An introduction to Natural Language Processing

With focus on Deep Learning approaches

João Lages

Pedro Balage

outsystems

[PORTO AI]

FARFETCH

Porto AI :: Quarterly Meeting #1

Porto, 30th March

Pedro Balage

- Lead Data Scientist @ Farfetch - Search team
- Research and Industry experience in **NLP**
- Affiliated to Instituto de Telecomunicações
- Member of organization for the Lisbon Machine Learning School (**LxMLS)**

FARFETCH

instituto de telecomunicações

LXMLS 2019 LISBON MACHINE LEARNING SCHOOL

João Lages

- Research Scientist @ Outsystems - AI Team
- Research and Industry experience in **NLP**
- MSc thesis in Deep Learning applied to Information Retrieval

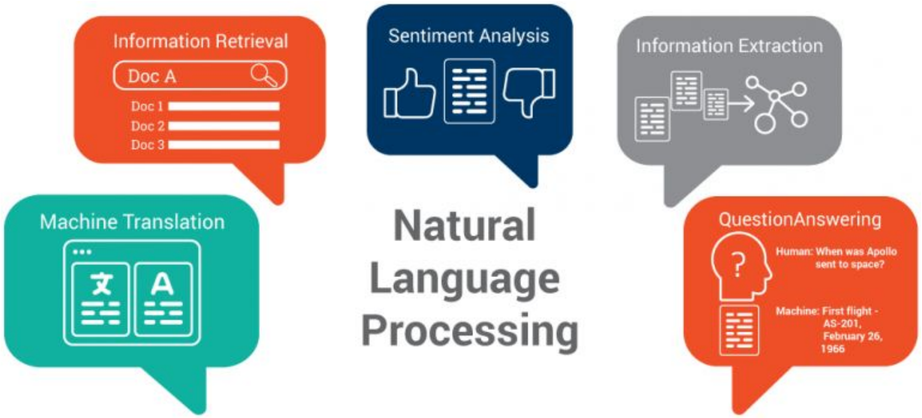
outsystems

JoaoLages

linkedin.com/in/joão-lages-7352a6129/

Why Natural Language Processing?

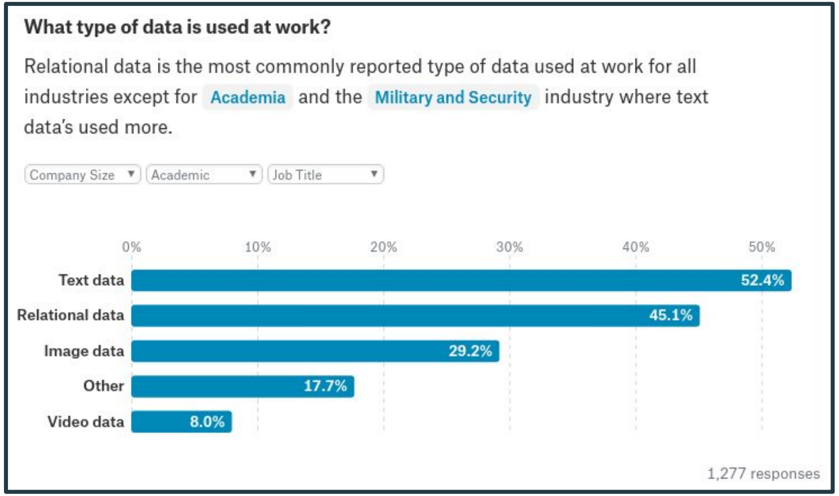
- Natural Language Processing (NLP) is a subfield of Artificial Intelligence that is focused on enabling computers to **understand** and **process human languages**, to get computers closer to a **human-level understanding of language**.



Why Natural Language Processing?

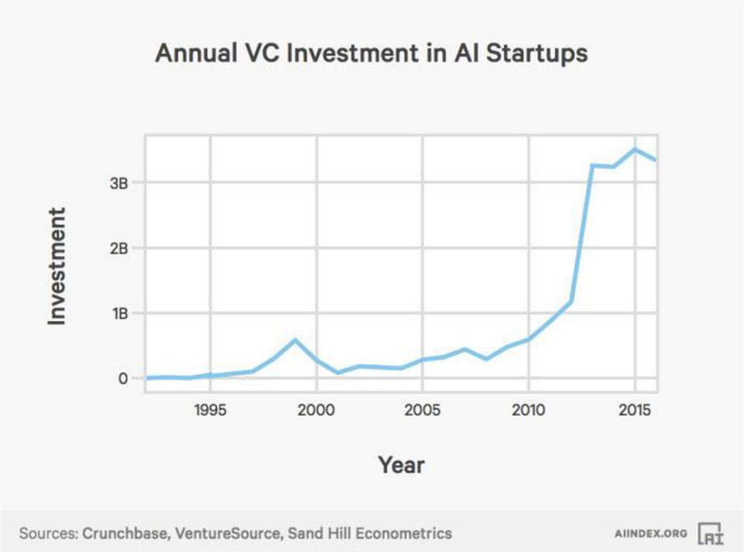
“The **next big step** for **Deep Learning** is **natural language understanding**, which aims to give machines the power to understand not just individual words but entire sentences and paragraphs.”

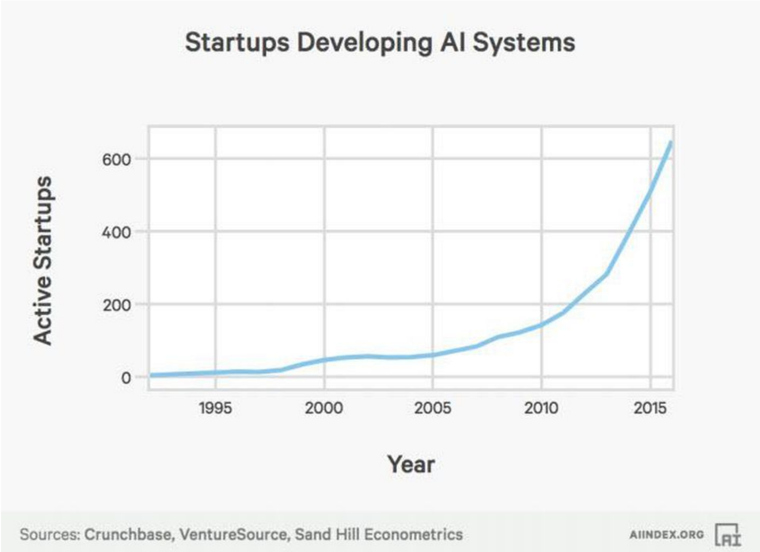
Yann LeCun, June 2015



https://www.kaggle.com/surveys/2017

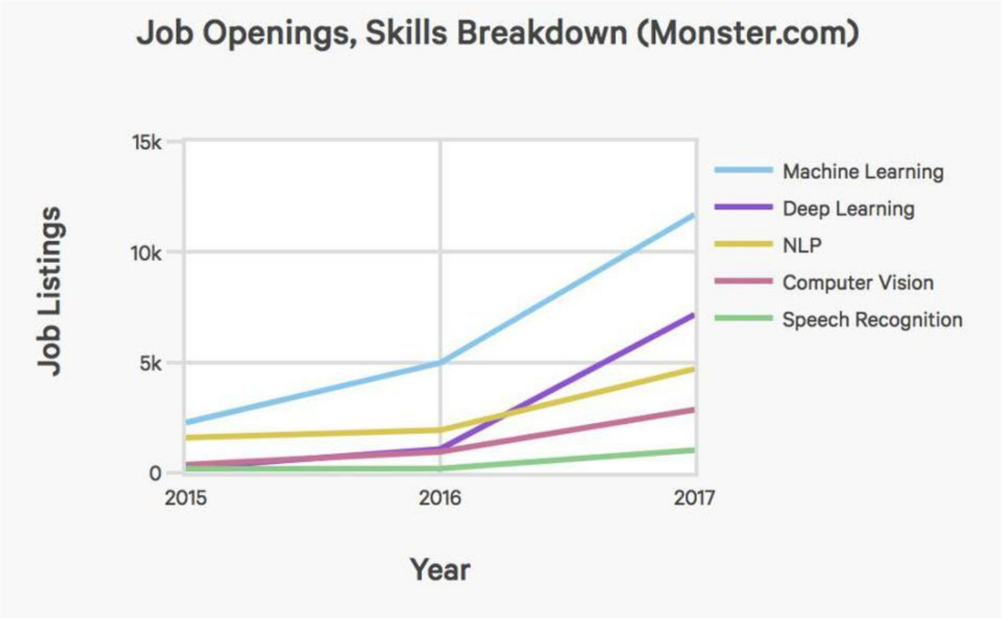
Why should I consider NLP for my career?





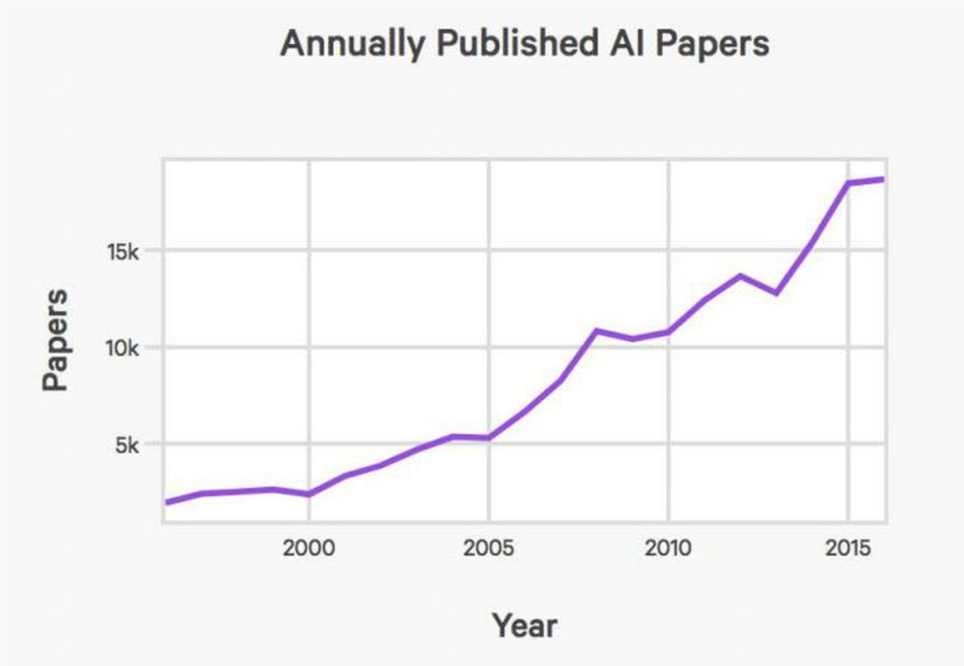
Source: Forbes - 2017 AI Index Report

Why should I consider NLP for my career?

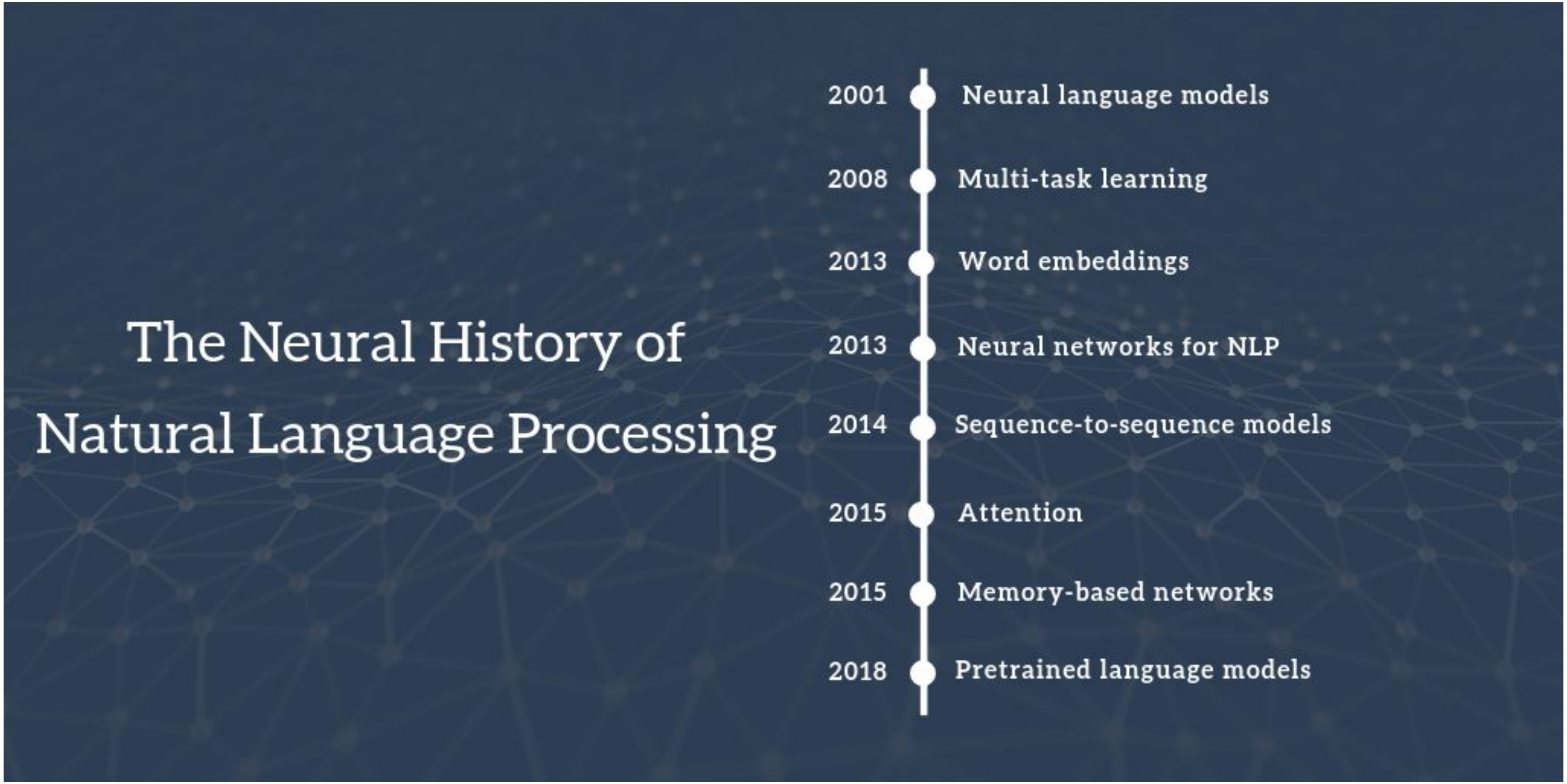


Source: Forbes - 2017 AI Index Report

The development of NLP has a very fast pace!



Source: Forbes - 2017 AI Index Report

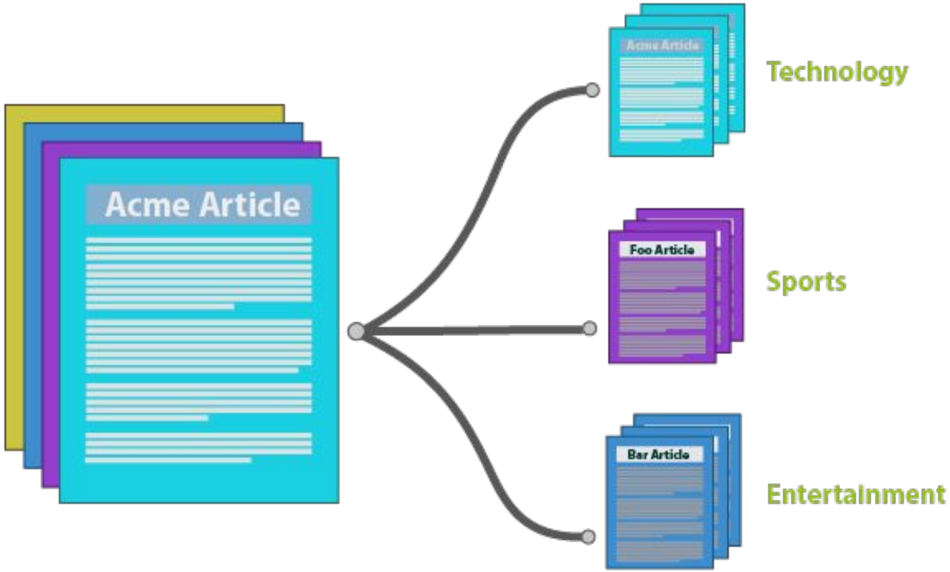


http://ruder.io/a-review-of-the-recent-history-of-nlp/

Some NLP Applications

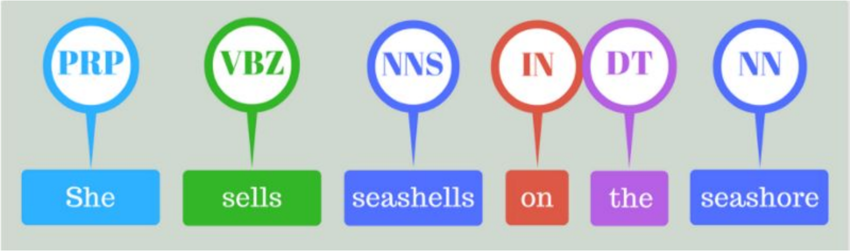
Classification

- Text classification
- Spam classification
- Sentiment analysis



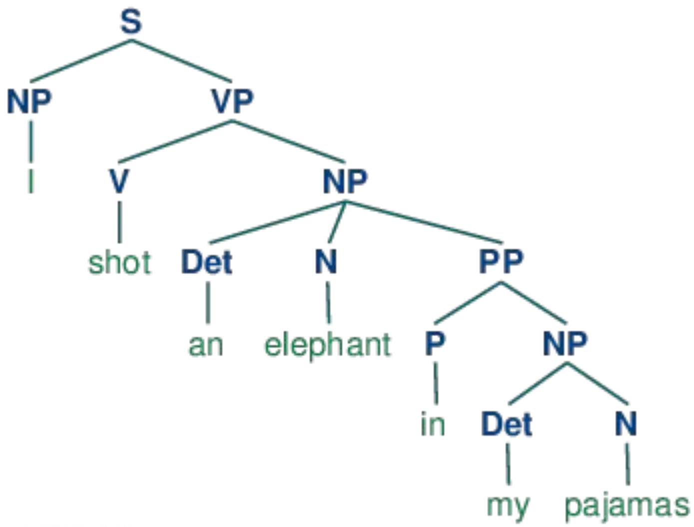
Sequence Labeling

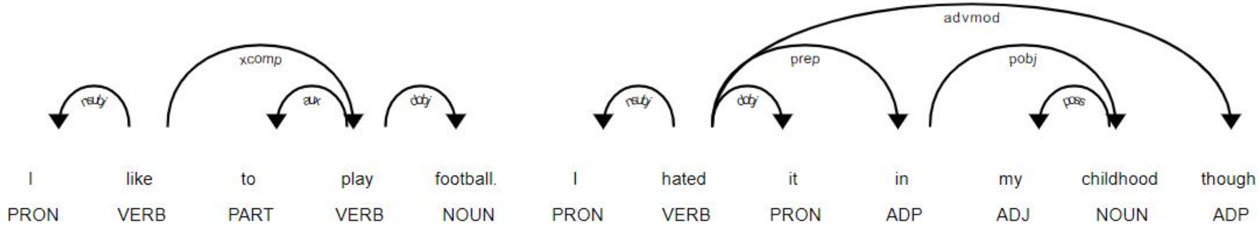
- Part-of-Speech Tagging
- Chunking
- Named-Entity Recognition



Parsing

- Syntactic parsing
- Semantic parsing

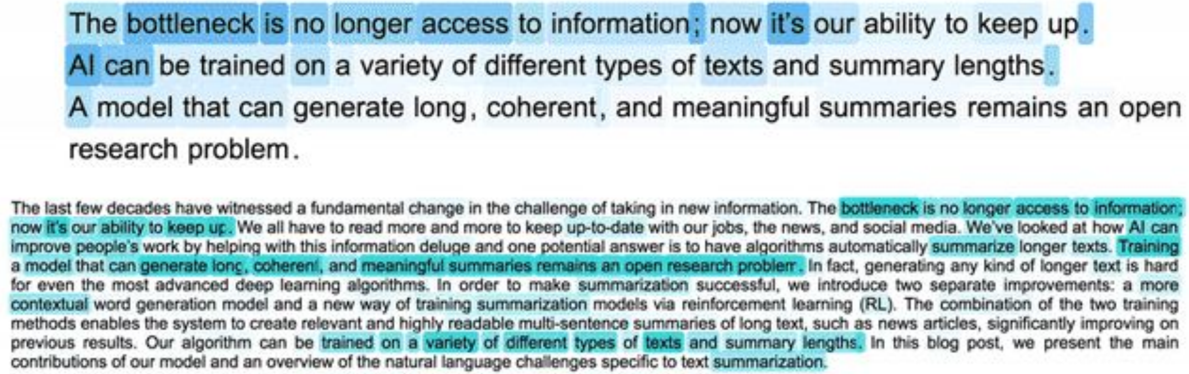




Summarization

Summaries can be:

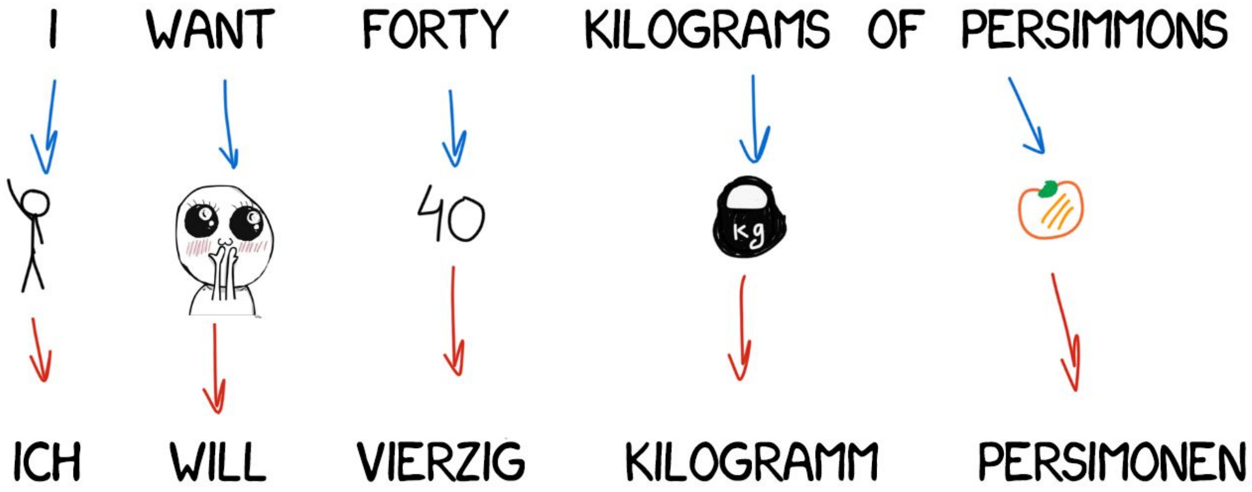
- Extractive
- Compressive
- Abstractive



The bottleneck is no longer access to information; now it's our ability to keep up. AI can be trained on a variety of different types of texts and summary lengths. A model that can generate long, coherent, and meaningful summaries remains an open research problem.

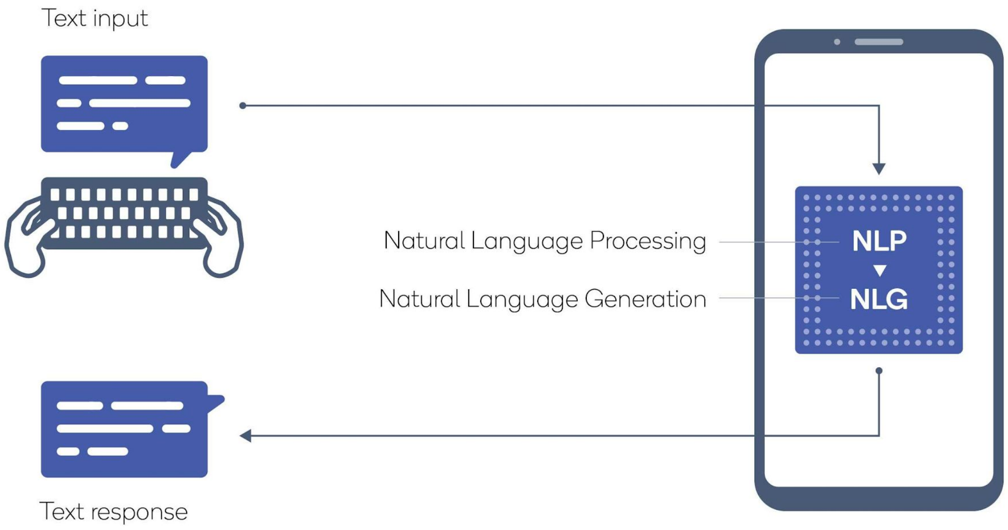
The last few decades have witnessed a fundamental change in the challenge of taking in new information. The bottleneck is no longer access to information; now it's our ability to keep up. We all have to read more and more to keep up-to-date with our jobs, the news, and social media. We've looked at how AI can improve people's work by helping with this information deluge and one potential answer is to have algorithms automatically summarize longer texts. Training a model that can generate long, coherent, and meaningful summaries remains an open research problem. In fact, generating any kind of longer text is hard for even the most advanced deep learning algorithms. In order to make summarization successful, we introduce two separate improvements: a more contextual word generation model and a new way of training summarization models via reinforcement learning (RL). The combination of the two training methods enables the system to create relevant and highly readable multi-sentence summaries of long text, such as news articles, significantly improving on previous results. Our algorithm can be trained on a variety of different types of texts and summary lengths. In this blog post, we present the main contributions of our model and an overview of the natural language challenges specific to text summarization.

Machine Translation

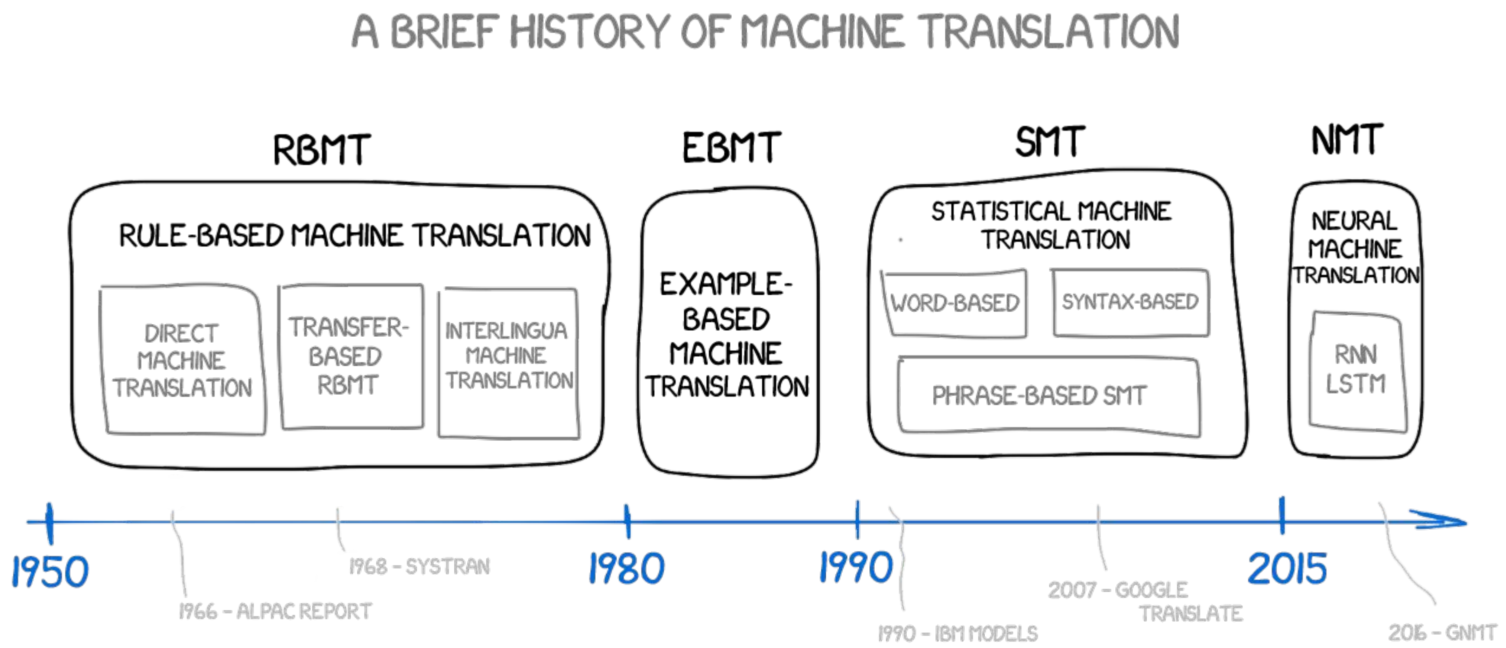


Question Answering

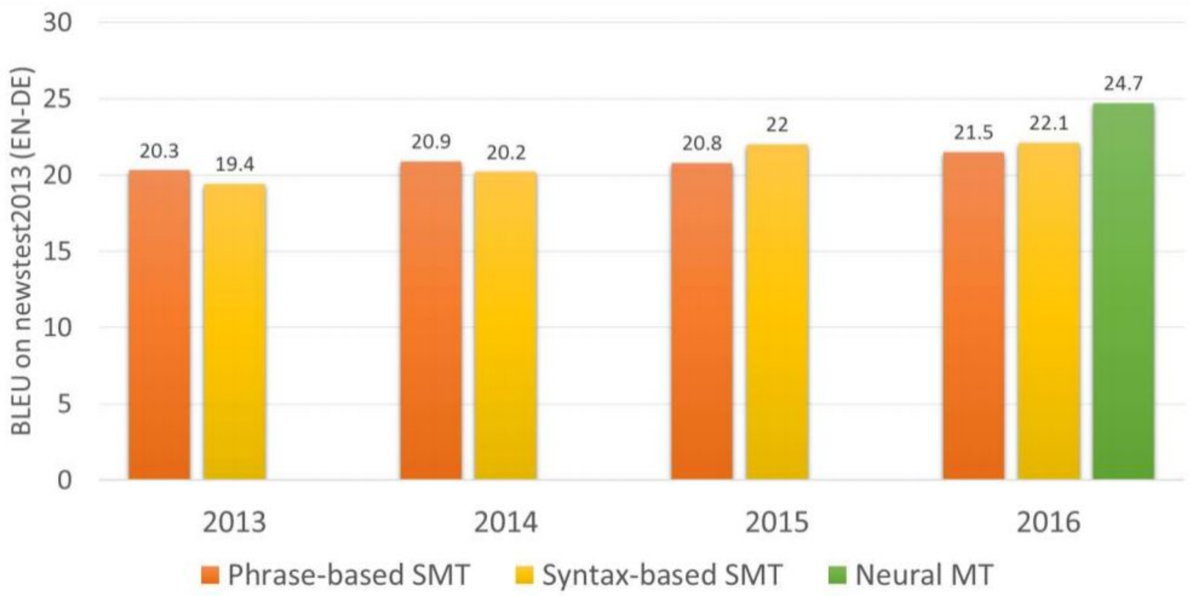
- Question Answering
- Conversational Agents (Chatbots)



Why deep learning approaches to NLP?


A BRIEF HISTORY OF MACHINE TRANSLATION
RBMT
RULE-BASED MACHINE TRANSLATION
DIRECT MACHINE TRANSLATION
TRANSFER-BASED RBMT
INTERLINGUA MACHINE TRANSLATION
EBMT
EXAMPLE-BASED MACHINE TRANSLATION
SMT
STATISTICAL MACHINE TRANSLATION
WORD-BASED
SYNTAX-BASED
PHRASE-BASED SMT
NMT
NEURAL MACHINE TRANSLATION
RNN LSTM
1950
1966 – ALPAC REPORT
1968 – SYSTRAN
1980
1990
1990 – IBM MODELS
2007 – GOOGLE TRANSLATE
2015
2016 – GNMT

Neural approaches are now state-of-the-art



Industry already adopted deep learning

Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi
yonghui,schuster,zhifengc,qvl,mnorouzi@google.com

Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, Jeffrey Dean

Abstract

Neural Machine Translation (NMT) is an end-to-end learning approach for automated translation, with the potential to overcome many of the weaknesses of conventional phrase-based translation systems. Unfortunately, NMT systems are known to be computationally expensive both in training and in translation inference – sometimes prohibitively so in the case of very large data sets and large models. Several authors have also charged that NMT systems lack robustness, particularly when input sentences contain rare words. These issues have hindered NMT's use in practical deployments and services, where both accuracy and

Googe's Neural Machine Translation is a better way to translate text.

Traditional Machine Learning

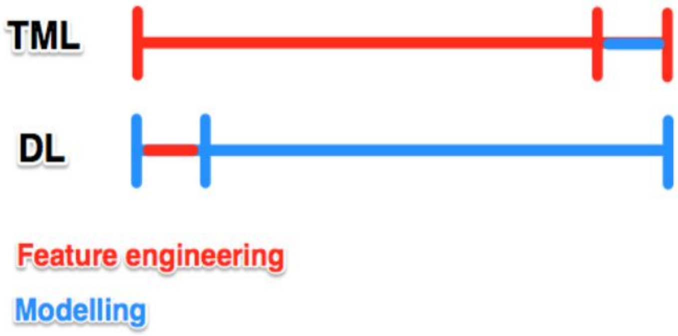
- Representation
  - Representation of my data in a feature space
- Hypothesis Model
  - Machine Learning algorithm to split the space

- What do I want to do?
  - Regression, Classification, Clustering
- Do I have data?
  - Supervised, Unsupervised, Semi-supervised

Deep Learning vs Traditional Machine Learning

- Traditional Machine Learning (TML)
  - Focus on feature engineering

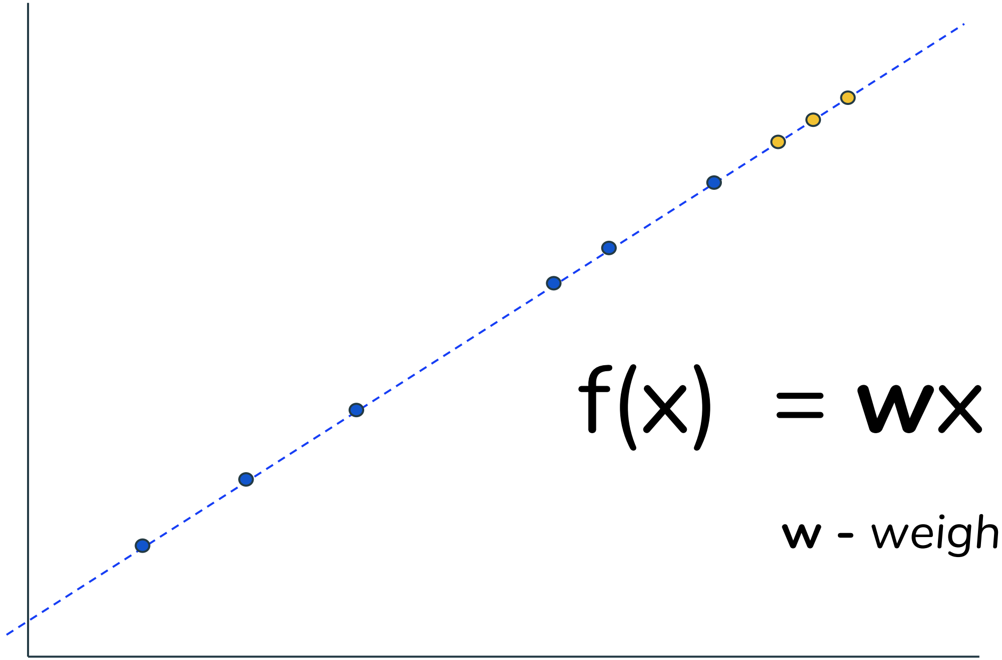
- Deep Learning (DL)
  - Focus on automatic learning word representations



Representations of Language		
Element	TML	DL
Phonology	All phonemes	Vector
Morphology	All morphemes	Vector
Words	One-hot encoding	Vector
Syntax	Phrase rules	Vector
Semantics	Lambda calculus	Vector

A very gentle introduction to deep learning

All started with: Linear Classifiers



$$f(x) = \mathbf{w}x + \boldsymbol{b}$$

w - *weight* **b** - *bias*

All started with: Linear Classifiers

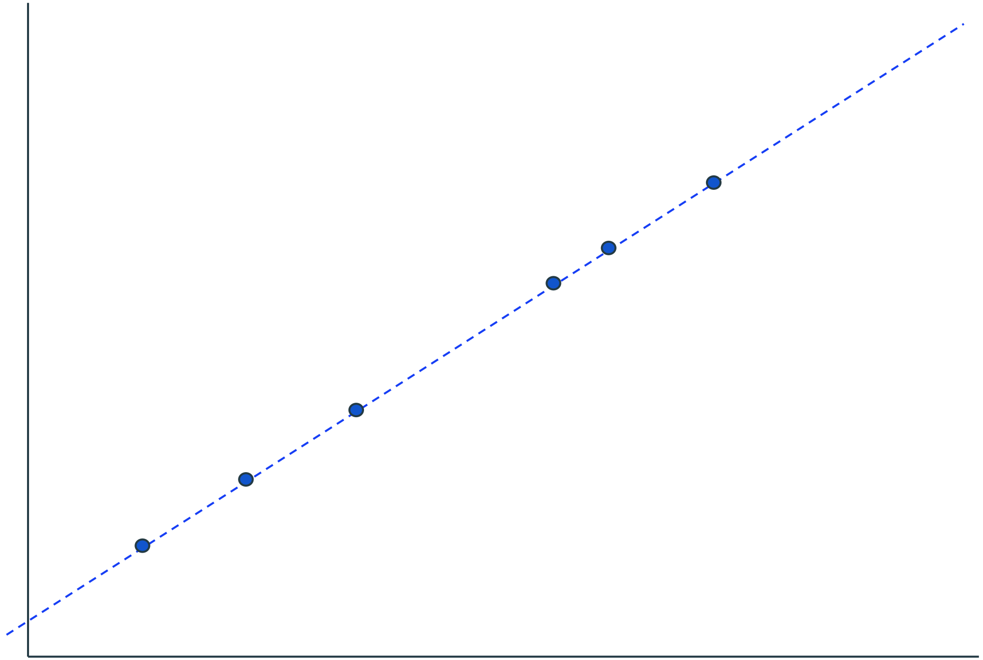
$$\begin{cases} 7 = \mathbf{w}2 + \mathbf{b} \\ 10 = \mathbf{w}3 + \mathbf{b} \end{cases}$$

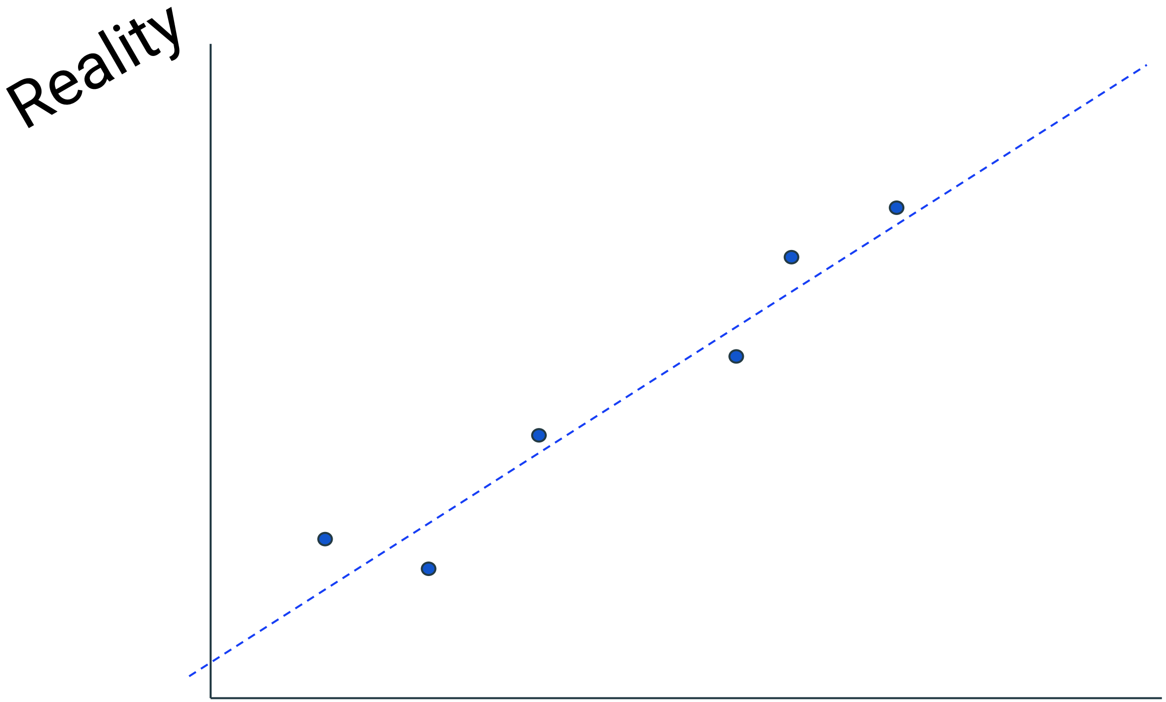
$$\mathbf{w} = 3, \mathbf{b} = 1$$

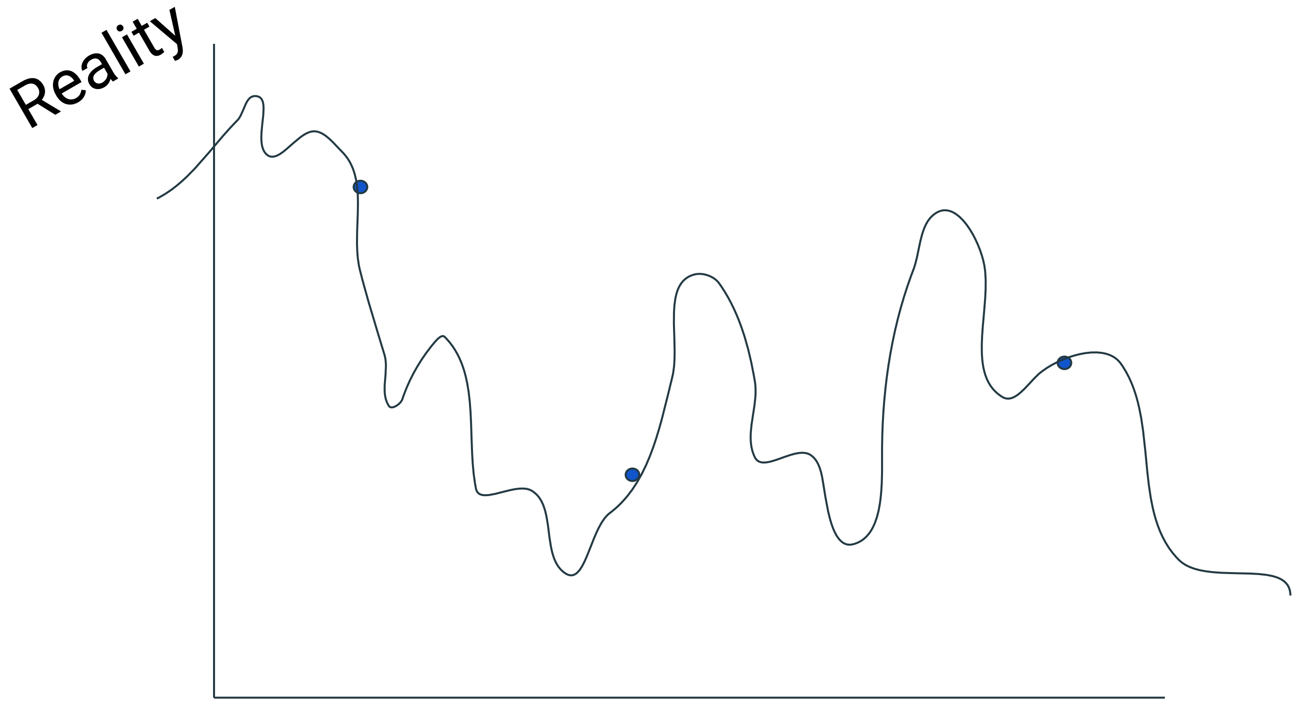
$$f(x) = \mathbf{w}x + \mathbf{b}$$

w - *weight* **b** - *bias*

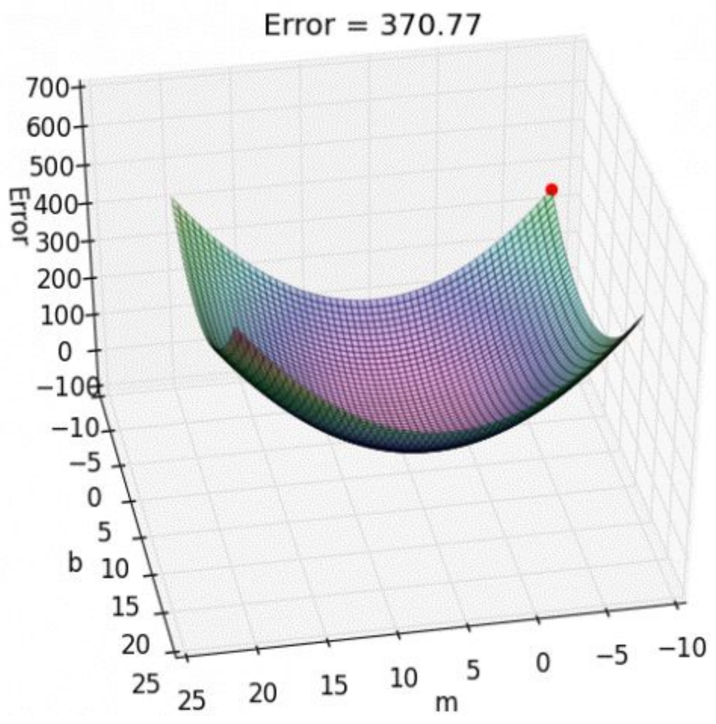
Expectation

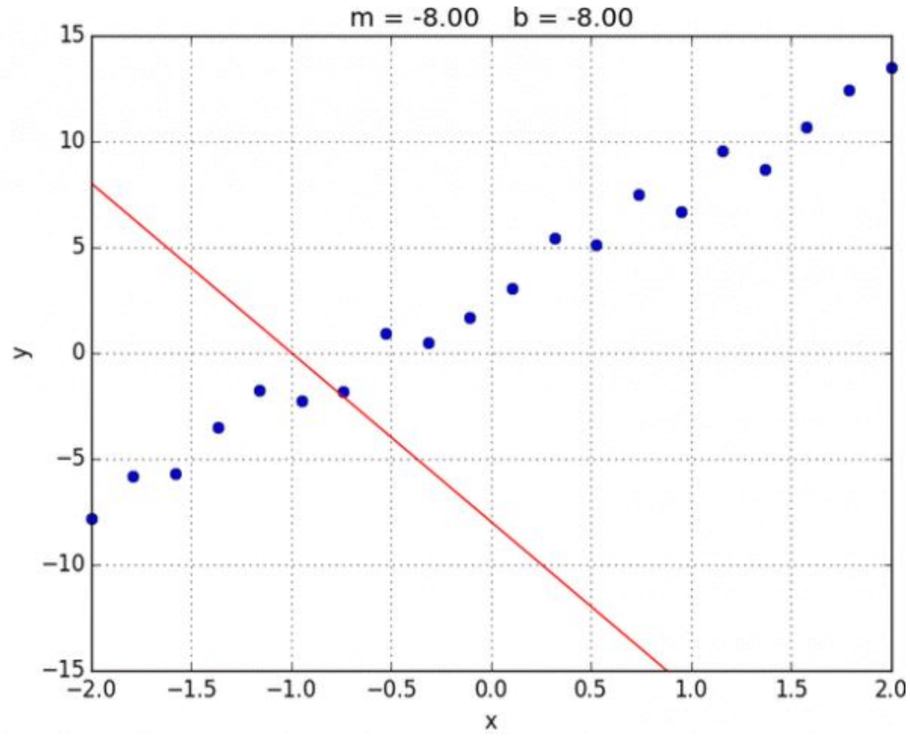



Reality

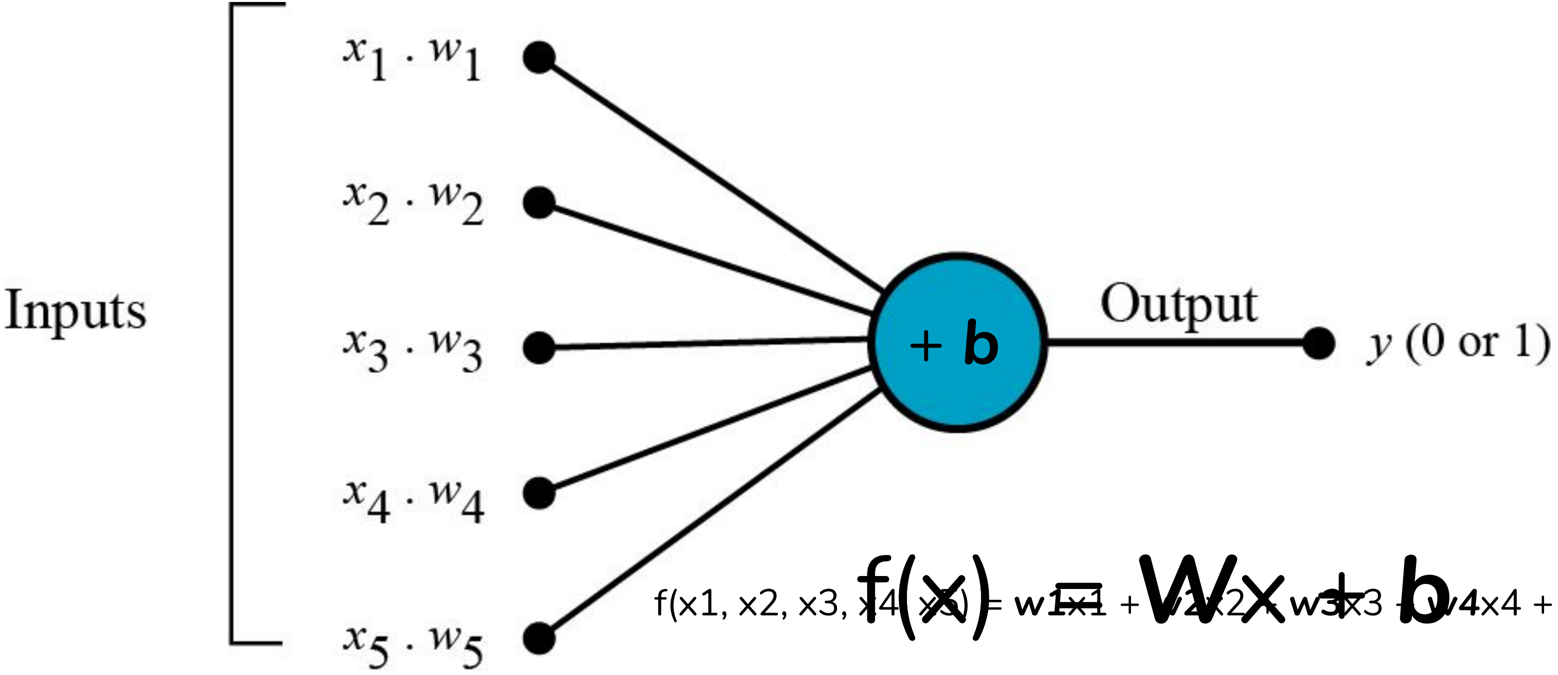

Reality

Gradient Descent - The power to fit the line





Perceptron - The basis of Deep Learning



$$f(x) = Wx + b$$

$$f(x1, x2, x3, x4, x5) = \mathbf{w1}x1 + \mathbf{w2}x2 + \mathbf{w3}x3 + \mathbf{w4}x4 + \mathbf{w5}x5$$

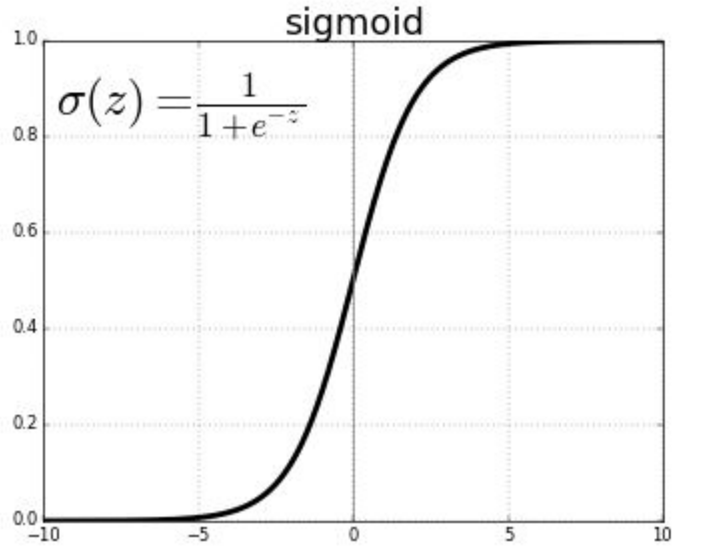
Modelling non-linear functions

Linear:

$$f(x) = \mathbf{W}x + \boldsymbol{b}$$

Non-Linear:

$$f(x) = \sigma(\mathbf{W}x + \boldsymbol{b})$$

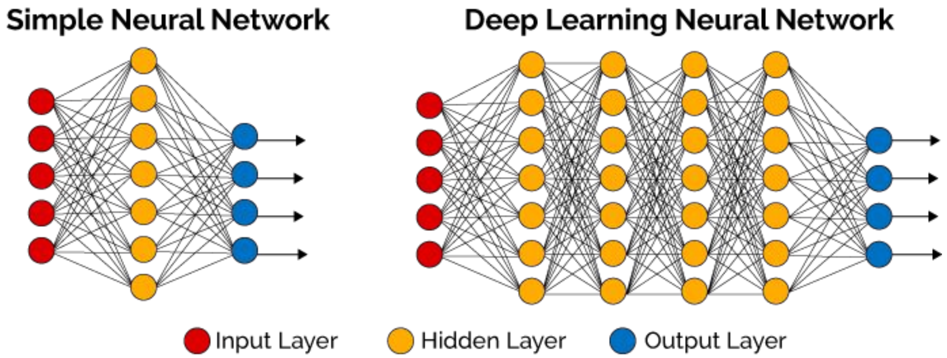


What is Deep learning?

- A **multi-layer perceptron** with more layers

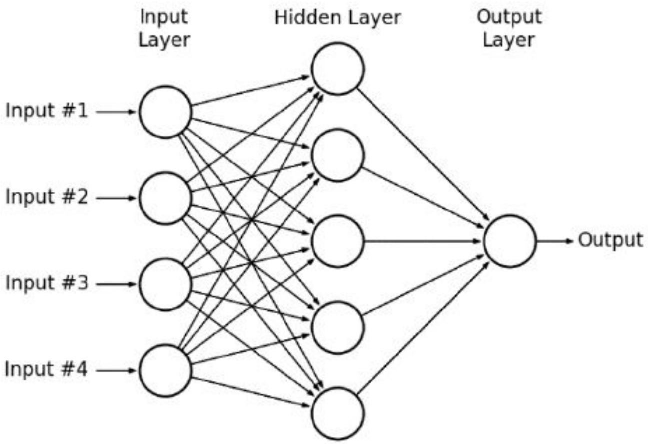
- New architectures:
  - Convolutional Neural Networks (CNN)
  - Recursive Neural Network (RNN)

- Deal with problems in training such huge networks
  - Regularization methods



Key concepts in Machine Learning

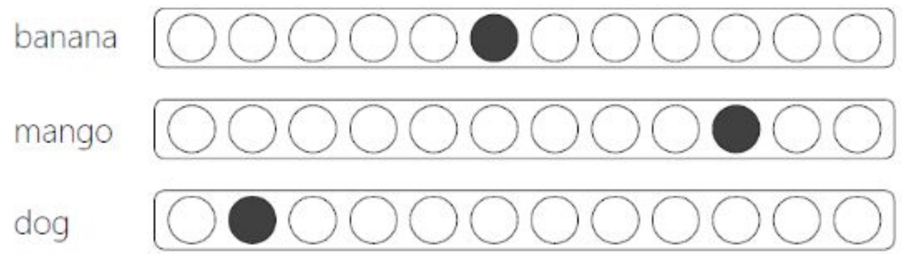
- Let's check some concepts on http://playground.tensorflow.org
    - Train and Test dataset
    - Learning Rate
    - Activation
    - Loss function
    - Hidden Layers
    - Batch size
    - Overfitting and Underfitting



Word Representations

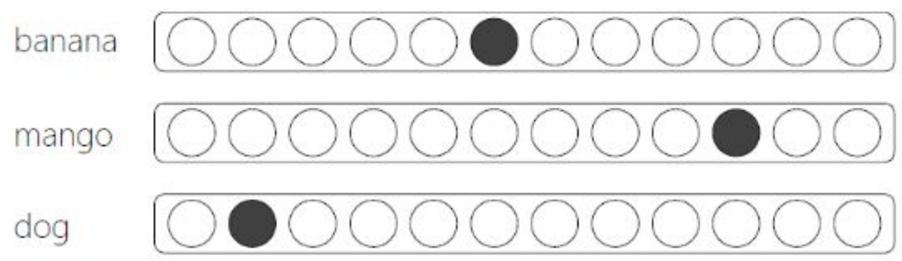
How to represent my words?

- **Local representations**
- Problems with this representation?
  - Sparsity
  - Vectors don't capture similarity properties.

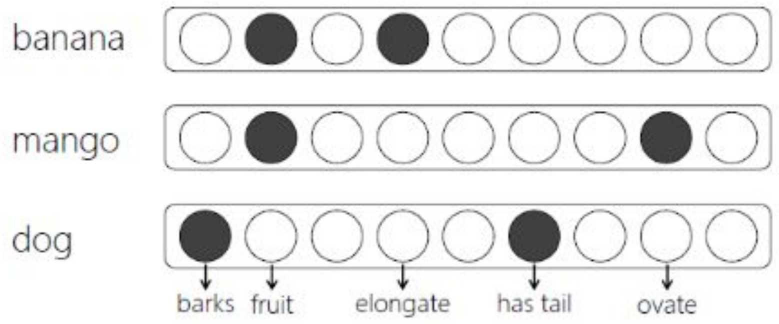


How to represent my words?

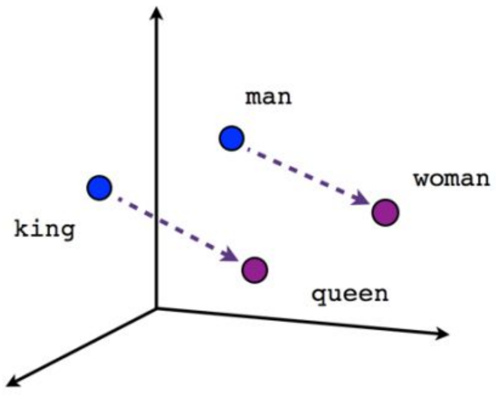
- **Local representations**
- Problems with this representation?
  - Sparsity
  - Vectors don't capture similarity properties.



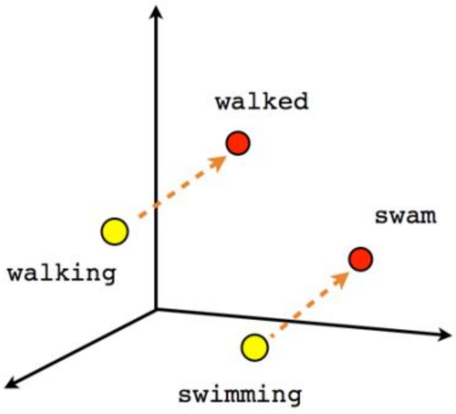
- **Distributed representations (embeddings)**
- Advantages of this representation?
  - More compact vectors
  - Capable of capturing similarities



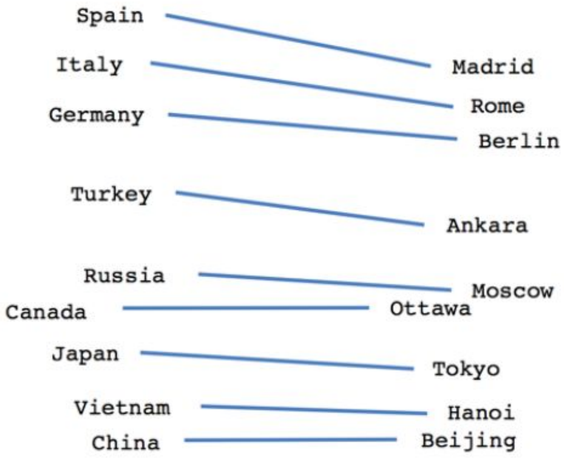
Embedding representations



Male-Female



Verb tense



Country-Capital

A very simple example

Text Classification with Keras

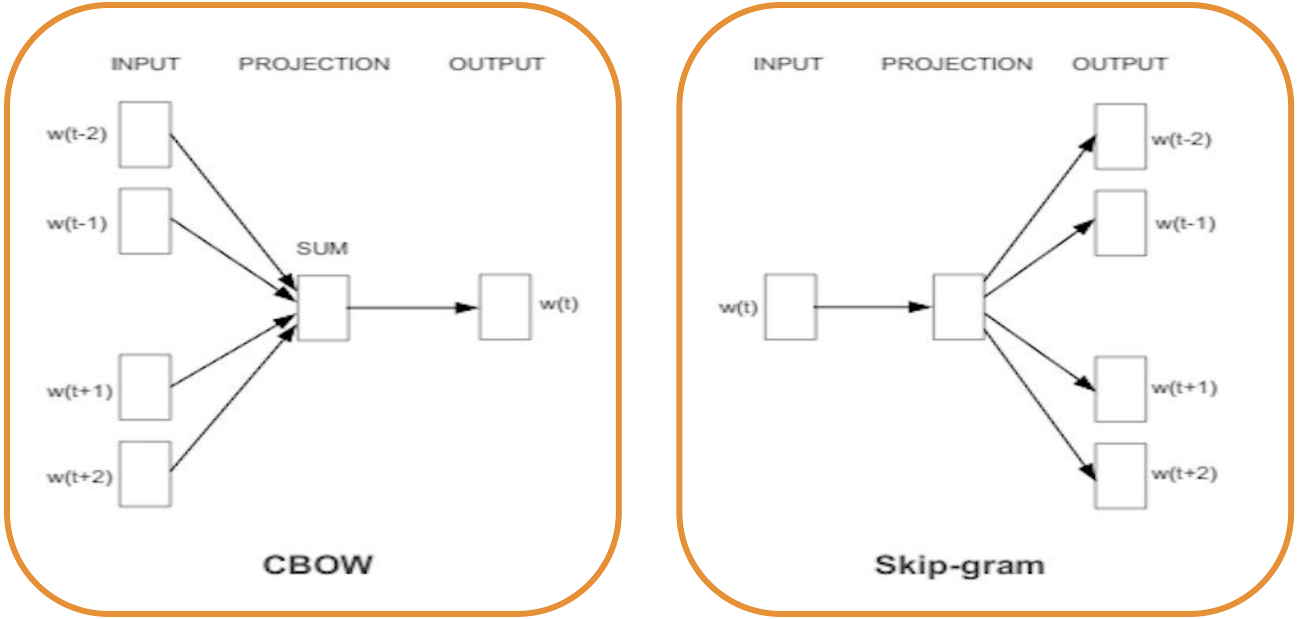
- http://localhost:8888/notebooks/Simple%20Text%20Classification.ipynb

Unsupervised ways to train embeddings

Word2Vec (2013)

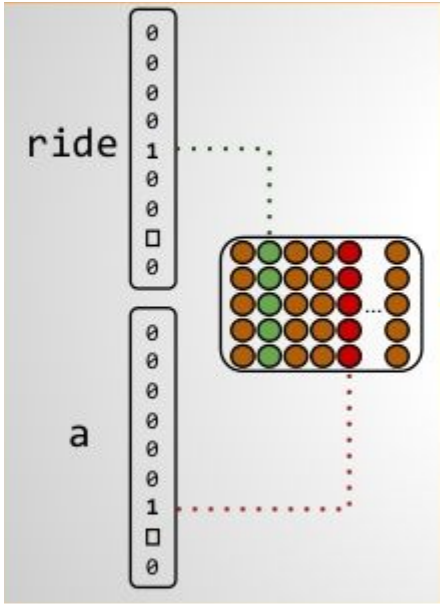
what I want to predict

NLP is simply awesome



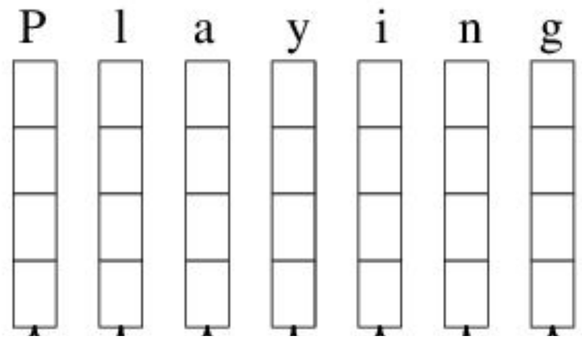
Word2Vec (2013)

- Final result is an embedding matrix that represents N words
- Is able to capture semantic relations between words
- Out of vocabulary (OOV) problem: How do I represent words that I haven't seen during training?

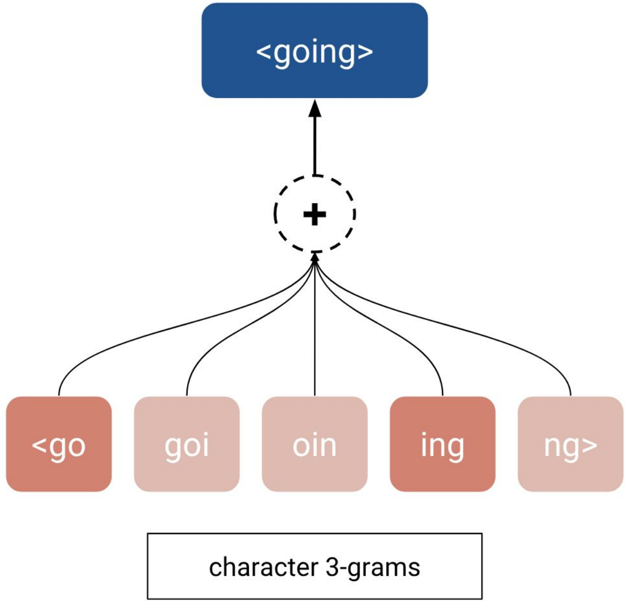


Other types of representations

- Character embeddings

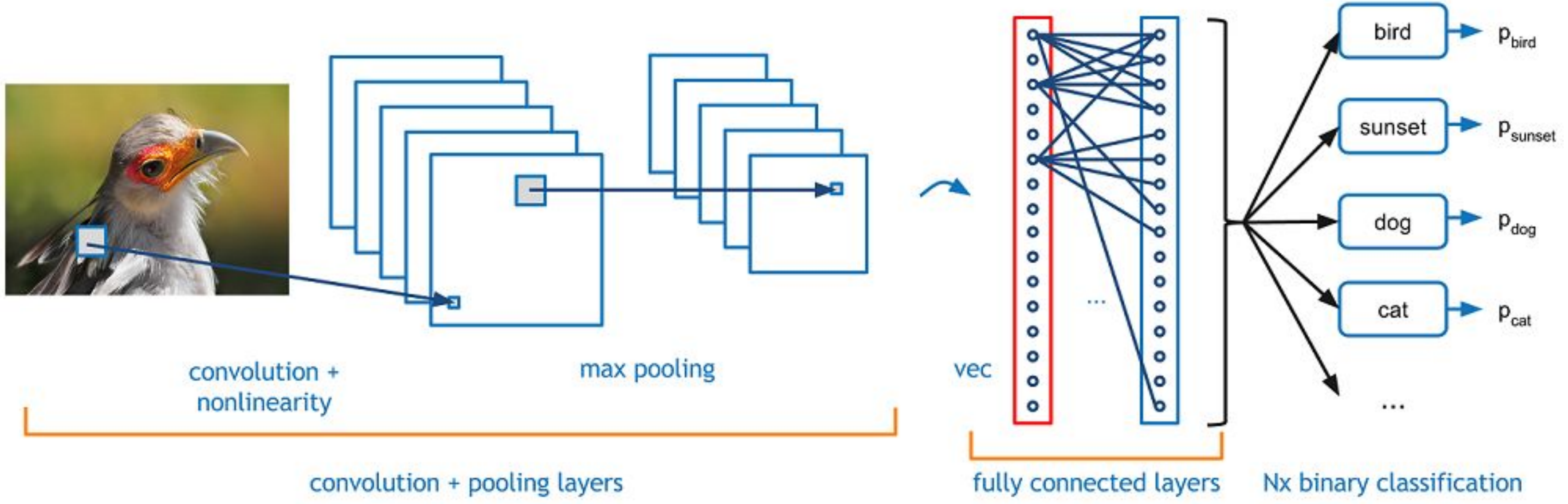


- Subword embeddings

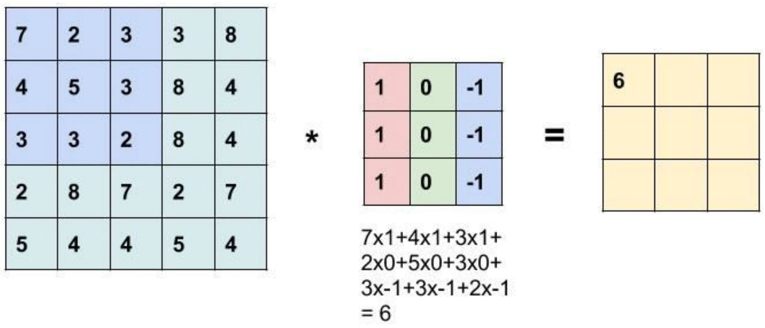


Advanced Neural Architectures

Convolutional Neural Networks

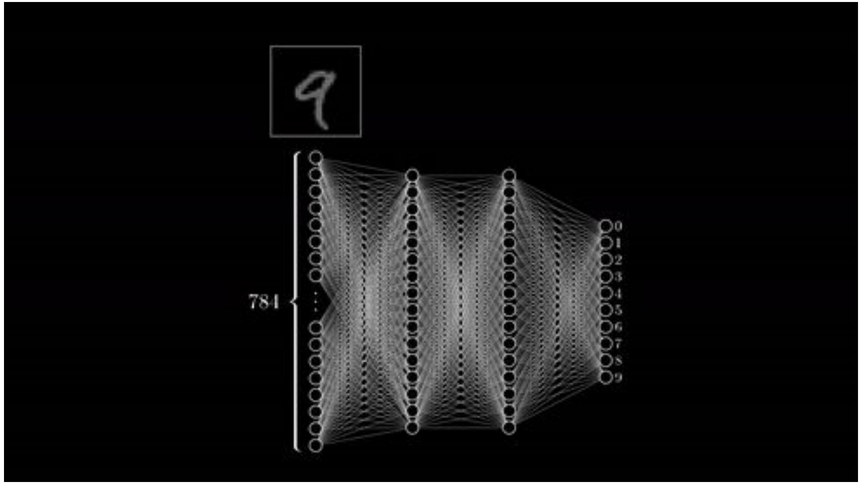


Convolutional Neural Networks



Convolution + Max Pooling

Convolutional Neural Networks



Hidden dimensions may identify specific patterns

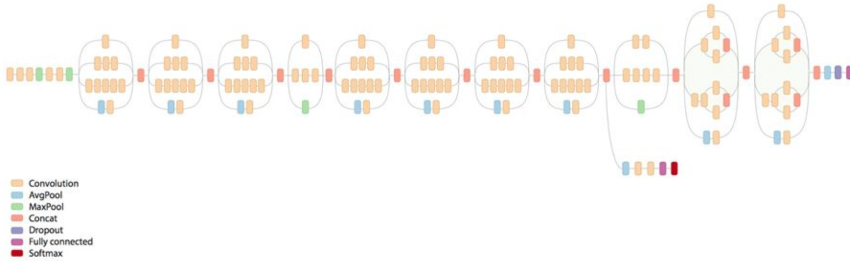
Convolutional Neural Networks

Well said Leo, well said

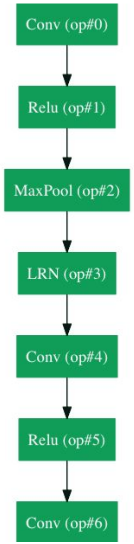
Convolutional Neural Networks

GoogLeNet (2015) was one of the first models that introduced the idea that CNN layers didn't always have to be stacked up sequentially.

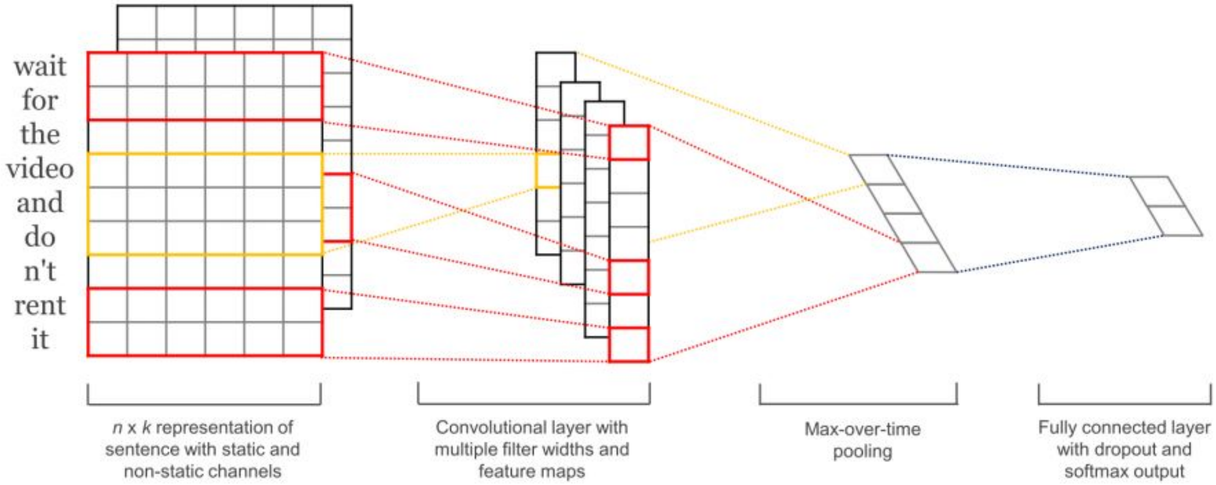
- 22 layer CNN



Another view of GoogLeNet's architecture.



Convolutional Neural Networks for NLP

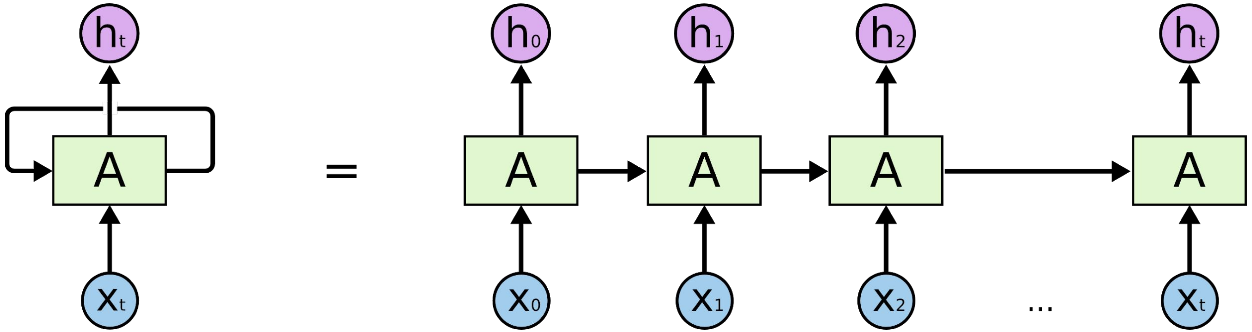


CNN architecture for text classification
Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification

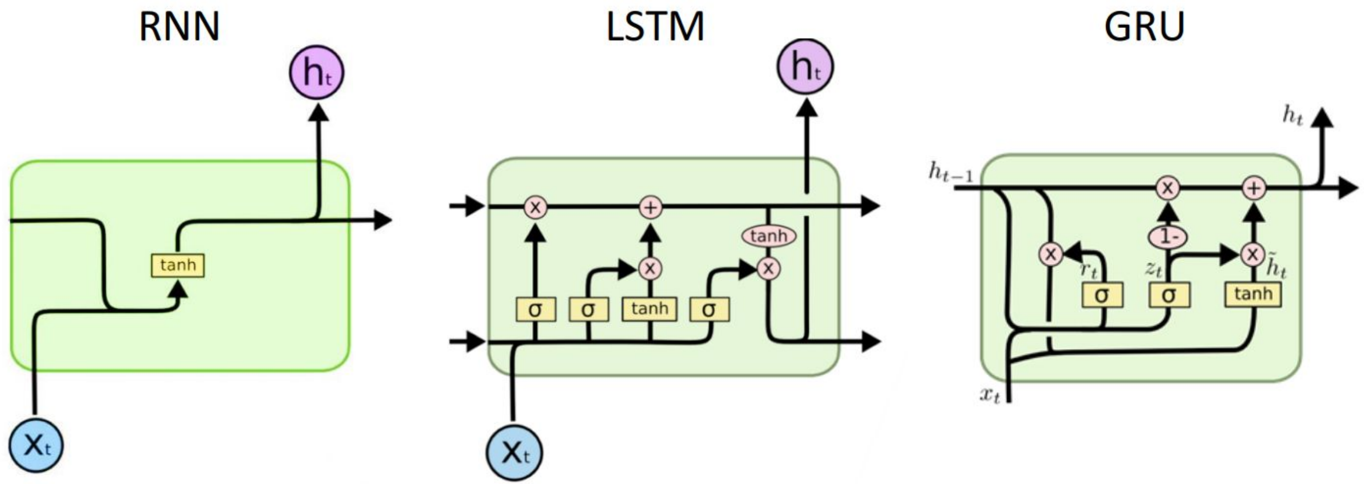
Convolutional Neural Networks

- Example - CNN for Text Classification.ipynb

Recurrent Neural Networks



Recurrent Neural Networks



Detailed comparison:
https://www.slideshare.net/YanKang/rnn-explore-71268007

Recurrent Neural Networks

- Example - RNN for Text Classification.ipynb

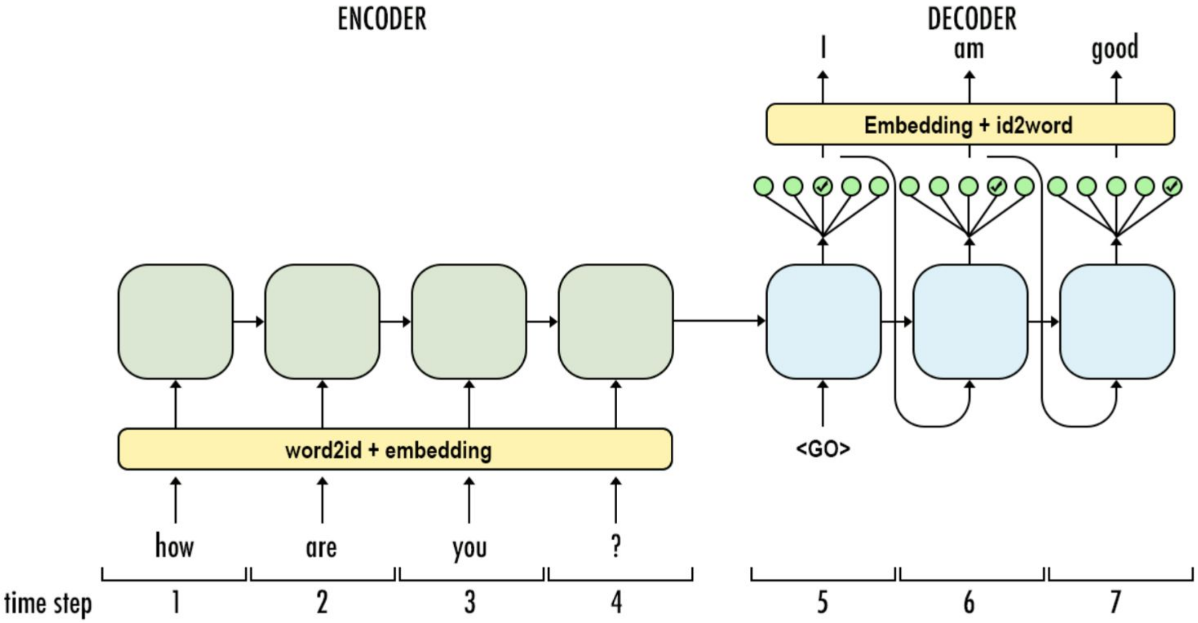
Language Modeling

Language modeling -- a "fill in the blank"-style next word prediction objective which allows models to learn generic sequence representations that generalize well to new tasks.

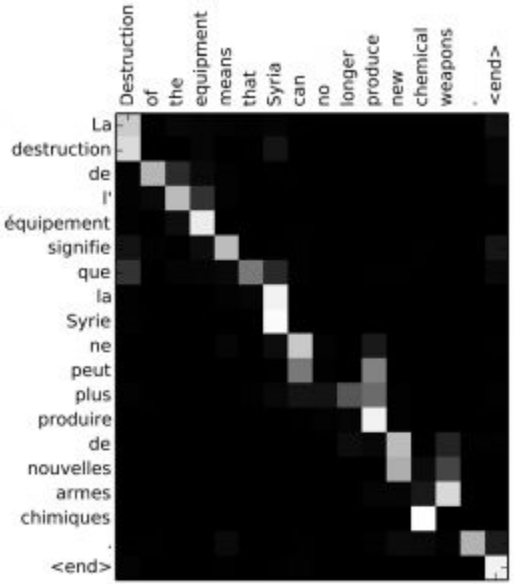
Language Modeling

- Example: Language Modeling.ipynb

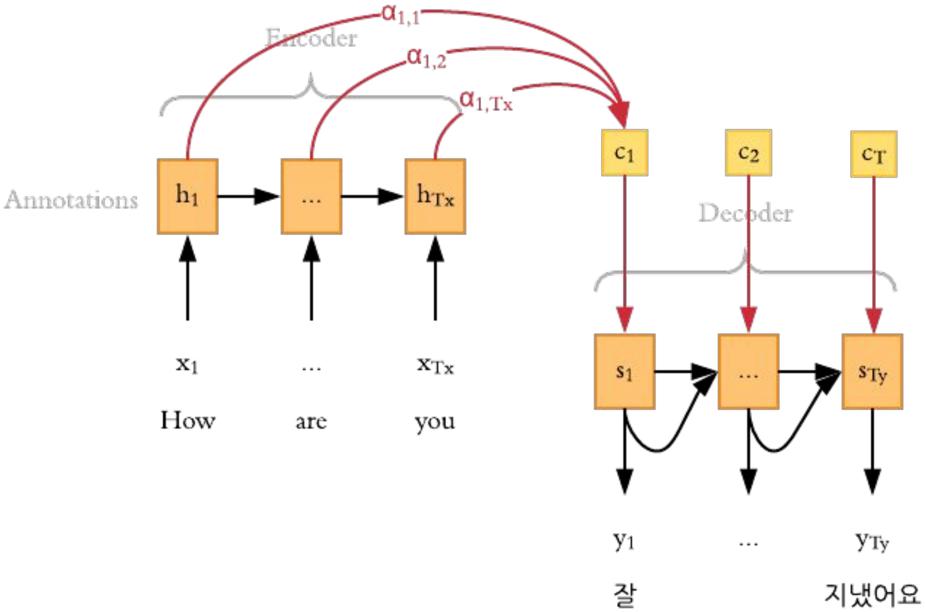
Seq2seq


ENCODER
DECODER
I
am
good
Embedding + id2word
<GO>
word2id + embedding
how
are
you
?
time step
1
2
3
4
5
6
7

Attention



Attention heatmap



Transformer

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com

Noam Shazeer* Google Brain noam@google.com

Niki Parmar* Google Research nikip@google.com

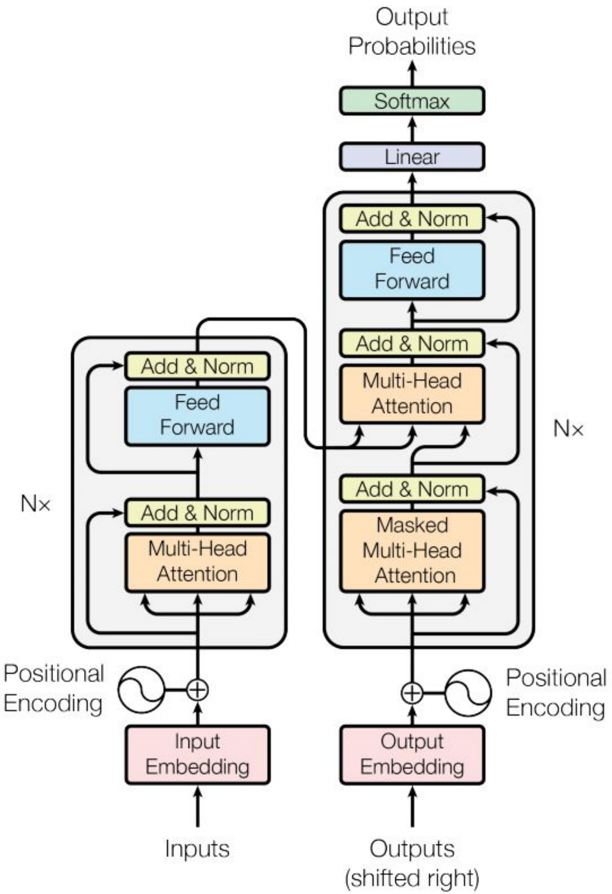
Jakob Uszkoreit* Google Research usz@google.com

Llion Jones* Google Research llion@google.com

Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

Łukasz Kaiser* Google Brain lukaszkaiser@google.com

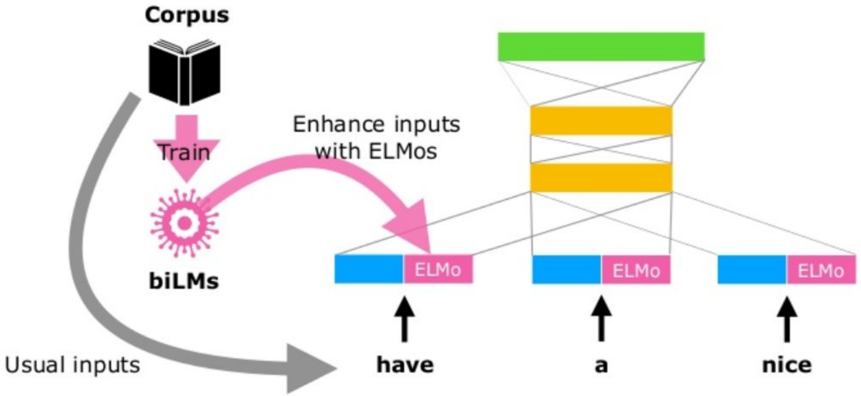
Illia Polosukhin* ‡ illia.polosukhin@gmail.com



Contextual Embeddings

ELMO

- The representations here differ from traditional word type embeddings in that each token is assigned a representation that is a function of the entire input sentence
- Vectors derived from a bidirectional LSTM that is trained with a coupled language model (LM) are used, and for that reason the authors of the paper called them **ELMo (Embeddings from Language Models) representations**

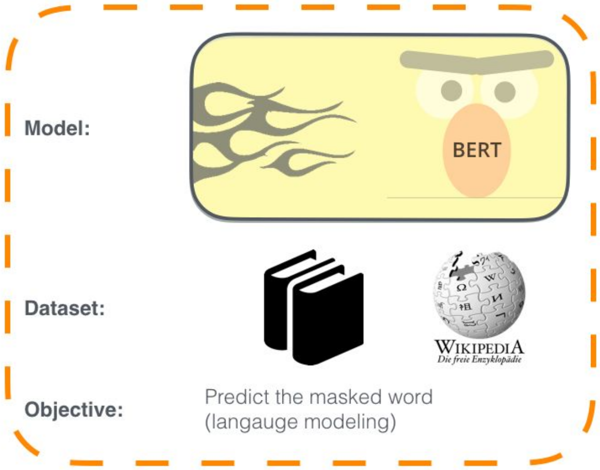


BERT

1 - Semi-supervised training on large amounts of text (books, wikipedia..etc).

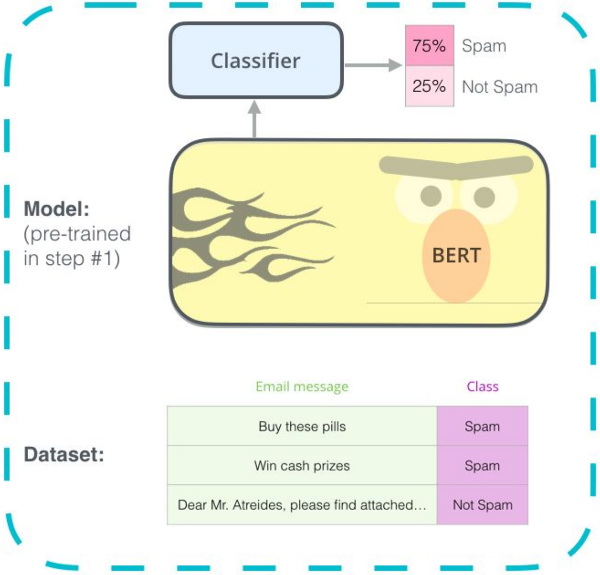
The model is trained on a certain task that enables it to grasp patterns in language. By the end of the training process, BERT has language-processing abilities capable of empowering many models we later need to build and train in a supervised way.

Semi-supervised Learning Step



2 - Supervised training on a specific task with a labeled dataset.

Supervised Learning Step



Practical Exercises

Additional Pointers

Additional Pointers

- https://www.coursera.org/learn/language-processing