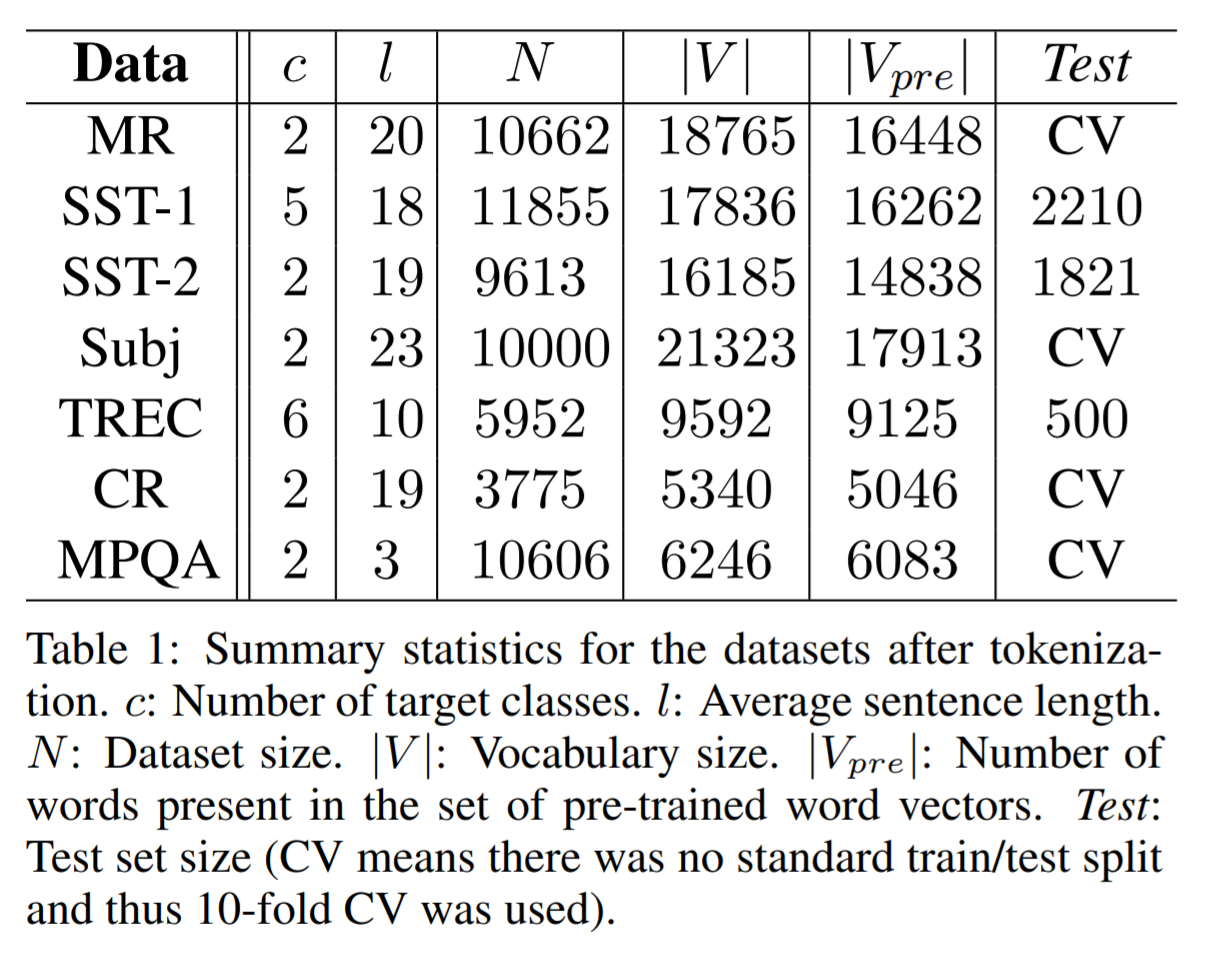
## Data sets (1)

* **MR**: Movie reviews with one sentence per review. Classification involves detecting positive/negative reviews (Pang and Lee, 2005). url: <https://www.cs.cornell.edu/people/pabo/movie-review-data/>
* **SST-1**: Stanford Sentiment Treebank—an extension of MR but with train/dev/test splits provided and fine-grained labels (very positive, positive, neutral, negative, very negative), re-labeled by Socher et al. (2013). url: <https://nlp.stanford.edu/sentiment/>
* **SST-2**: Same as SST-1 but with neutral reviews removed and binary labels.
* **Subj**: Subjectivity dataset where the task is to classify a sentence as being subjective or objective (Pang and Lee, 2004).

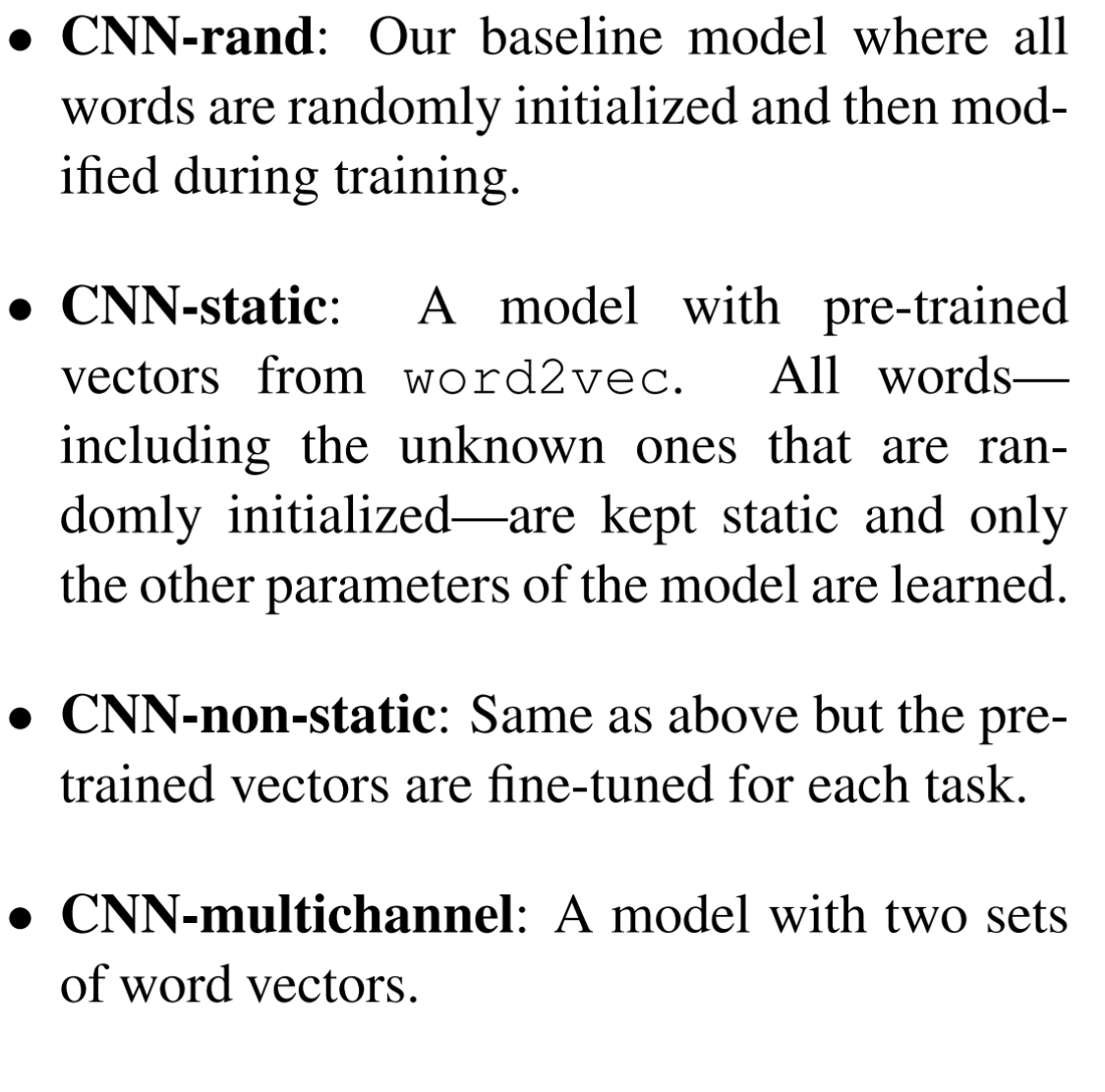
## Data sets (2)

* **TREC**: TREC question dataset—task involves classifying a question into 6 question types (whether the question is about person, location, numeric information, etc.) (Li and Roth, 2002). url: <https://cogcomp.seas.upenn.edu/Data/QA/QC/>
* **CR**: Customer reviews of various products (cameras, MP3s etc.). Task is to predict positive/negative reviews (Hu and Liu, 2004). url: <https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html>
* **MPQA**: Opinion polarity detection subtask of the MPQA dataset (Wiebe et al., 2005). url: <https://mpqa.cs.pitt.edu/corpora/mpqa_corpus/>

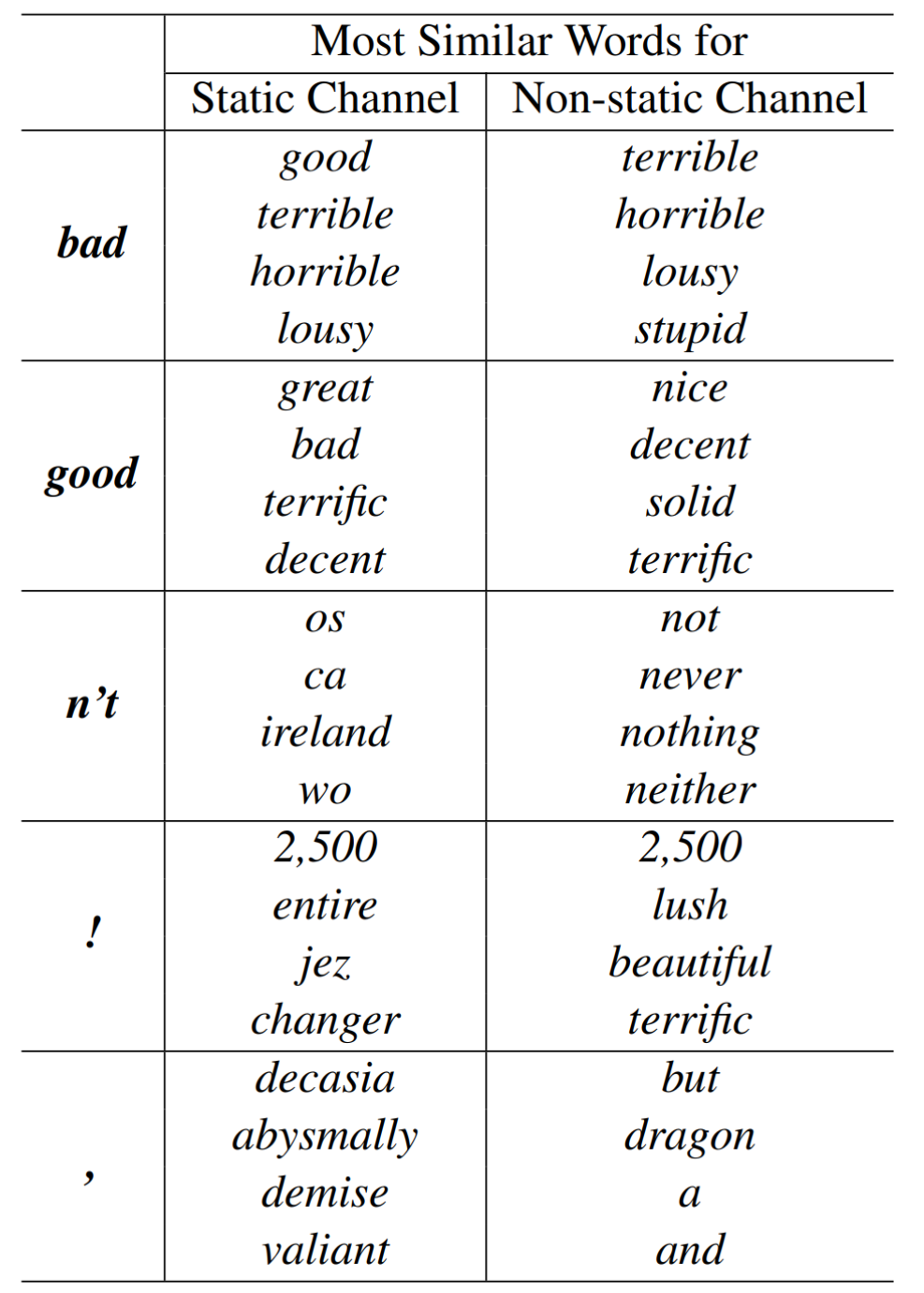
## Data sets statistics



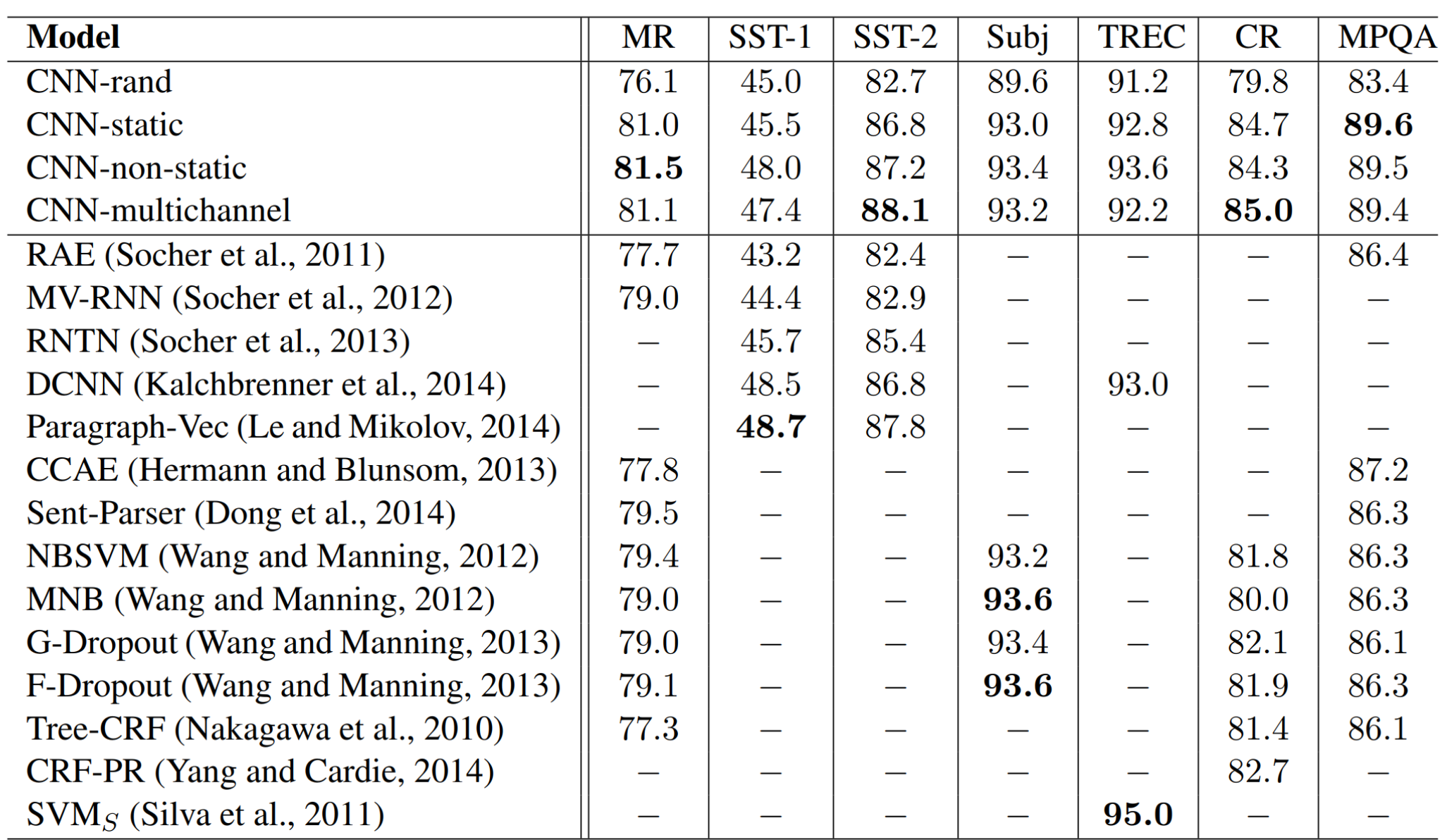
## CNN variations



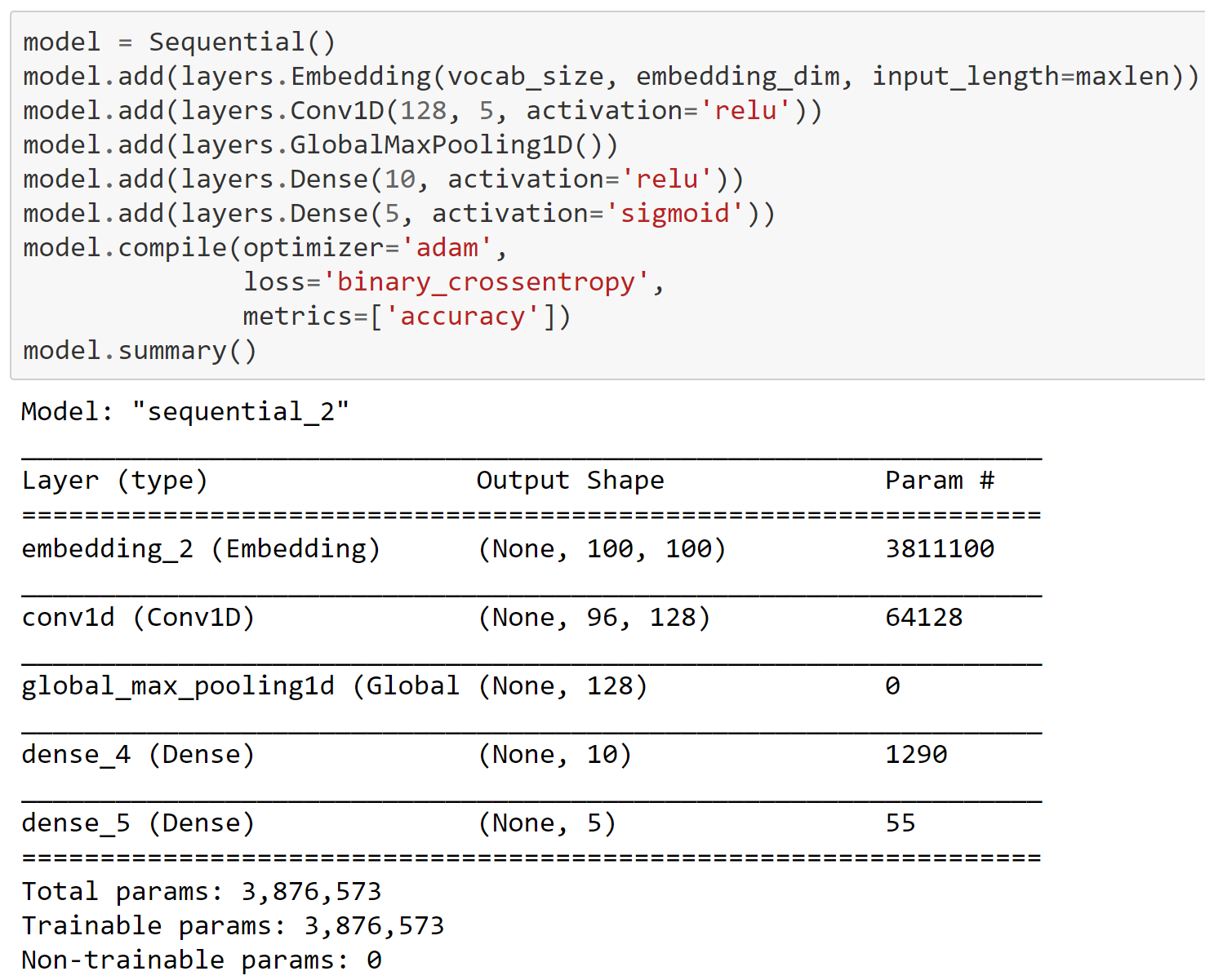
## Similar words



## Results

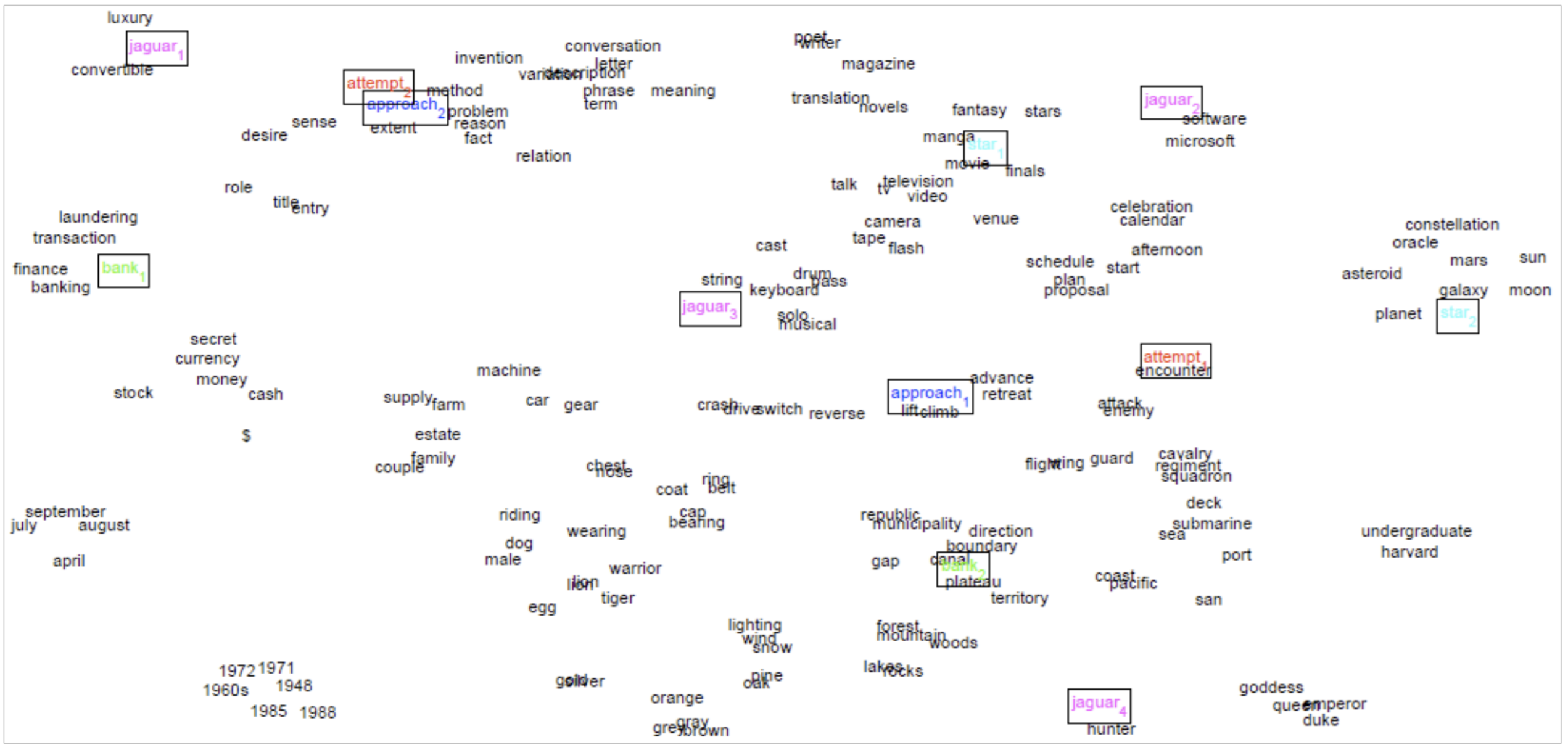


## CNN in Keras



# Contextual Word Embeddings & Transformers

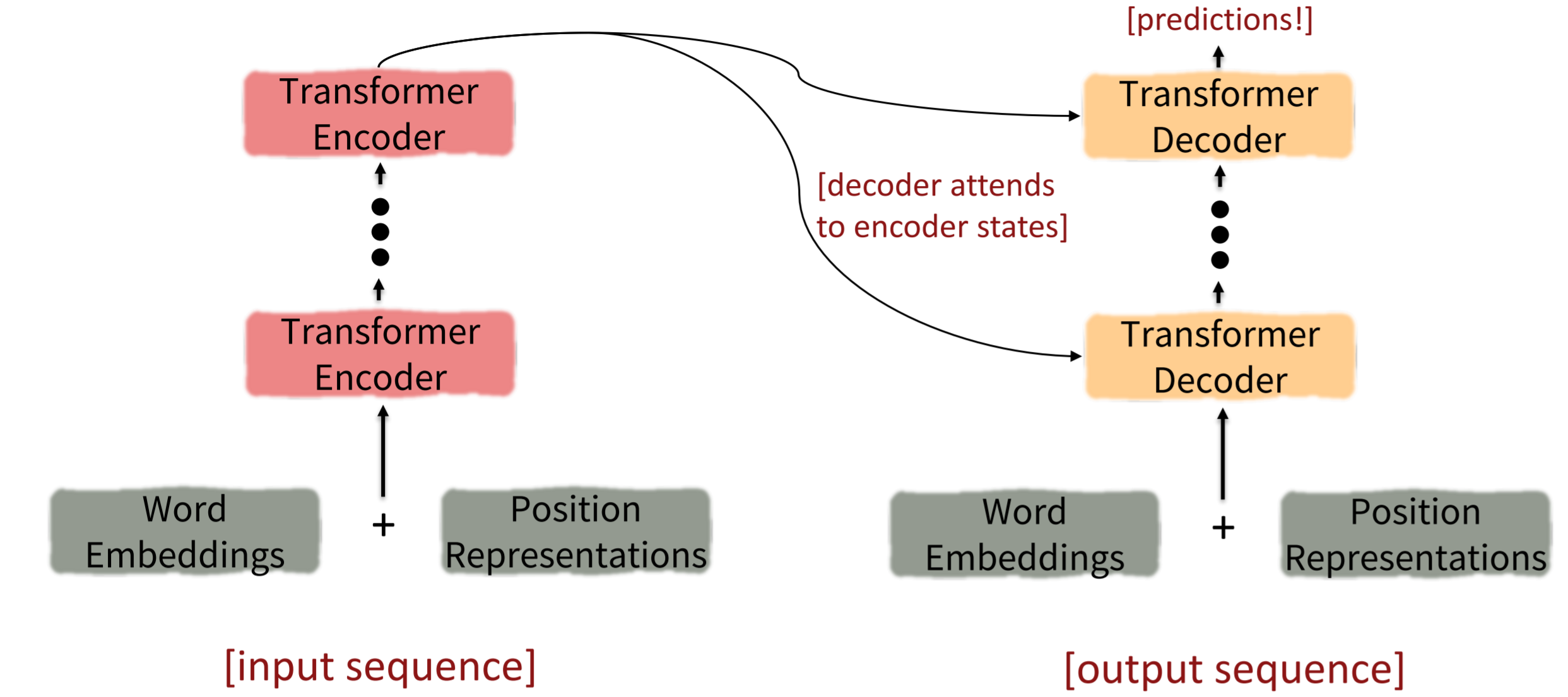
## Contextual Word Embeddings



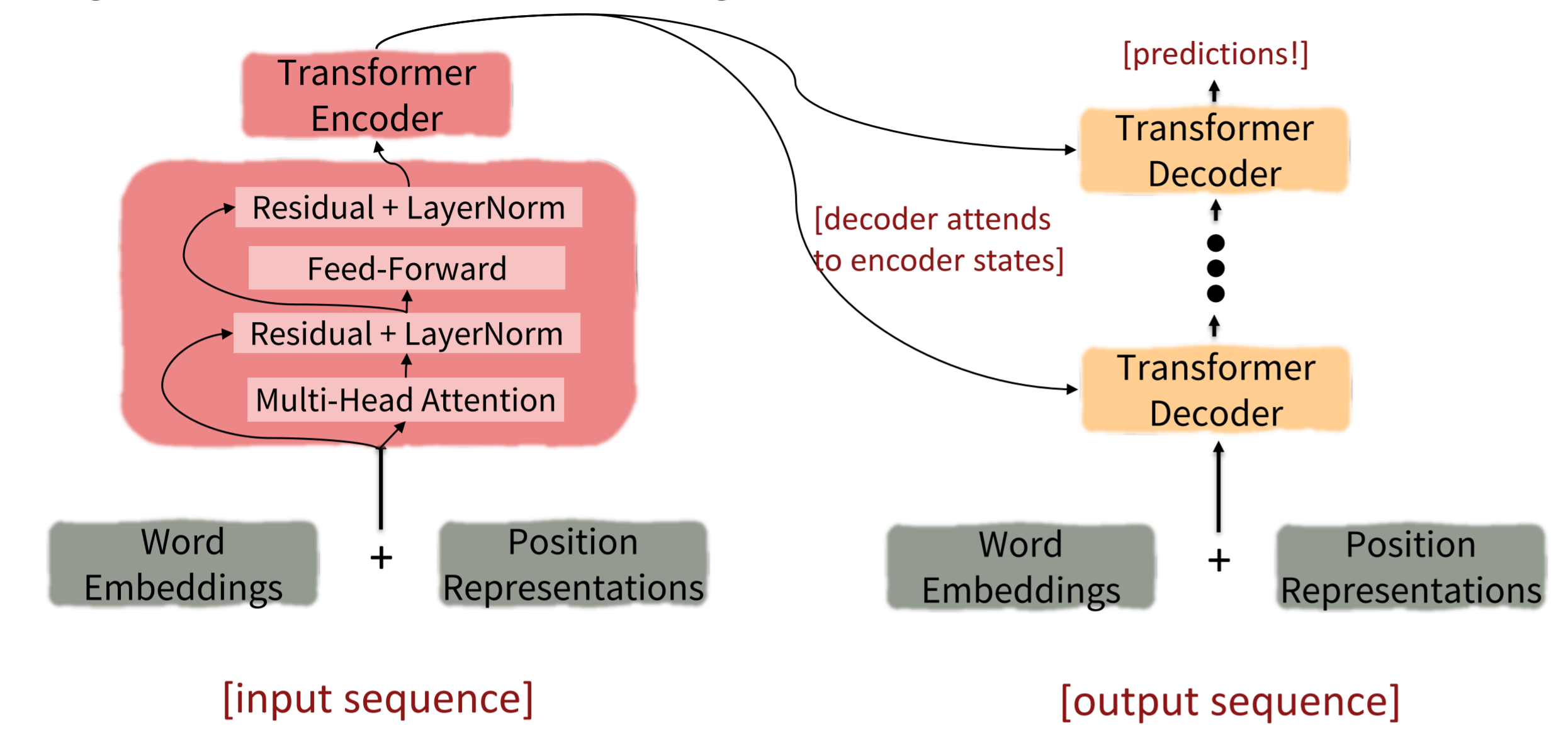
## Transformers

* A transformer adopts an encoder-decoder architecture.
* Transformers were developed to solve the problem of sequence transduction, or neural machine translation. That means any task that transforms an input sequence to an output sequence.
* More details on the architecture and implementation:
  + <https://arxiv.org/abs/1810.04805>
  + <http://nlp.seas.harvard.edu/2018/04/03/attention.html>
  + <https://jalammar.github.io/illustrated-transformer/>

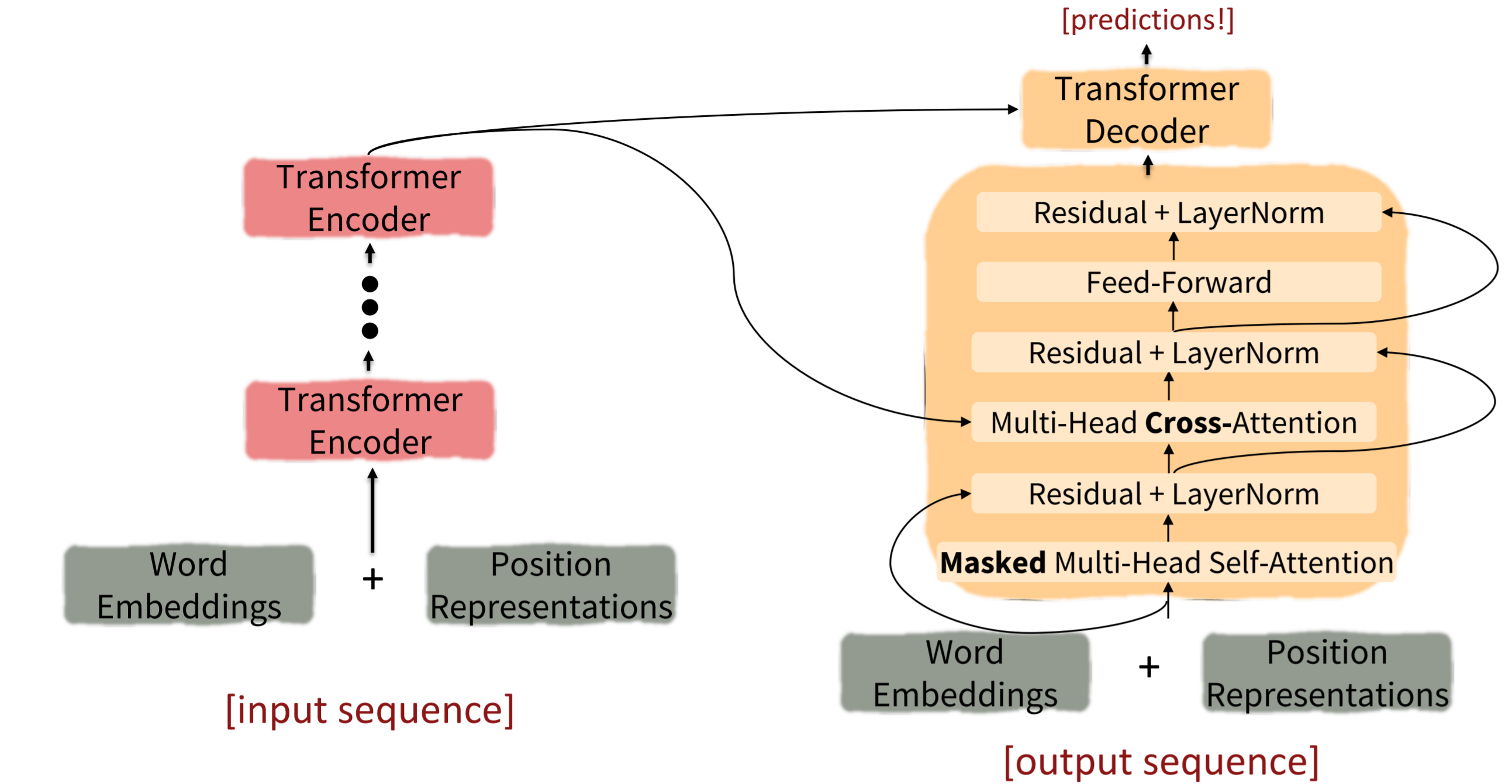
## The Transformer Encoder-Decoder



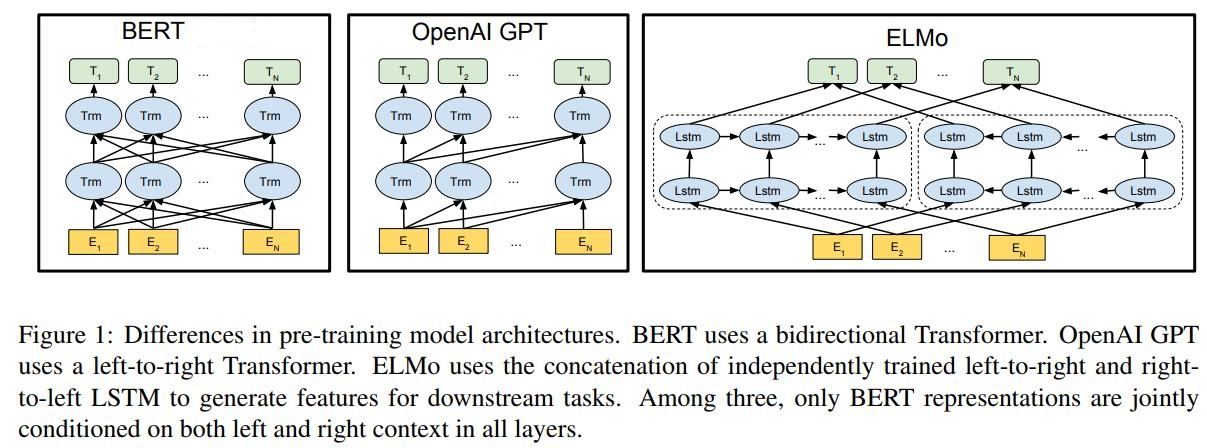
## The Transformer Encoder-Decoder



## The Transformer Encoder-Decoder



## Transformers



## BERT: Bidirectional Encoder Representations from Tranformers

## BERT: Bidirectional Encoder Representations from Tranformers

## What kinds of things does pretraining learn?

There’s increasing evidence that pretrained models learn a wide variety of things about the statistical properties of language:

* **Trivia:** Utrecht University is located in …

## Transformers

Graphical user interface, text, application, email

Description automatically generated

## Transformers

Graphical user interface, text, application, email

Description automatically generated

## Transformers

Graphical user interface, text, application, email

Description automatically generated

## What kinds of things does pretraining learn?

There’s increasing evidence that pretrained models learn a wide variety of things about the statistical properties of language:

* **Basic arithmetic:** I was thinking about the sequence that goes 1, 1, 2, 3, 5, 8, 13, 21, …

## Transformers

Graphical user interface, text, application

Description automatically generated

## What kinds of things does pretraining learn?

There’s increasing evidence that pretrained models learn a wide variety of things about the statistical properties of language:

* **Reasoning:** Garry went into the kitchen to make some tea. Standing next to Garry, Carrie pondered her destiny. Carrie left the …

## Transformers

Graphical user interface, text, application, email

Description automatically generated

## Transformers

Graphical user interface, text, application, email

Description automatically generated

## Transformers

Graphical user interface, text, application, email

Description automatically generated

## Transformers

* Write with Transformer: <https://transformer.huggingface.co/>
* Talk to Transformer: <https://app.inferkit.com/demo>
* Transformer model for language understanding: <https://www.tensorflow.org/text/tutorials/transformer>
* Pretrained models: <https://huggingface.co/transformers/pretrained_models.html>

# Summary

## Summary

* Convolutional Neural Networks
* Transformers
  + “Small” models like BERT have become general tools in a wide range of settings
  + GPT-3 has 175 billion parameters
* These models are still not well-understood

# Time for Practical 7!