Natural Language Processing summary

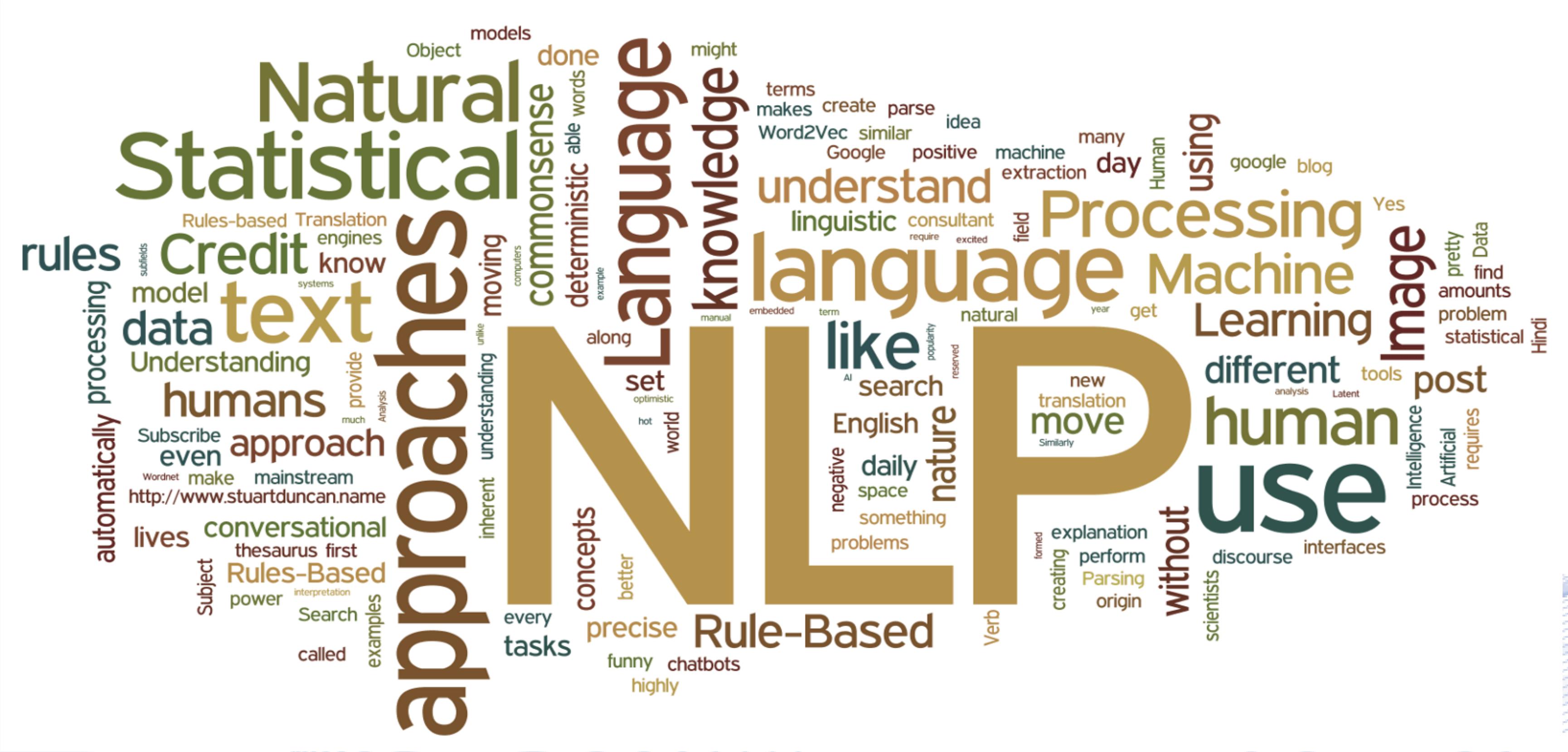
D. Karamouzas, Prof. Ioannis Pitas
Aristotle University of Thessaloniki
pitas@csd.auth.gr
www.aiia.csd.auth.gr
Version 1.2.1

Artificial Intelligence & Information Analysis Lab

Contents

- What is Natural Language Processing
- History
  - Symbolic NLP
  - Statistical NLP
  - Neural NLP
- Methods: Rules, Statistics, Neural networks
- Word Representations
  - Fixed (sparse)
    - One-hot encoding
    - Bag-of-words, TF-IDF
  - Distributed (dense)
    - Classic embeddings
    - Contextualized embeddings
- Common NLP Tasks

What is NLP ?


NLP
approaches
Language
language
Statistical
Natural
Processing
Machine
Learning
knowledge
text
Credit
data
human
use
understand
humans
approach
rules
Rule-Based
Rules-Based
Rules-based
Translation
Image
like
different
post
commonsense
deterministic
conversational
Understanding
processing
automatically
model
know
engines
linguistic
consultant
search
English
nature
daily
space
something
problems
move
translation
new
explanation
perform
Parsing
origin
without
discourse
interfaces
Intelligence
Artificial
requires
process
statistical
problem
amounts
find
pretty
Data
Hindi
Yes
tools
natural
get
using
Human
day
many
machine
extraction
google
blog
terms
makes
create
parse
idea
Word2Vec
similar
Google
positive
models
Object
done
might
words
able
moving
understanding
inherent
concepts
better
set
along
world
precise
tasks
every
funny
chatbots
highly
called
examples
Search
power
Subject
lives
thesaurus
first
http://www.stuartduncan.name
mainstream
make
even
Subscribe
provide
negative
creating
scientists
Verb

Short definition

The automatic manipulation of natural language, like speech and text, by software

or,

Automatic methods that take natural language as input or produce natural language as output

Has been around for more than 50 years and grew out of the field of linguistics with the rise of computers

Natural Language

The way we, humans, communicate with each other

Speech and text

Given the importance of this type of data, we must have methods to understand and reason about natural language

Challenge

Natural Language is :

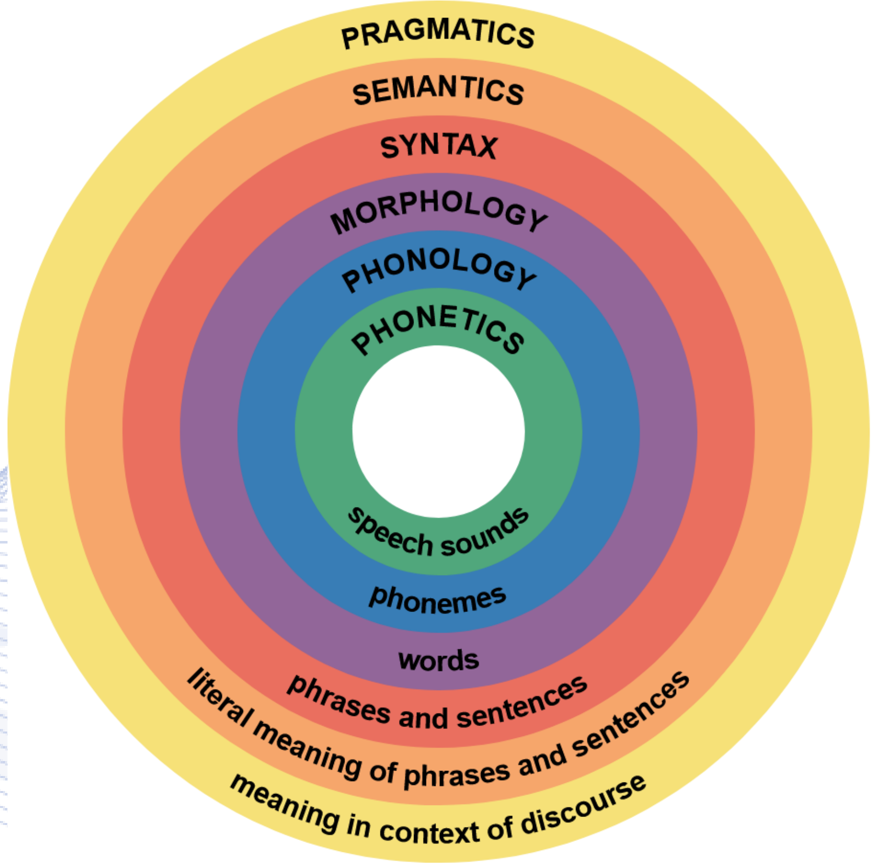
- Messy
- Ambiguous
- changing and evolving
- Not defined by formal rules

So it’s hard working with such data.

From Linguistics to NLP

Linguistics is the scientific study of language, including its grammar, semantics, and phonetics

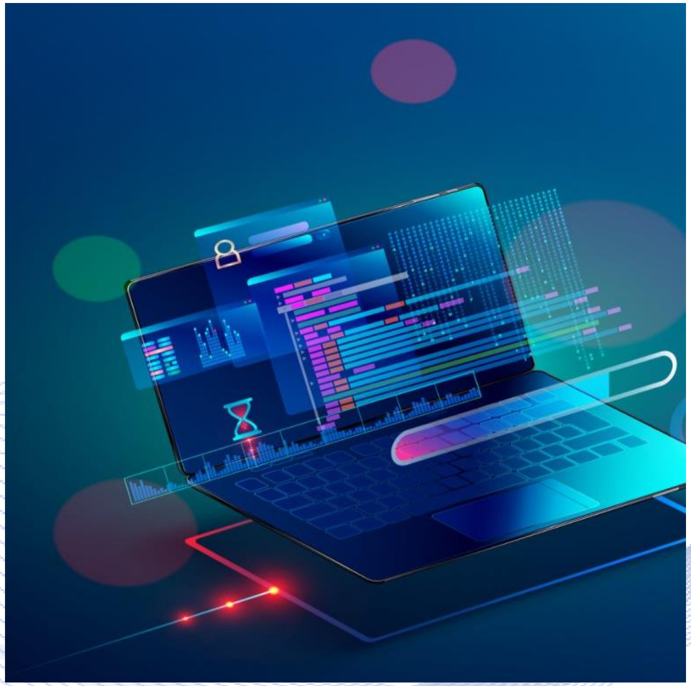
Many problems in natural language understanding resist clean mathematical formalisms



Computational Linguistics

The study of linguistics using the tools of computer science

Use computers to handle large text data efficiently and lead to new discoveries



CL vs NLP

Computational linguistics has both a scientific and an engineering side.

- **Engineering side** -> Natural language processing (NLP): building computational tools that do useful things with language
- **Scientific side** -> Seeks to study/understand language using computers and corpora

Same means , different goal

NLP researchers will build a useful system and show that it works really well

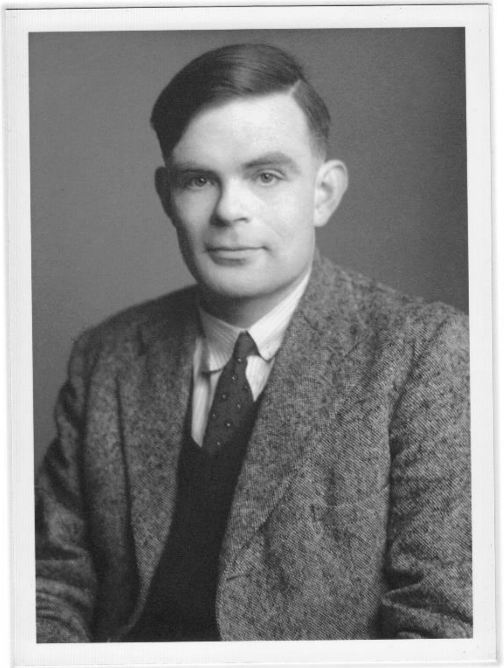
CL researcher would be more interested in which features are useful indicators and why

History

Roots

Alan Turing ,1950, "Computing Machinery and Intelligence" , Turing test -> Test if a machine exhibits human-like intelligence

This task involves the automated interpretation and generation of natural language

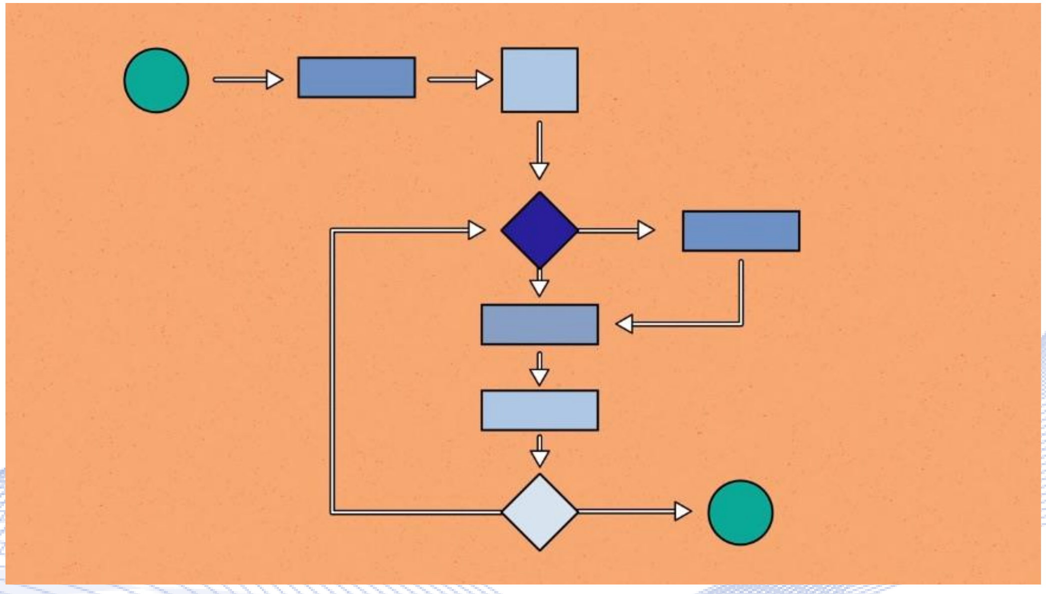


History - periods

- Symbolic NLP (1950s - early 1990s)
- Statistical NLP (1990s - 2010s)
- Neural NLP (2010s - present)

Symbolic NLP (classical programming)

Given a collection of hand written rules the computer emulates natural language understanding by applying those rules to the data it is confronted with



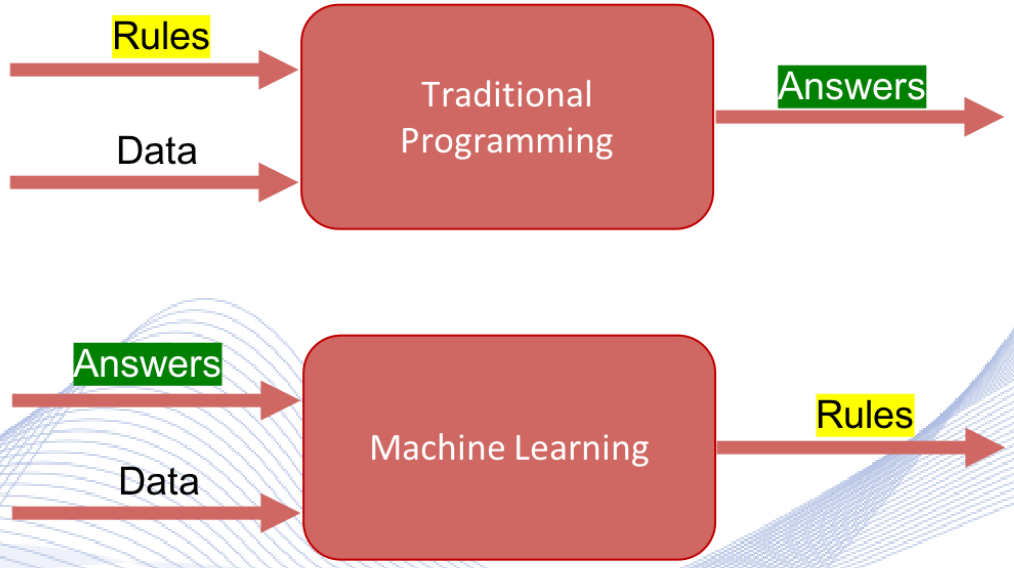
Symbolic NLP

- **1950s**: Georgetown experiment - fully automatic translation
- **1960s**: ELIZA - a simulation of a psychotherapist
- **1970s**: PARRY - the first chatterbot
- **1980s**: Lesk algorithm - rule-based parsing, semantics

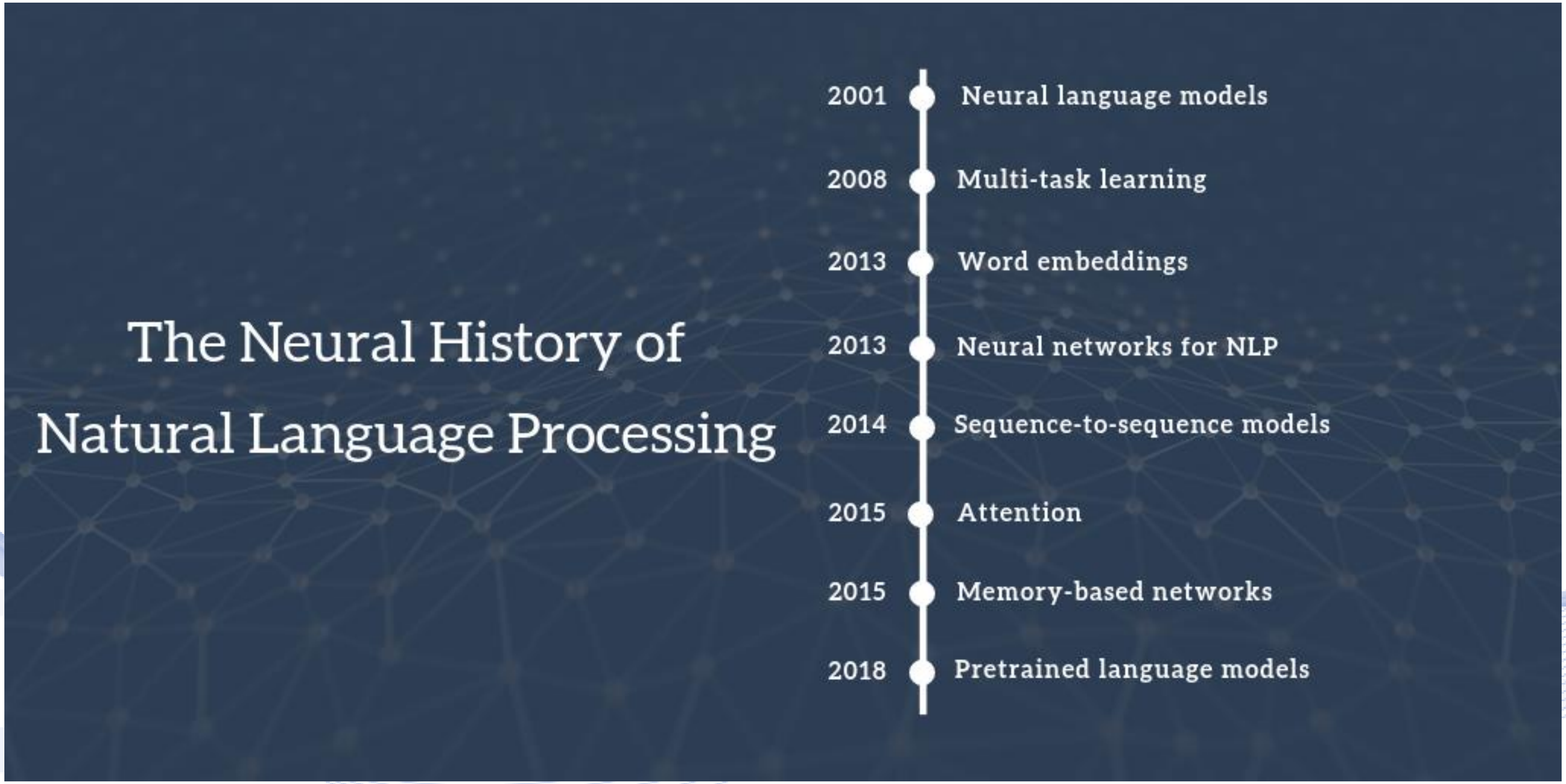
Statistical NLP

Up to 1980s -> hand-written rules

Late 1980s -> statistical approach / Data-Driven methods / machine learning algorithms



Neural NLP



Artificial Intelligence & Information Analysis Lab

Neural NLP

- **2001 - Neural language models**: A feed-forward neural network was proposed by Bengio for language modelling

- **2008 - Multi-task learning**: Sharing the look-up tables (word vectors) between two models trained on different tasks was proposed by Collobert

- **2013 - Word embeddings**: Word2Vec was proposed by Mikolov to learn vector representations from huge corpora

Neural NLP

- **2013 - Neural networks for NLP**: RNNs (Sutskever) and CNNs (Kalchbrenner) started to get adopted in NLP, as well as the combination of those (Wang)

- **2014 - Sequence-to-sequence models**: Encoder – Decoder architecture proposed by Sutskever

- **2015 - Attention**: This mechanism, proposed by Bahdanau, allowed the decoder to look back at the source sequence hidden states

Neural NLP

- **2015 - Memory-based networks**: Models with a more explicit memory have been proposed by Graves, Weston etc.

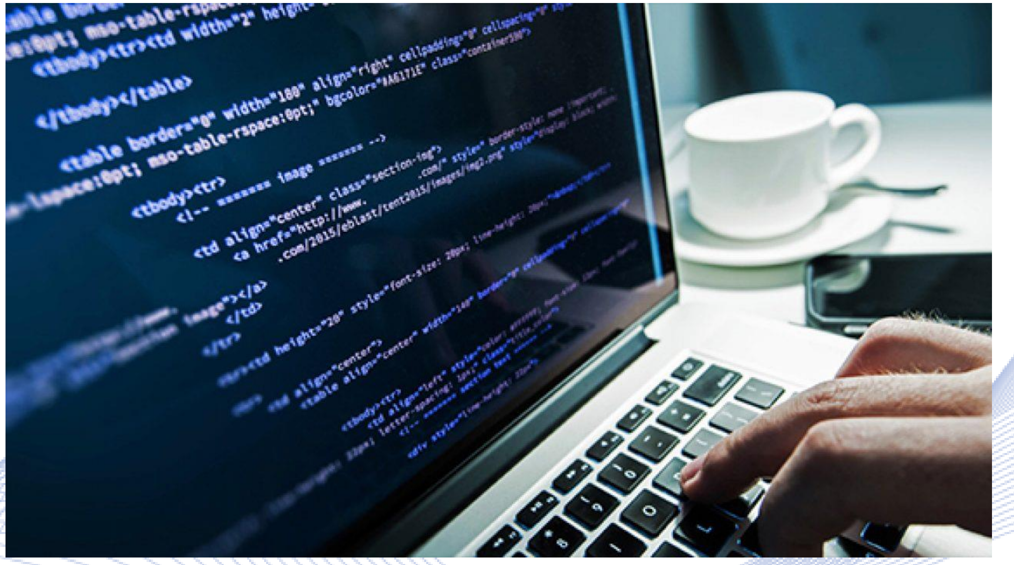
- **2018 - Pretrained language models**: Language models trained in huge corpora to find good embeddings can now be used for diverse range of downstream tasks – e.g BERT, proposed by Devlin et al. which is the current SOTA across a variety of NLP tasks

Methods: Rules, statistics, neural networks

Artificial Intelligence &
Information Analysis Lab

Rules

Hand-coding of a set of rules, coupled with a dictionary lookup: such as by writing grammars or devising heuristic rules for stemming



Statistics

Statistical revolution (1990s) -> Machine learning

Machine-learning -> Using statistical inference to automatically learn such rules through the analysis of large corpora

These algorithms take as input a large set of "features" that are generated from the input data by the programmer (manually)

Why ML ?

Cons of hand-crafted rules

- Not at all obvious where the effort should be directed
- Handling unfamiliar and erroneous input is extremely difficult
- Systems can only be made more accurate by increasing the complexity of the rules -> hard process

ML Algorithms for NLP

Supervised ML

- Support Vector Machines
- Bayesian Networks
- Maximum Entropy
- Conditional Random Field
- Decision Trees
- Random Forests
- K-nearest Neighbor

Unsupervised ML

- Clustering
- Latent Semantic Indexing
- Matrix Factorization

NNs Feature Learning

ML methods -> Manual Feature Extraction

Cons of manually designed features

- Overspecified or incomplete
- Long time to design and validate
- Only get you to a certain level of performance

NNs Feature Learning

Neural networks -> automatic feature learning

Pros of learned features

- Continually and automatically improve
- Easy to adapt
- Fast to train

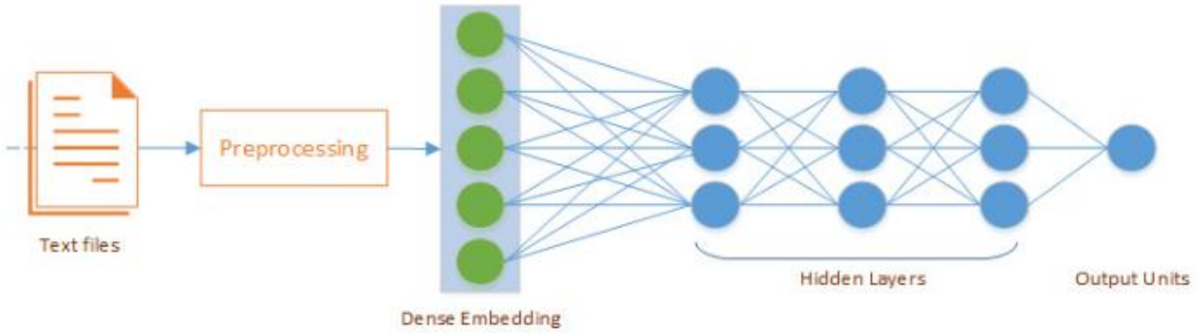
Artificial Intelligence & Information Analysis Lab

Types of NNs for NLP

- Embedding Layers
- Multilayer Perceptrons (MLP)
- Convolutional Neural Networks (CNNs)
- Recurrent Neural Networks (RNNs)
- Hybrid – Combinational Neural Networks

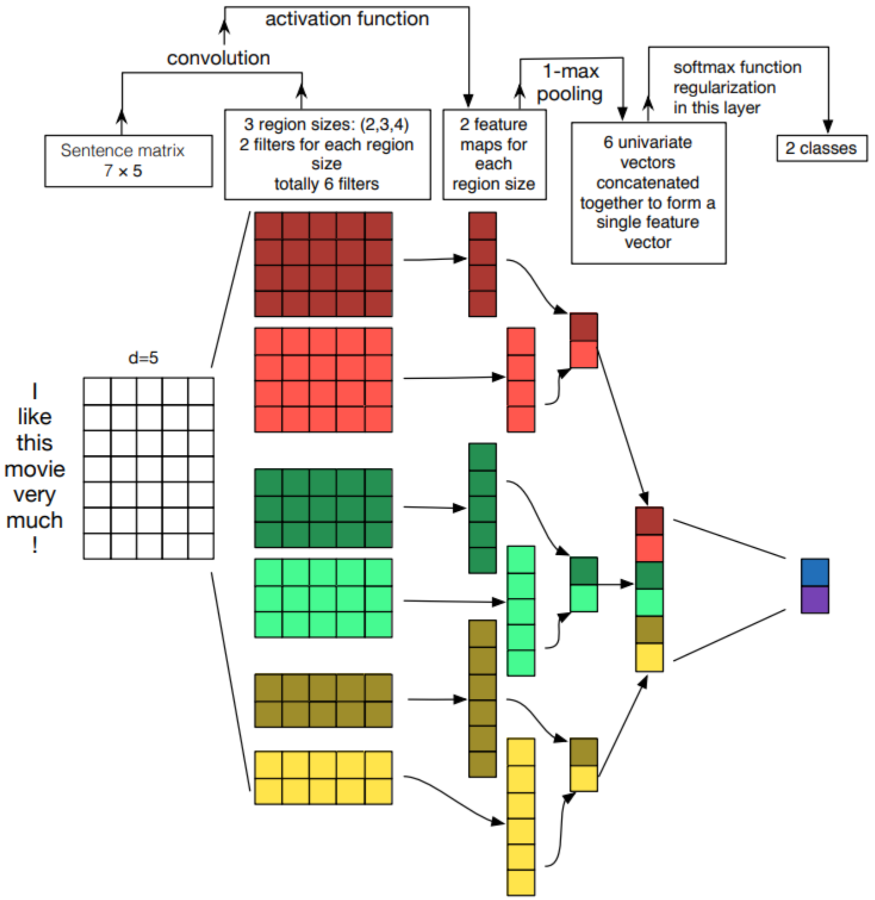
Embedding Layer

- Used on the front end of a neural network
- The one-hot encoded words are mapped to the word vectors
- Word vectors are concatenated before being fed as input to an MLP
- Each word may be taken as one input in a sequence when using RNN



Convolution Neural Networks (CNNs)

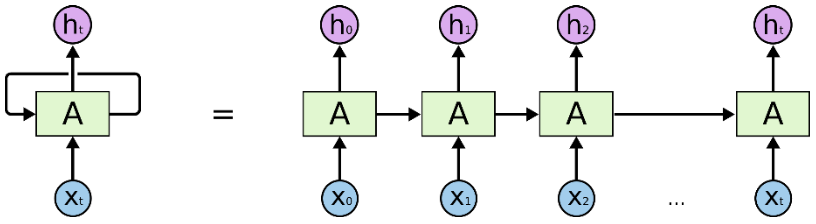
- Traditionally used in image processing
- Can be also used for text
- Easily parallelized for GPUs
- Good classification results



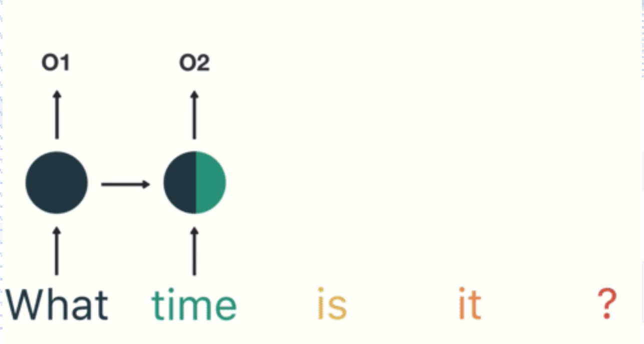
Source: Zhang and Wallace (2015)

Recurrent Neural Networks (RNNs)

- Good for dealing with sequential data (as text)
- Consider information of previous nodes
- Why is it useful ?
  - Example: Try to predict the direction of a ball moving
- RNNs mostly works by using LSTM or GRU for text classification (due to vanishing gradient problem)



Source: colah's blog



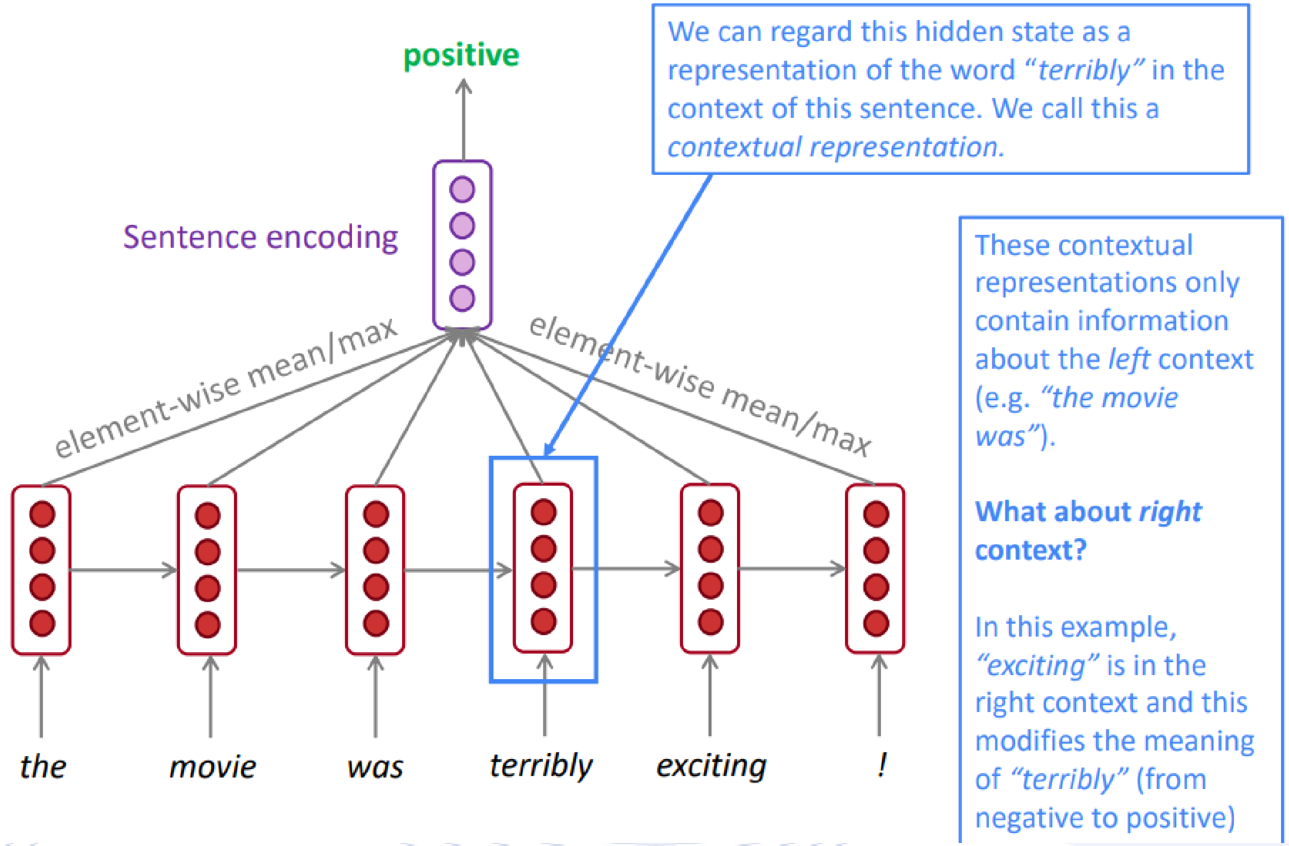
Source: link

Artificial Intelligence & Information Analysis Lab

Recurrent Neural Networks (RNNs)

- Other types
  - Bi-directional RNNs
  - Multi-layer RNNs

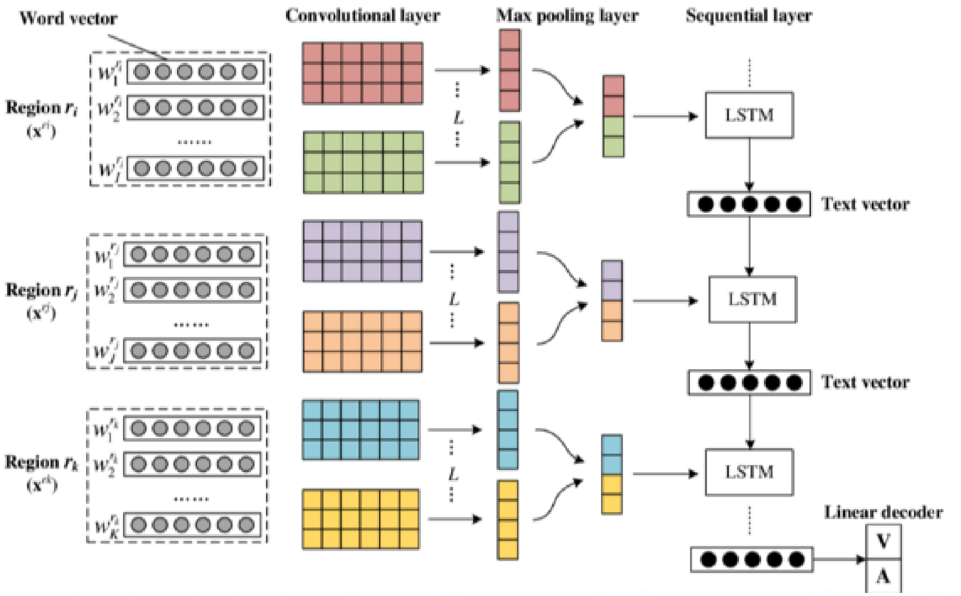
- Disadvantages
  - Slow
  - Difficult to be parallelized

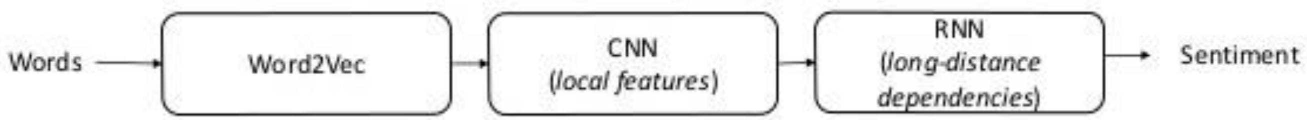


Source: *Stanford's lectures from* Natural Language Processing with Deep Learning CS224N/Ling284

Artificial Intelligence & Information Analysis Lab

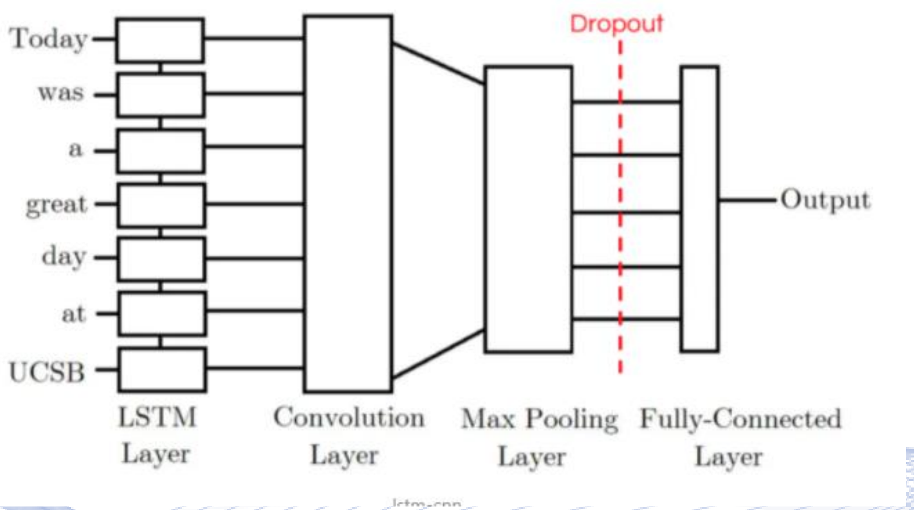
CNN + RNN



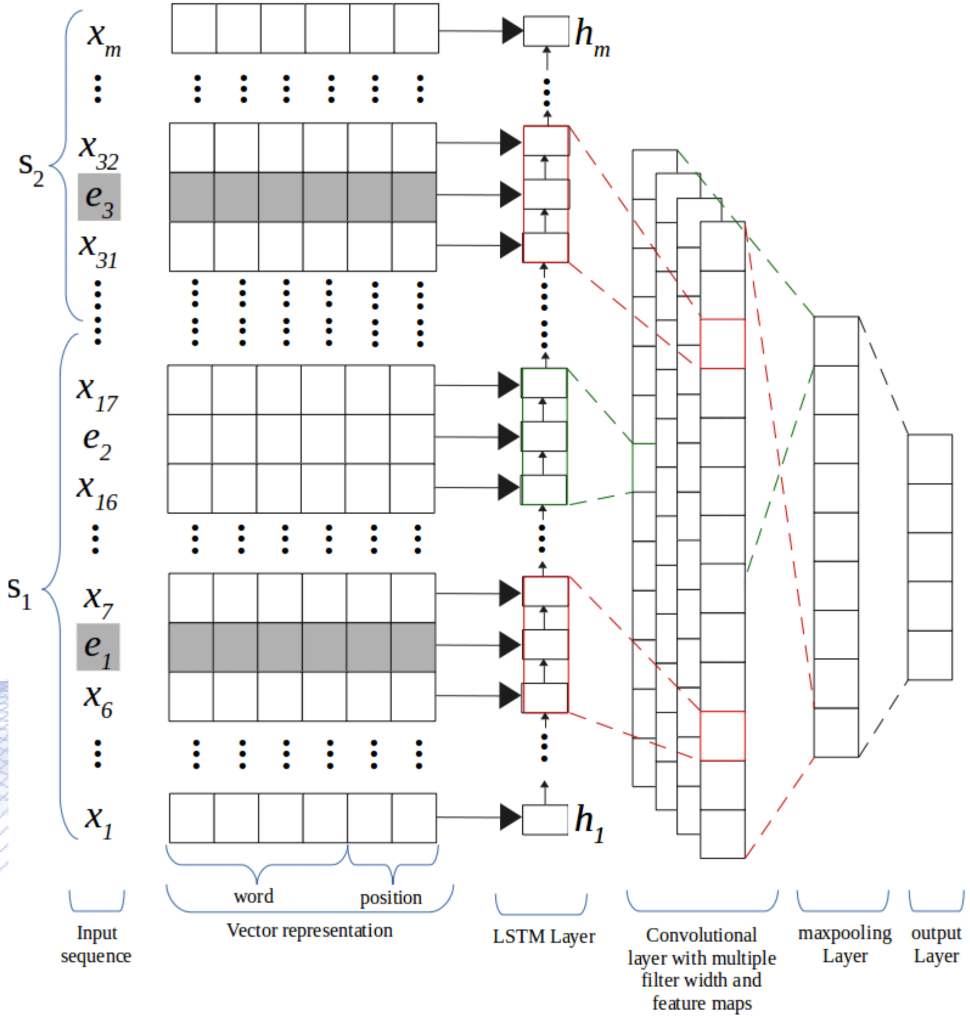


Takes advantage of the coarse grained local features generated by CNN and long-distance dependencies learned via RNN

RNN + CNN

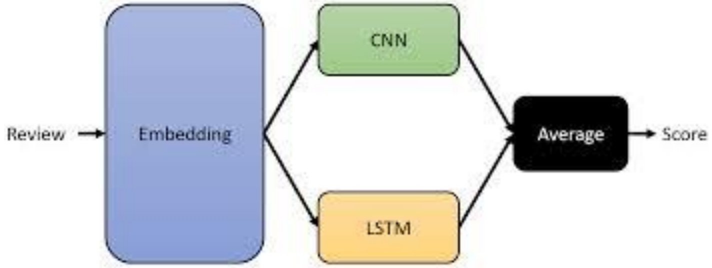


Today
was
a
great
day
at
UCSB
Dropout
Output
LSTM Layer
Convolution Layer
Max Pooling Layer
Fully-Connected Layer

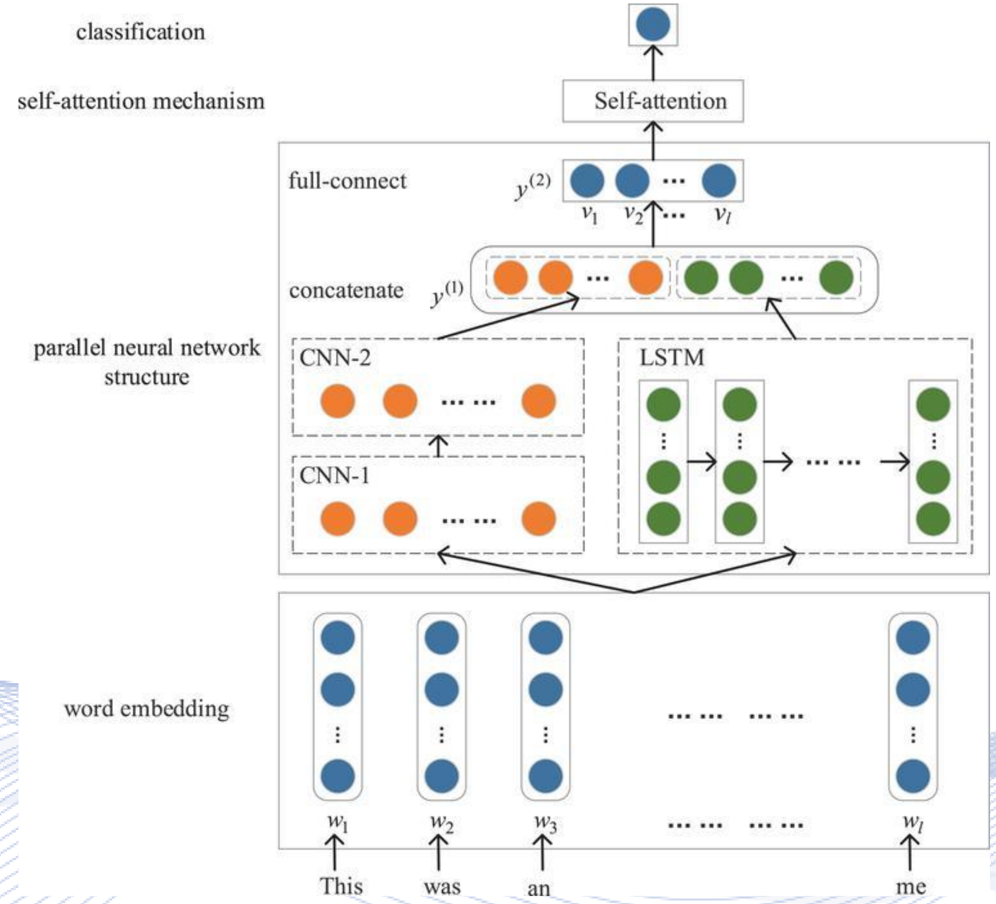


x_m
h_m
S_2
x_{32}
e_3
x_{31}
x_{17}
e_2
x_{16}
S_1
x_7
e_1
x_6
x_1
h_1
word
position
Input sequence
Vector representation
LSTM Layer
Convolutional layer with multiple filter width and feature maps
maxpooling Layer
output Layer

CNN // RNN



Review
Embedding
CNN
LSTM
Average
Score



classification
self-attention mechanism
Self-attention
full-connect
$y^{(2)}$
v_1 v_2 ... v_l
concatenate
$y^{(1)}$
parallel neural network structure
CNN-2
CNN-1
LSTM
word embedding
w_1 w_2 w_3 w_l
This
was
an
me

Word representations

Words represented as vectors

Word representations (vectors)

- **Fixed representations (sparse)**

  Every dimension/value of the feature vector corresponds to a specific word

- **Distributed representations (dense)**

  Features/dimensions of the vector do not correspond to words from the vocabulary but to some meaning/entity

Fixed representations

- **One-Hot Encoding**
- **Bag of words** (count vectors), **TF-IDF**

One-Hot Encoding

- Vocabulary size vector
- Only corresponding column value is 1

Color
Red
Red
Yellow
Green
Yellow

Red	Yellow	Green
1	0	0
1	0	0
0	1	0
0	0	1

Artificial Intelligence &
Information Analysis Lab

Bag of words

BOW Representation

Representing the sentence, “it is the best of the best ”

It	is	the	best	of	a	an
[1	,1	,1	,1	,1	,0	,0]

(only the words present in the document are activated)

(or)

[1	,1	,2	,2	,1	,0	,0]

(the word count is taken into consideration instead of activation)

- Represent context (phrases, sentences, paragraphs, documents)
- Binary values (binary BOW)
- Frequency counts (BOW)

TF-IDF

- Term Frequency-Inverse Document Frequency scores

			TF			TF-IDF		
Term	**DF**	**IDF**	**d1**	**d2**	**d3**	**d1**	**d2**	**d3**
car	18,16	1.6	27	4	24	44.5	6.6	39.6
autol	6,72	2.0	3	33	0	6.2	68.6	0
insur.	19,24	1.6	0	33	29	0	53.4	46.9
best	25,23	1.5	14	0	17	21	0	25.5

Distributed representations (embeddings)

- **Classic word embeddings**
  - Word2Vec (2013)
  - GloVe (2014)
  - FastText (2016)
- **Contextualized word embeddings**
  - CoVe (2017)
  - ELMo (2018)
  - OpenAI GPT (2018)
  - BERT (2018)

About Embeddings

- Word semantics are *embedded* in the vector representation
- Similar meanings -> similar representations

- Embedding values are learned like the weights of a NN in training
- Language models are used to train/learn good embeddings with large corpora, in a word prediction task

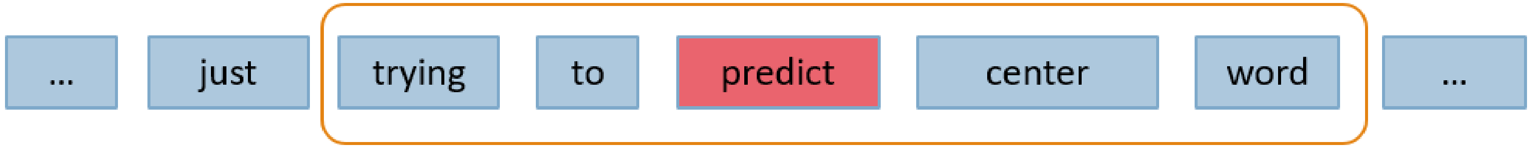
Classic word embeddings

Language models for learning embeddings

Word2Vec (concept)

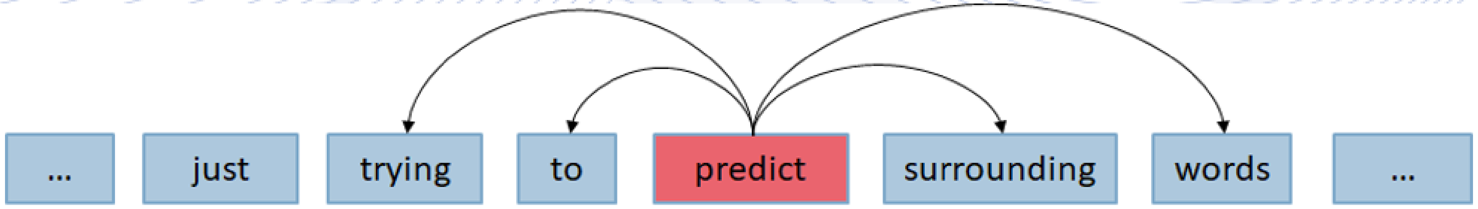
- Continuous Bag-of-Words (CBOW)
  - Predict center word given surrounding words

input1	input2	input3	input4	output
trying	to	center	word	predict



- Skip-Gram
  - Predict surrounding words given center word

Input	output1	output2	output3	output4
predict	trying	to	center	word



GloVe

Marry the **global** text statistics of matrix factorization techniques like Latent Semantic Analysis with the **local** context-based learning in word2vec

- Constructs a matrix of term co-occurrences from the whole corpus

  For each word (e.g. water), compute P(k|water) = probability of k and water to co-occur, where k=word from the vocabulary

- High-dimensional context matrix is reduced by normalizing counts and log-smoothing

FastText

- Extends word2vec's skip-gram model which ignored the internal structure of words, by taking into account morphology
- Subword units are considered, and words are represented by the sum of the vector representations of its character n-grams + word itself

Example: where, n=3 -> <wh, whe, her, ere, re>, <where>

Sharing the representations across words allows to learn reliable representations for rare words (previous models produced poor embeddings for rare words)

Contextualized word embeddings

Language models for learning contextualized embeddings

CoVe

- Embeddings are learned in a translation task
- Encoder-decoder model is trained (supervised)

  Encoder: two-layer bidirectional LSTM

  Decoder: attentional unidirectional LSTMs

- Encoder must learn how to capture syntactic and semantic meanings of words, and output contextualized embeddings
- Then, this pre-trained encoder can be used for a downstream task

ELMo

- Embeddings are learned by training a language model in an unsupervised manner
- Architecture : stacked bidirectional LSTMs
- The model learns to predict the next/previous word given the previous/next ones (bidirectional)

- Still an extra model is needed for downstream tasks (ELMo only gives us the embeddings)

OpenAI GPT

- Unlike ELMo, GPT is trained only to predict the future
- Architecture : Based on Transformer’s decoder
- Can be used directly for all end tasks with only slight modifications

vs ELMo

- ELMo gives richer embeddings due to its bidirectional nature (understands better the context of the word)
- ELMo can’t be used directly for all end tasks (only encoding-front end)

BERT

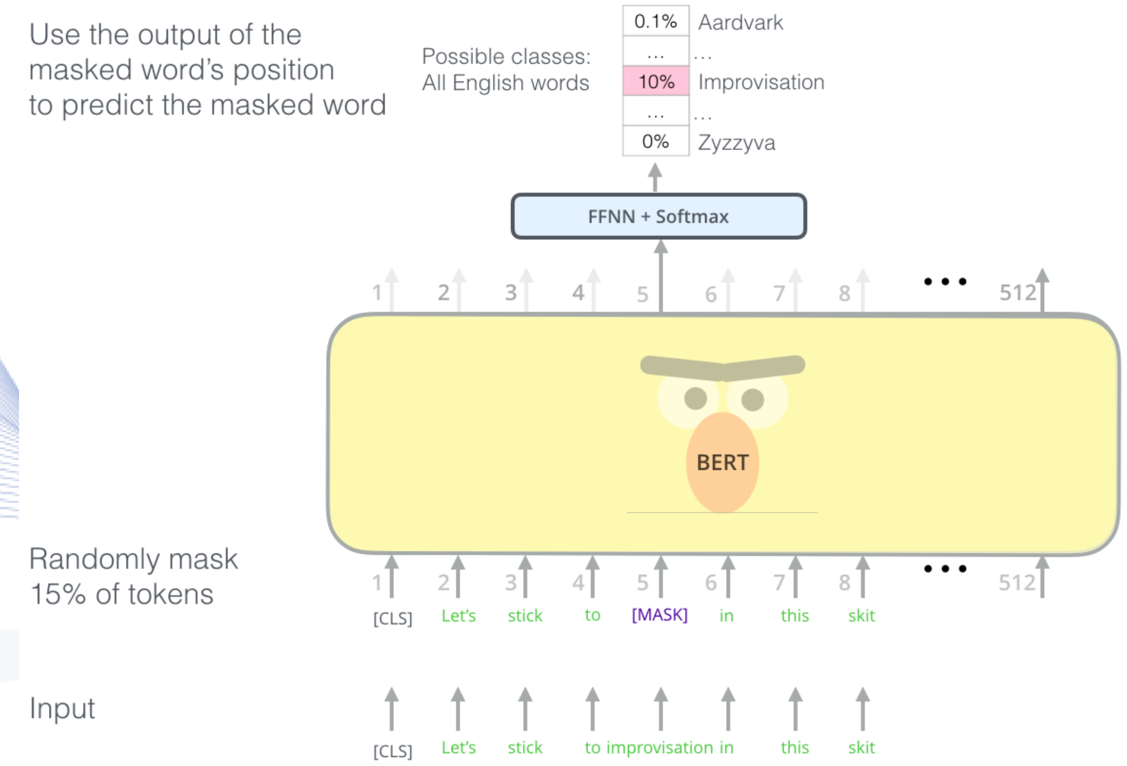
- Architecture : Multi-layer bidirectional Transformer encoder
- It's pre-training (unsupervised) consists of two tasks:

  Mask Language Model: Find the masked/hidden words by looking at their context

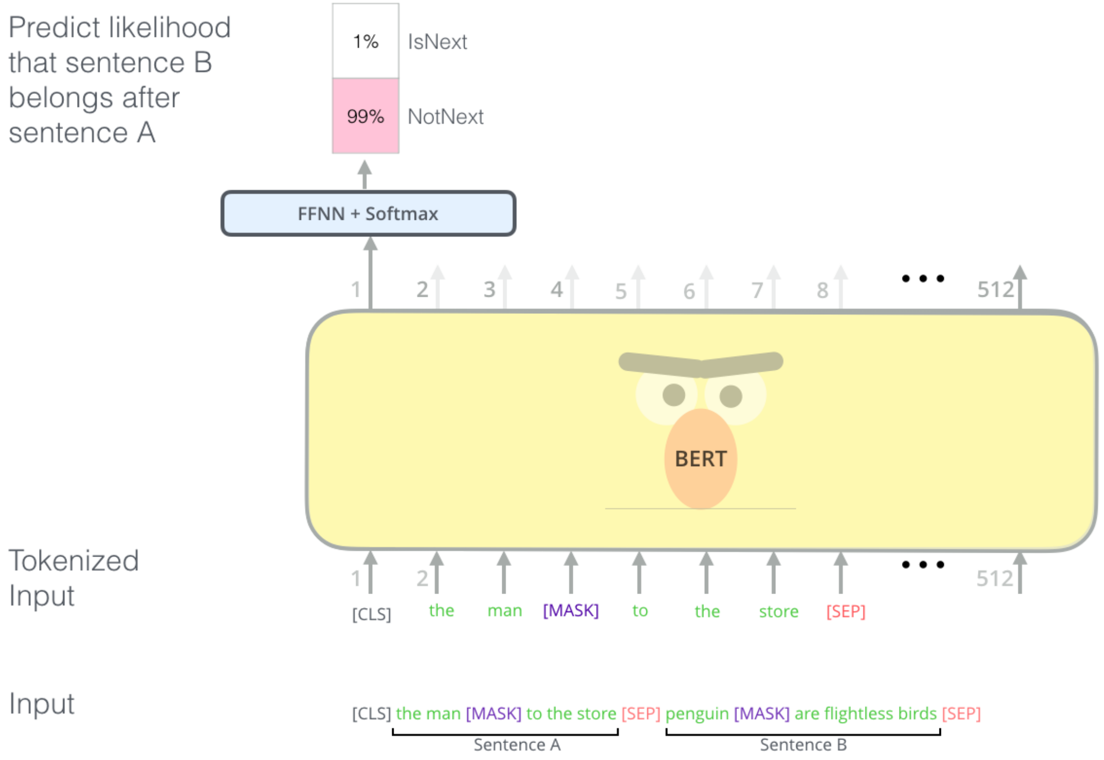
  Next Sentence Prediction: With two sentences as inputs, A and B, predict the order that they appear in

- Trained to predict the context from both left and right (bidirectional)
- Can be used directly for downstream tasks with small modifications

BERT training



Masked Language Model



Two-sentence Tasks

Artificial Intelligence & Information Analysis Lab

Common NLP tasks

Overview

- Text and speech processing
- Morphological analysis
- Syntactic analysis
- Lexical semantics (of individual words in context)
- Relational semantics (semantics of individual sentences)
- Discourse (semantics beyond individual sentences)
- Higher-level NLP applications

Text and speech processing

- **Speech recognition**: Given a sound clip of a person or people speaking, determine the textual representation of the speech

- **Text-to-speech**: Given a text, transform those units and produce a spoken representation

- **Word segmentation (Tokenization)**: Separate a chunk of continuous text into separate words

Higher-level NLP applications

- **Automatic summarization**: Produce a readable summary of a chunk of text

- **Book generation**: Creation of full-fledged books

- **Question answering**: Given a human-language question, determine its answer

- **Machine translation**: Automatically translate text from one human language to another

Q & A

Thank you very much for your attention!

More material in
http://icarus.csd.auth.gr/cvml-web-lecture-series/

Contact: Prof. I. Pitas
pitas@csd.auth.gr