An Introduction to NLP with BERT

16th Nov 2019

Milind K Thombre
Founding Engineer, OpenInterview
milind.thombre@openinterview.co.in
https://github.com/thombrem

$ whoami

milind-thombre-full-stack-developer

1995 -BE(Electronics- VIT)

1995-2016 (IT and S/W Product Dev industry in India and US as well as various entrepreneurial stints)

2016-2018 - ME (Comp- MIT) specializing in ML/Cloud

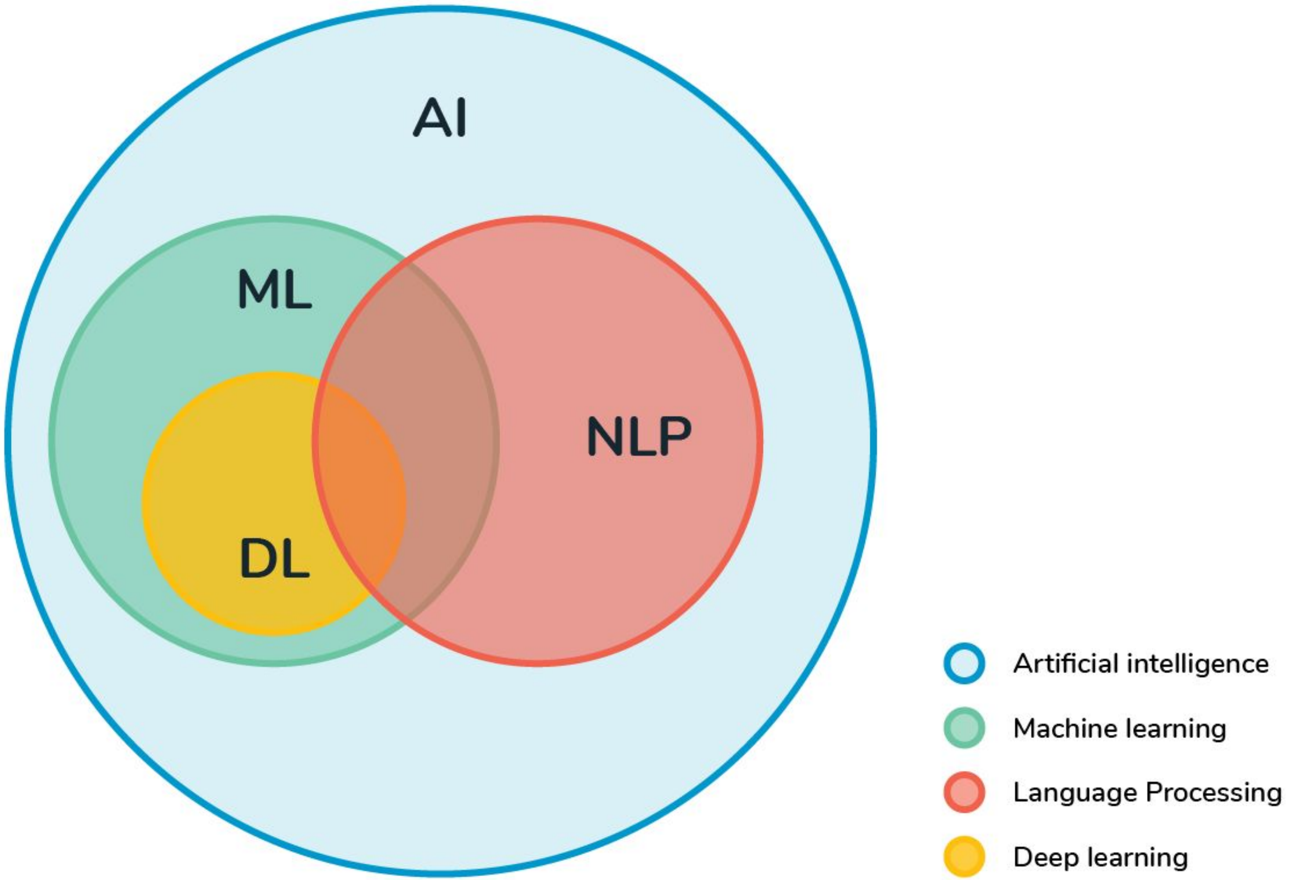
2019-Present - OpenInterview (Founding Engineer)

Current Technical Interests {NLP-Bots, Python,ML/AI, Cloud, SaaS}

$ finger OpenInterview

Mission: *Building Meritocracies with AI*

Product: “Video interview bot and code evaluation SaaS platform that automates talent acquisition for the technology industry”


AI
ML
DL
NLP
Artificial intelligence
Machine learning
Language Processing
Deep learning

NLP Knowledge Areas

More Deeper Application of NLP

Group 1	Group 2	Group 3
Cleanup, Tokenization	Information Retrieval and Extraction (IR)	Machine Translation
Stemming	Relationship Extraction	Automatic Summarization/ Paraphracing
Lemmatization	Named Entity Recognation (NER)	Natural Language Generation
Part of Speech Tagging	Sentiment Analysis/Sentance Boundary Dismbiguation	Reasoning over Knowledge Based
Query Expansion	World sense and Dismbiguation	Quation Answering System
Parsing	Text Similarity	Dialog System
Topic Segmentationand Recognation	Coreference Resolution	Image Captioning & other Multimodel Tasks
Morphological Degmentation (Word/Sentences)	Discourse Analysis	

Agenda

Section 1:

- Introductions
- Background and development of BERT
- ML/Deep Learning Refresher
- Popular Deep Learning Frameworks

Section 2:

- Current Benchmarks of various models for typical NLP tasks
- Major Tasks in NLP and their Problem Statements
- Transfer Learning and Ensemble Learning and why it is relevant here
- BERT’s anatomy explained

Agenda

Section 3:

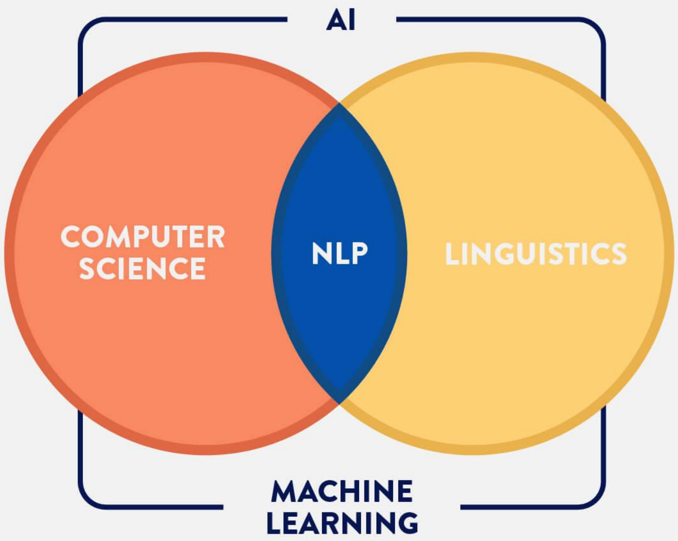
- Current Scientific Community in NLP (Who's Who)
- Some Applications, Products that use NLP
- Open Problems in NLP
- Measuring the success of NLP Applications
- Novel Product Ideas Brainstorming Session

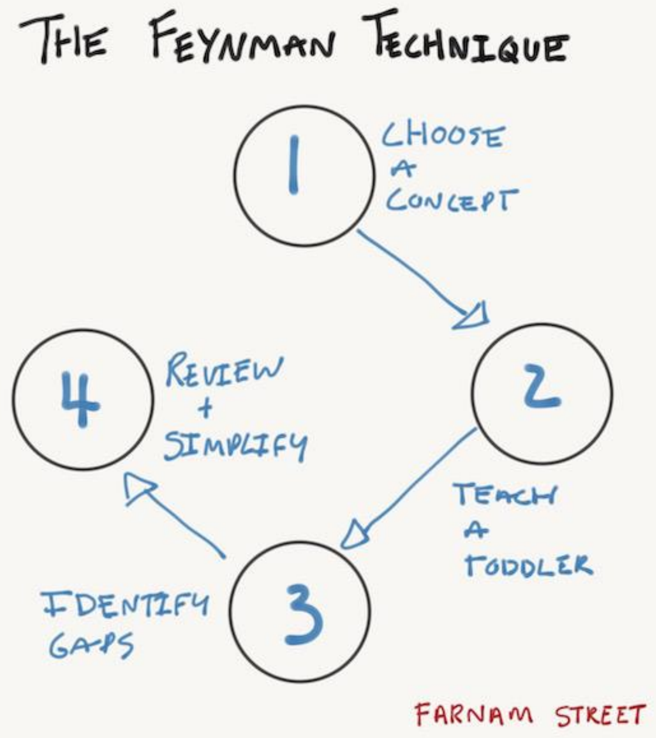
Extras:

- Fun Quizzes at the end of each section
- Brainstorming Session at the very end
- Keeping it Interactive (Q&A)

WHAT IS NATURAL LANGUAGE PROCESSING?

The interdisciplinary field of computer science and linguistics. NLP is the ability for computers to understand human language.




THE FEYNMAN TECHNIQUE
1
CHOOSE A CONCEPT
2
TEACH A TODDLER
3
IDENTIFY GAPS
4
REVIEW + SIMPLIFY
FARNAM STREET

History repeats itself for those who refuse to learn from it and change!

In order to be Truly Innovative, we must first learn what has already been achieved by the Human Race, so as not to reinvent the wheel!

-Yours truly

History of NLP and run up to BERT

- POS tagging
- Phrase structure rules -> Parse trees -> NLG
- Knowledge Graph (70+Billion entries, 2016)
- Rule-based Bots (syntactic rules)
- ML-based Dialog Systems
- Early Speech Recognition (Bell Labs - telephone dialler)
- Modern Deep Neural Network based ASR (Spectrograms, FFT, Phonemes)

https://youtu.be/fOvTtapxa9c (PBS)

Machine Learning Refresher

1. Classification (S)
2. Regression(S)
3. Clustering (unsupervised)

Definition:

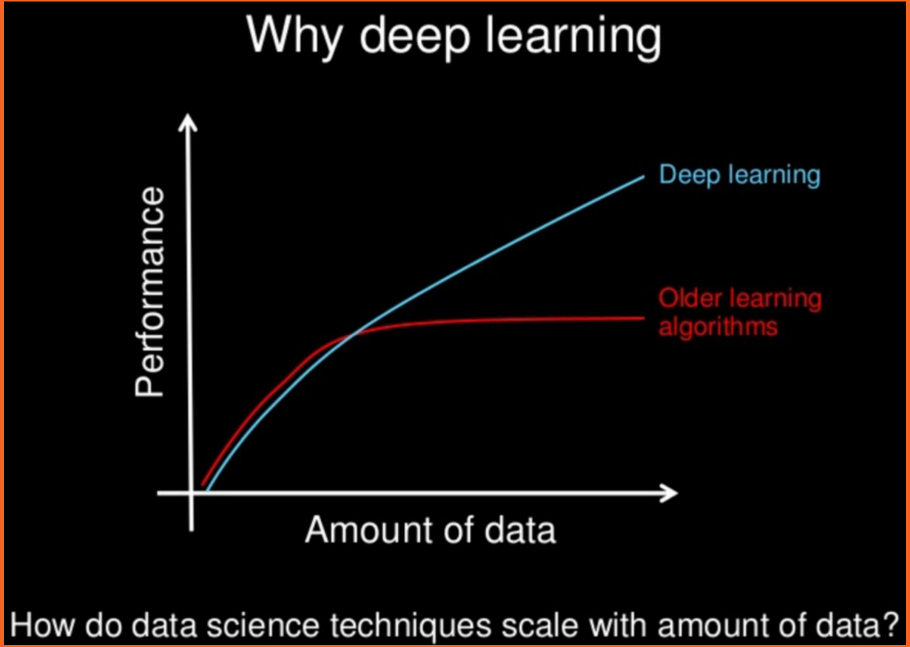
“A computer program is said to learn from experience *E* with respect to some class of tasks *T* and performance measure *P* if its performance at tasks in *T*, as measured by *P*, improves with experience *E* ”

ML Engineer’s Timesheet

1. Problem/Hypothesis definition and ML task identification (e.g. this is a classification problem)
2. Function Discovery or Mapping cause(s) to effects
3. Training Data Gathering (human validated data)
4. Feature Engineering by Application of Human’s Domain knowledge: Modeling of Features a.k.a variables aka.attributes
5. Dimensionality reduction (e.g. Principal Component Analysis)
6. Algorithm Selection and choosing tuning parameters (e.g. choosing k for a clustering problem)
7. Model Training
8. Prediction for unlabelled datasets
9. Hypothesis Testing

Why Bother with DL?

Performance improves dramatically with increasing scale of training data!



Digression - The Nested Hierarchy of Concepts in our Universe

1. Shape Square and Circle
2. Cat vs Dog Classifier (Automatically Identify the features that are significant for classification
3. Evolution (Amoeba -> Human)
4. Sex determination in a developing embryo (Vishnu's Dashavataras, Matsya, Kurma, varaha etc.)
5. Fusion in the Stars (Hydrogen -> Helium-> Supernovae -> Gold (Hiranyagarbha - literally means "golden womb" ~ Rig Veda)

Connecting the dots: Why does DL work so well?

1. Deep Learning and the Laws of Physics -Our Universe, transformations and the Power of 4!
2. Causal Hierarchy-Each causal layer contains progressively more data

Deep Learning Definition

“Deep learning is a particular **kind of machine learning** that achieves great power and flexibility by learning to represent the world as **nested hierarchy of concepts**, with each concept defined in relation to **simpler concepts**, and more abstract representations computed in terms of less abstract ones.”

Deep Learning Refresher

- Loss Functions <<graphic>>
- Gradient Descent
- Learning Rate
- Hyperparameter Tuning
- Regularization
- Optimization
- Multi-Class Classification with Softmax

Hyperparameters

- **Hyperparameters** are settings that can be tuned to control the behavior of a **machine learning** algorithm
- They vary by ML model (CNN/RNN)
- Manually set but
- Algorithms such as **Grid Search and Random Search** can infer Hyperparameters automatically

Common Hyperparameters

- **Learning rate – α**
- Momentum – β
- Adam's hyperparameter – β1, β2, ε
- **Number of hidden layers**
- **Number of hidden units for different layers**
- Learning rate decay
- Mini-batch size

Hyperparameter Tuning

- Parameters versus Hyperparameters (know the difference!)
- Bias vs Variance
- Regularization
  - L2 regularization
  - Dropout
  - Data Augmentation
  - Early Stopping
- Normalizing (-mean /variance)
- Normalizing the inputs makes the cost function symmetric making it easier for Gradient Descent to find global minima quickly

- Weight Initialization for Deep Neural Networks to speed up training
- Learning rate Optimization: Gradient descent, Momentum, RMSprop, Adam
- Tools for Automated Hyperparameter tuning: List

Convolutional vs Recurrent NNs

CNN

- CNN is a feed forward neural network
- 4 layers : Convolution layer, ReLU layer, Pooling and Fully Connected Layer
- Every layer has its own functionality and performs feature extractions and finds out hidden patterns
- Typical use cases: Image recognition and object classification
- Link

RNN

- CNN considers only the current input while RNN considers the current input and also the previously received inputs
- It can memorize previous inputs due to its internal memory aka LSTM Long Short Term Memory)
- 4 Types or RNN's : One to One, One to Many, Many to One and Many to Many.
- RNN can handle sequential data while CNN cannot
- Typical use cases: In RNN, the previous states is fed as input to the current state of the network. RNN can be used in NLP, Time Series Prediction, Machine Translation, etc.
- Link

Popular Deep Learning Frameworks

Tensorflow 2.0- Google - Adopters (AirBnB,Intel, Twitter) - Language:Python, Active Community support, Works on static computation graph, Ships with Keras (Simplify)

Caffe - Old, Languages: C, C++, Python, MATLAB, and CLI, Limitation: No support for granular neural network layers

PyTorch - Facebook, OpenSource, Tensorflow Competitor, Language: Python, Dynamically updated graph

Microsoft Cognitive Toolkit (Previously CNTK)- Languages: Python, C++, and CLI, Higher performance and scalability while operating on multiple machines.

Popular Deep Learning Frameworks

Sonnet by DeepMind- Built on top of Tensorflow,

MXNet - Apache project,Languages: C ++, Python, R, Julia, JavaScript, Scala, Go, and Perl, very effectively parallel on multiple GPUs and many machines

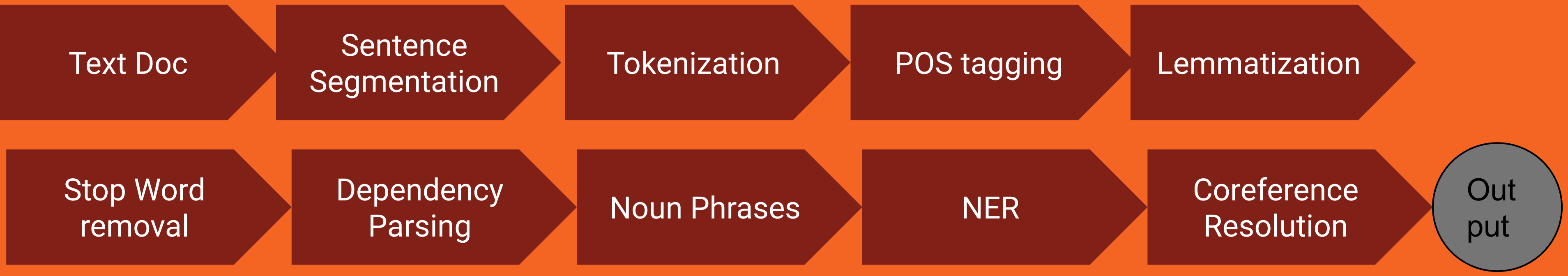
Chainer-Was the leader in dynamic computation graphs that allowed inputs of varying lengths (Typical need of NLP), Language: Python(NumPy, CuPy), Fastest Python based framework, better GPU & GPU data center performance than TensorFlow.

Popular Deep Learning Frameworks

DL4J (short for Deep Learning for Java), supported by Hadoop and Spark Architectures, Android-edge computing etc.

Fun Quiz # 1

Traditional NLP Pipeline



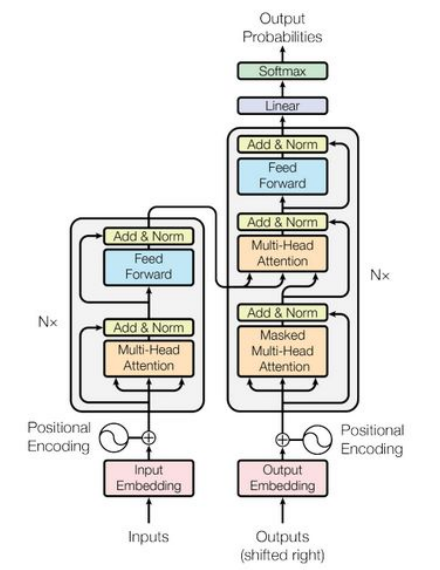
Typical NLP (Functional) Tasks

Text Classification	• Representation: bag of words • Goal : predict tags, categories, sentiment • Application: filtering spam emails, classifying documents based on dominant content
Word Sequence	• Representation: sequences (preserves word order) • Goal: language modeling - predict next/previous word(s), text generation • Application: translation, chatbots, sequence tagging (predict POS tags for each word in sequence), named entity recognition
Text Meaning	• Representation: word vectors, the mapping of words to vectors (*n*-dimensional numeric vectors) aka embeddings • Goal: how do we represent meaning? • Application: finding similar words (similar vectors), sentence embeddings (as opposed to word embeddings), topic modeling, search, question answering

Typical NLP (Functional) Tasks

Sequence to Sequence	• Many tasks in NLP can be framed as such • Examples are machine translation, summarization, simplification, Q&A systems • Such systems are characterized by encoders and decoders, which work in complement to find a hidden representation of text, and to use that hidden representation
Dialog Systems	• 2 main categories of dialog systems, categorized by their scope of use • Goal-oriented dialog systems focus on being useful in a particular, restricted domain; more precision, less generalizable • Conversational dialog systems are concerned with being helpful or entertaining in a much more general context; less precision, more generalization

Transformers (Core Building Block of BERT)



- Attention is all you need - Ashish Vaswani@Google Brain et al
- Sequence 2 Sequence models
- Encoder-Decoder stack
- Attention
  - Scaled Dot-Product Attention
  - Multi-Head Attention
- Applications of Attention in our Model
- Position-wise Feed-Forward Networks
- Embeddings and Softmax
- Positional Encoding
- Why Self Attention (vs recurrent and Convolutional attn layers)
- Results: BLEU Score for machine translation
- Conclusion: F1 Score achieved is 88.3 to 93.3 (depending on training dataset and other params)

Transformers (Core Building Block of BERT)

- RNN based Models: Encoder-Decoder
- Limitation: Unable to deal with long range dependencies
- Here, “transduction” means the conversion of input sequences into output sequences.

- The idea behind Transformer is to handle the dependencies between input and output with attention and recurrence completely.

Algorithm Time Complexities

Layer Type	Complexity per layer	Sequential Operations	Max. Path Length
Self-Attention	O(n^2 · d)	O(1)	O(1)
Recurrent	O(n · d^2)	O(n)	O(n)
Convolutional	O(k · n · d^2)	O(1)	$O(\log_k(n))$
Self-Attention (restricted)	O(r · n · d)	O(1)	O(n/r)

BERT - Bidirectional Encoder Representations from Transformers

- BERT is a powerful deep-learning model developed by Google based on the transformer architecture. BERT has shown state-of-the-art results and a number of the most common NLP tasks and can be used as a starting point for building NLP models in many domains
- BERT abstracts away some of the most complicated and time-consuming aspects of building an NLP and evidence has shown that BERT can be used to reduce the amount of data required to train a high performing model by over 90%.
- BERT also reduces production complexity, development time, and increases accuracy.

BERT - How does it work?

- **BERT** is a method of pre-training **language** representations, meaning that we train a general-purpose "**language** understanding" **model** on a large text corpus (like Wikipedia), and then use that **model** for downstream NLP tasks that we care about (like question answering).

BERT - Models

- There are **two** models introduced.
- **BERT** base – 12 layers (transformer blocks), 12 attention heads, and 110 million **parameters**.
- **BERT** Large – 24 layers, 16 attention heads and, 340 million **parameters**.
- **ALBERT** A Light BERT now outperforming **BERT**

BERT’s Modus Operandi

Pre-training:

- Masked Language Modeling (MLM)
- Next Sentence Prediction (NSP)
- BERT’s Model Architecture: BERT base and BERT Large

Fine Tuning:
Training the model with Your DATA!

BERT-as-a-service Installation

pip install -U bert-serving-server bert-serving-client

Current Benchmarks of various NLP models

SQuAD - Stanford Question Answering Dataset

GLUE - General Language Understanding Evaluation used for Natural Language Understanding tasks

BLEU - Bilingual Evaluation understudy, Used by BERT, typically for translation tasks

DecaNLP - **Natural Language Decathlon,** Spans 10 NLP tasks.

State of Art NLP Model Metrics

The **F** measure (F1 **score** or **F score**) is a measure of a test's accuracy and is defined as the weighted harmonic **mean** of the **precision** and **recall** of the test
Diagram

Which models are current Leaders? **This changes EVERYDAY!**

Link to SQuAD Leaderboard

DecaNLP **Leaderboard**

Transfer Learning

Transfer learning is a research problem in machine **learning** that focuses on storing knowledge gained while solving one problem and applying it to a different but related problem.

For example, knowledge gained while **learning** to recognize cars could apply when trying to recognize trucks.

Ensemble Learning

Ensemble learning helps improve **machine learning** results by combining **several models.**

Ensemble methods are meta-algorithms that combine several **machine learning** techniques into one predictive model in order to **decrease variance** (bagging), **bias** (boosting), or **improve predictions** (stacking)

Fun Quiz #2

Order in Chaos vs Unpredictably Random

The Galton Board (1876)

- Heights of people in the population
- Predicting Likelihood of Outcomes
- Regression to the Mean
- Statistical Inference

Youtube: https://youtu.be/Kq7e6cj2nDw

Order in Chaos!

Chaos is based on the idea that minute differences in your starting condition can magnify into large results.

Every step along the way can be perfectly predicted if you have enough precision, but the longer it runs the more any imprecision magnifies.

E.g. “Outliers” - Bringing up children

NLP: Human Beings are predictable, language has structure

NLP Who’s Who (present)

Yoav Artzi | Cornell | BERTScore, Robotics , NLP etc

Emily M. Bender |U of Washington| Multilingual Grammar and Translation

Yoav Goldberg | Bar Ilan University |Neural Network based NLP

Matthew Honnibal | Founder@Explosion AI |Author of spaCy

Ines Montani | Founder @ Explosion AI | Maker of spaCy

Jeremy Howard | Founder @ fast.ai, Faculty @ University of San Francisco| AI/NLP MOOCs

Christopher Manning | Director @ Stanford AI Lab, CS & Linguistics Professor @ Stanford

NLP Who's Who (present)

Sebastian Ruder | Research Scientist @ DeepMind | Unsupervised Cross-lingual Representation Learning

Vered Shwartz | Postdoc @ Allen AI and UW NLP | lexical semantic relations

Richard Socher | Chief Scientist @ Salesforce | deep learning, natural language processing and computer vision

Rachael Tatman |Data Scientist @ Kaggle, Linguistics PhD |

Rachel Thomas | Director @ USF Center for Applied Data Ethics & Founder @ fast.ai | Ethics, AI accessibility, bias in machine learning

Products that use NLP (unordered)

- NLTK (OpenSource)
- Spacy (free)
- SnatchBot (codeless design)
- Slack
- RocketChat
- MSFT linguistics API, Text Analytics API (Azure Microservices)
- Google Natural Language API, and other services on GCP
- Watson NLU
- Stanford CoreNLP
- Amazon Comprehend

Open Problems in NLP

Hard (Work still in Progress)

Text Summarization - to take input as text document(s) and try to condense them into a summary.

Machine dialog system (detecting missing info in what is said etc)

Open Problems in NLP

Intermediate (making good progress)

- **Sentiment analysis-**
- **Coreference resolution** -
- **Word sense disambiguation**
- **Parsing** - the basic problem of parsing sentences.
- **Machine Translation** - translating sentences from one language to another, best example would be Google translate.
- **Information Translation** - to take a text as input and represent it in a structured form like a database entries.

Fun Quiz #3

Brainstorming session 4+1 Rules

1. **No judgements.** This is the first rule of creativity in general.
2. **Think freely.** As I said before, no matter how crazy it is; while brainstorming, ideas are neither silly nor impossible.
3. **Big numbers.** The more ideas, the better.
4. **Many heads** are better than one

Brainstorming session Process

1. Sample 3 random ideas
2. Create a large List of Ideas by each Group
3. Discussion, Criticism(identify gaps) and Refinement (plug holes)
4. Literature survey, was it invented already? (Google, patents etc)

Q&A

Thank You!

Linkedin: https://linkedin.com/in/milindthombre

Github: https://github.com/thombrem

Email ID for Applause! : milind.thombre@openinterview.co.in

Twitter: @Interview.Open

Linkedin : OpenInterview.in

Youtube: OpenInterview

Facebook: OpenInterview

Backup[Back]

Image + Algorithms

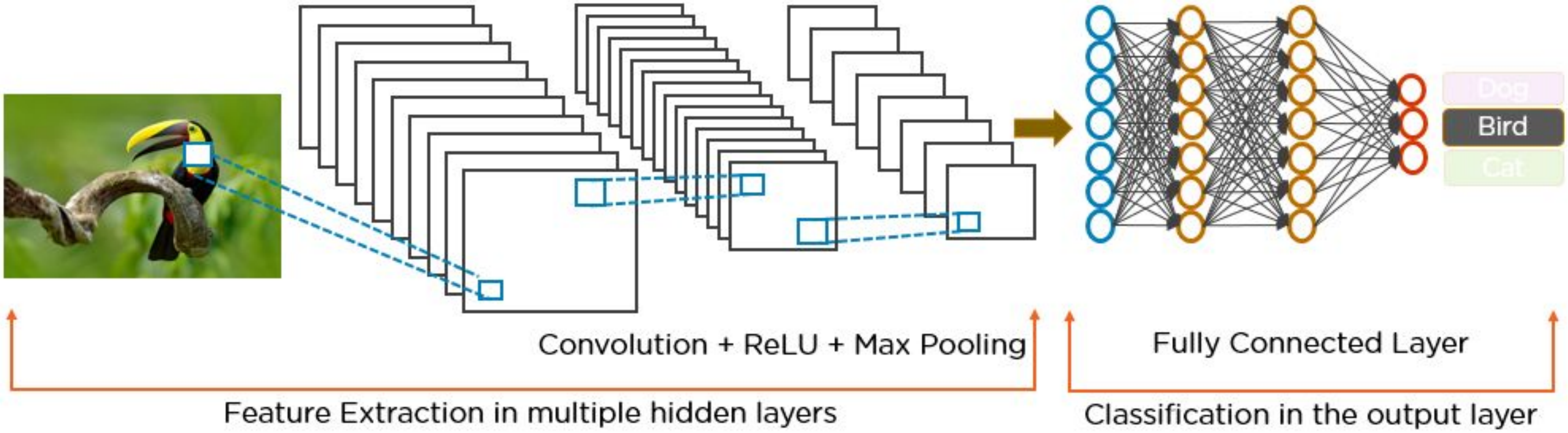
- SageMaker,
- Comet.ml,
- Weights&Biases (OpenAI),
- DeepCognition,
- AzureML,
- Cloud ML

Back

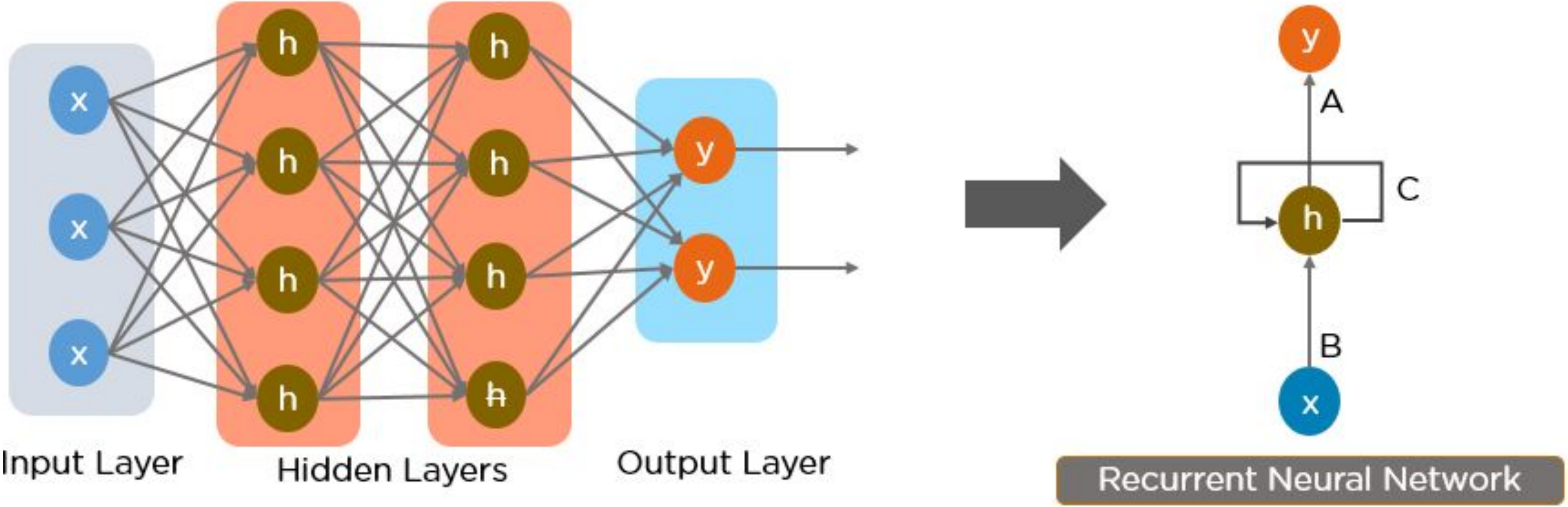
More Deeper Application of NLP

Group 1	Group 2	Group 3
Cleanup, Tokenization	Information Retrieval and Extraction (IR)	Machine Translation
Stemming	Relationship Extraction	Automatic Summarization/ Paraphracing
Lemmatization	Named Entity Recognation (NER)	Natural Language Generation
Part of Speech Tagging	Sentiment Analysis/Sentance Boundary Dismbiguation	Reasoning over Knowledge Based
Query Expansion	World sense and Dismbiguation	Quation Answering System
Parsing	Text Similarity	Dialog System
Topic Segmentationand Recognation	Coreference Resolution	Image Captioning & other Multimodel Tasks
Morphological Degmentation (Word/Sentences)	Discourse Analysis	

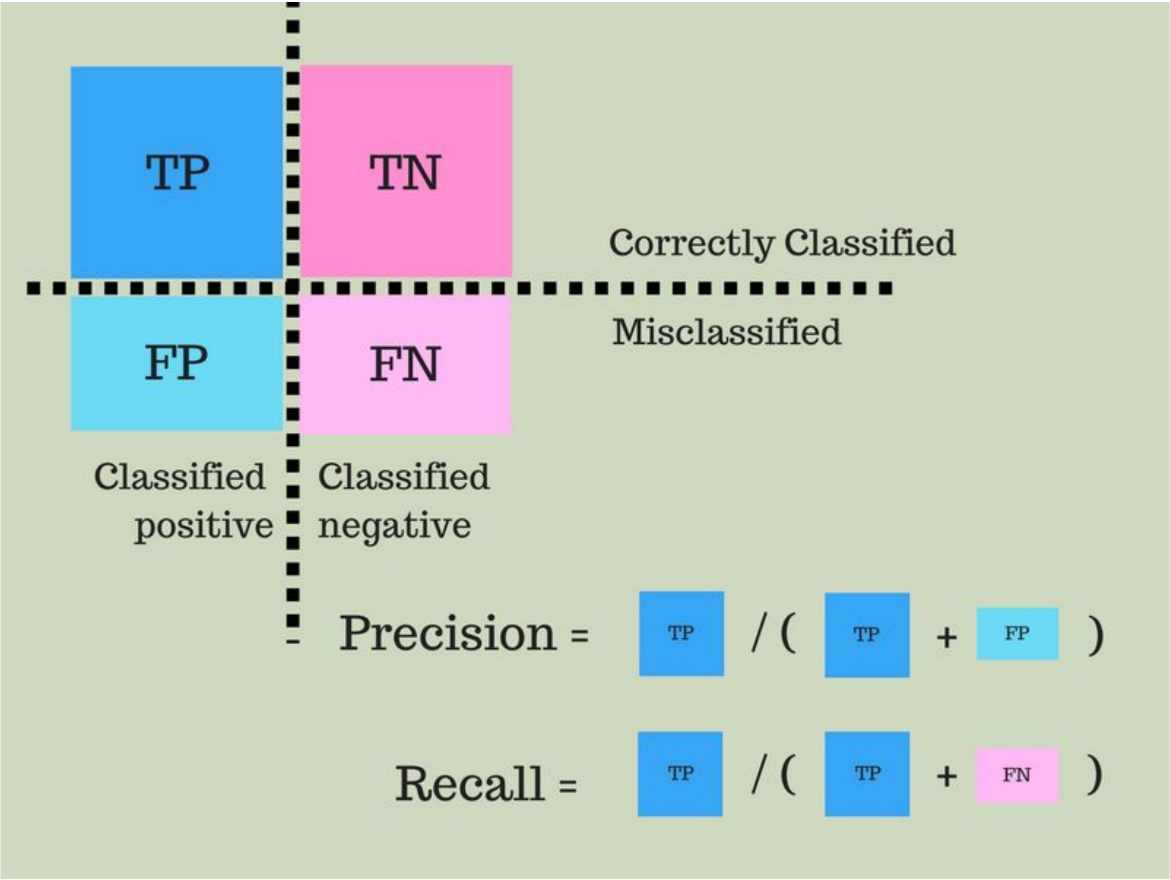
Back


Bird
Convolution + ReLU + Max Pooling
Fully Connected Layer
Feature Extraction in multiple hidden layers
Classification in the output layer

[Back](#)


x
x
x
h
h
h
h
h
h
h
h
y
y
Input Layer
Hidden Layers
Output Layer
y
A
C
h
B
x
Recurrent Neural Network

Back


TP
TN
Correctly Classified
FP
FN
Misclassified
Classified positive
Classified negative
Precision = TP / (TP + FP)
Recall = TP / (TP + FN)

[Back](#)

Fun Quiz 1