About Myself

- **Vikas Kumar**
  Consultant @ Deloitte USI
  Analytics & Cognitive, Applied AI Team
  (since April 2019)

Background

- Consultant @EXL
- Graduation from IIT Kharagpur (*2017*)

- Working on NLP and Computer Vision problems
- Active Kaggler since 2017 (3 Silver medals , 1 Bronze)

Linkedin: linkedin.com/in/vikashiitkgp/   Kaggle: kaggle.com/vikas15

kaggle

Competitions Expert

Current Rank	Highest Rank
1976	1499
of 112,358	

0	3	1

| 2018 Data Science Bowl | 172nd |
| a year ago·Top 5% | of 3634 |

| Quora Insincere Questions ... | 173rd |
| 4 months ago·Top 5% | of 4037 |

| TalkingData AdTracking Fr... | 193rd |
| a year ago·Top 5% | of 3951 |

- Know some Analytics already?
- Familiar with Natural Language Processing?
- Expertise in deep learning?

Machine Translation
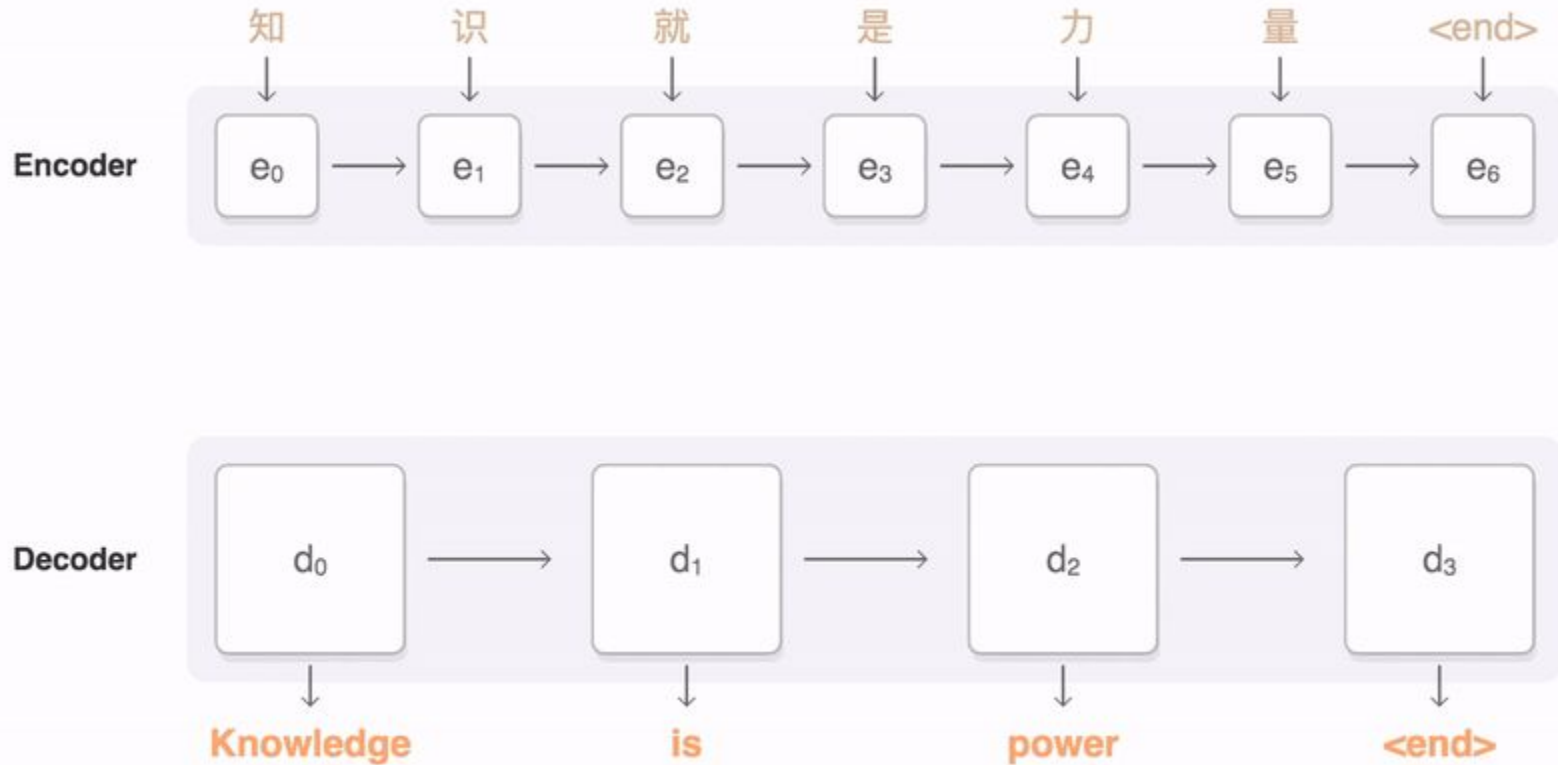
Name Entity Recognition

"There was nothing about this storm that was as expected," said **Jeff Masters**, a meteorologist and founder of **Weather Underground**. "**Irma** could have been so much worse. If it had traveled 20 miles north of the coast of **Cuba**, you'd have been looking at a (Category) 5 instead of a (Category) 3."

Person   **Organization**   **Location**

Question Answer System

The Normans (Norman: Nourmands; French: Normands; Latin: Normanni) were the people who in the 10th and 11th centuries gave their name to Normandy, a region in **France**. They were descended from Norse ("Norman" comes from "Norseman") raiders and pirates from Denmark, **Iceland** and Norway who, under their leader Rollo, agreed to swear fealty to King Charles III of West Francia. Through generations of assimilation and mixing with the native Frankish and Roman-Gaulish populations, their descendants would gradually merge with the Carolingian-based cultures of West Francia. The distinct cultural and ethnic identity of the Normans emerged initially in the first half of the 10th century, and it continued to evolve over the succeeding centuries.

When were the Normans in Normandy?
- *Ground Truth Answers: 10th and 11th centuries*
  Prediction: **10th and 11th centuries**
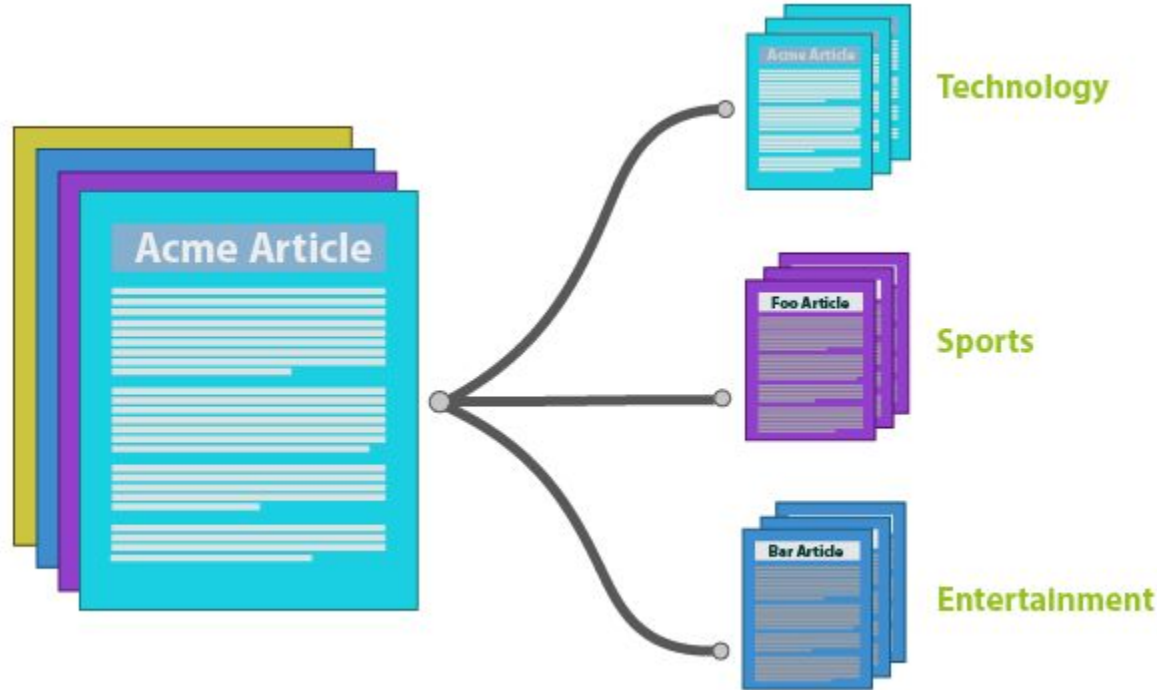
In what country is Normandy located?
- *Ground Truth Answers: France*
  Prediction: **France**

Reference

Text Classification

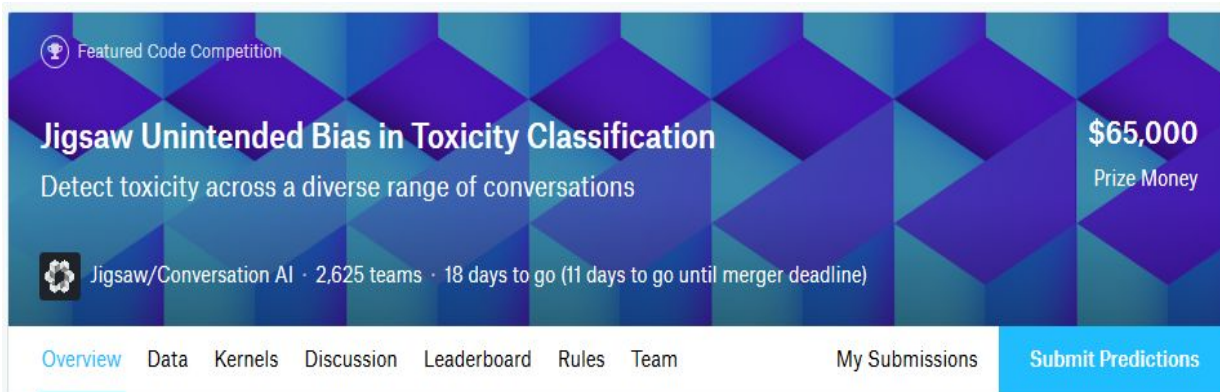# Text Classification Example

Kaggle Text Classification Challenge

Kaggle Toxic Classification Challenge

Competition Link

Problem Statement :-
- To detect toxic comments and minimize unintended model bias
- Build toxicity models that operate fairly across a diverse range of conversations

Last day of submission - 27 June, 2019

Sample Dataset

comment_text	target
And, as an American, I am happy to tell you freeloaders to go pound sand.	0
And they are receiving all the perks of being Canadian without paying into our taxes.	0
I'm embarrassed to be called an American with this buffoon leading the country.	1
Just remember, men wore jocks long before they wore helmets. Priorities!	0
good points mister gay, our mayor spendowitz better get on the job eh.	1
If motley was trolling then i could have a problem with his posts and anonymity. As i see it he doesnt troll and see no reason to know his backgroud information. Paul didnt you try and link with me on linkedin. If i wanted people to dig up personal information about me i would just give it to them. Anon	0
The royal parasites are back!	1
and you have to say that about the liberals too idiot, more voted against justin than voted for him! give it a rest, this bs is meaningless	1
All form of hate need to be eliminated racism, homophobia, Islamophobia, Anti-Semitism, discrimination against Transgenders. We must get behind Justin, Obama, Hillary and Premier Wynne and fight Trump and this racist Quebec law!	0
Dont do business with Wells Fargo,period, divest.	0
THAT IS A STUPID COMMENT,COULD YOU HAVE DONE BETTER.	1

What top kagglers are using for Text classification ?

# Baseline Models: Text Classification

2012	Bag of words (tf-idf)
2013	LSTM (word Embeddings - word2vec)
2015	LSTM (word Embeddings + attention)
2018-19	Pre-trained Language Model (ULMFiT, BERT) (Transfer Learning in NLP)

Traditional Approach : TF-IDF

2012	Bag of words (tf-idf)
2013	LSTM (word Embeddings - word2vec)
2015	LSTM (word Embeddings + attention)
2018-19	Pre-trained Language Model (ULMFit, BERT) (Transfer Learning in NLP)

Traditional Approach : TF-IDF

train_X

'This is good and awesome',

'This is bad'

Pos

Neg

Fit

TFIDFVectorizer
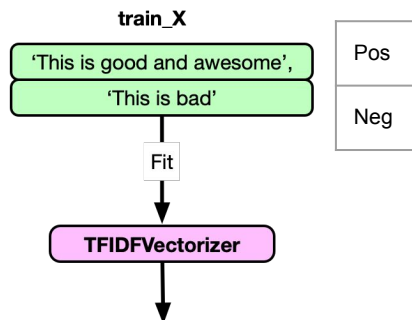
TF-IDF = Term Frequency x Inverse Document Frequency

TF(t) = (Number of times term t appears in a document) / (Total number of terms in the document)

IDF(t) = log (Total number of documents / Number of documents with term t in it)

Term Frequency
No of time word appears/No of total terms in Document

This	is	good	bad	awesome	and
1/5	1/5	1/5	0	1/5	1/5
1/3	1/3	0	1/3	0	0

Inverse Document Frequency
-log(ratio of documents that include the word)

This	is	good	bad	awesome	and
log(2/2)	log(2/2)	log(2/1)	log(2/1)	log(2/1)	log(2/1)

X

Features

This	is	good	bad	awesome	and
0	0	1/5*log(2/1)	0	1/5*log(2/1)	1/5*log(2/1)
0	0	0	1/3*log(2/1)	0	0

Pos

Neg

Is sequence of words in sentence important?

He only could solve that problem. ⟶ Positive

He could solve only that problem. ⟶ Negative

GROFERS Google LOGICAi kaggle CODING NINJAS
www.codingninjas.in

RNN / LSTM (word Embeddings - word2vec)

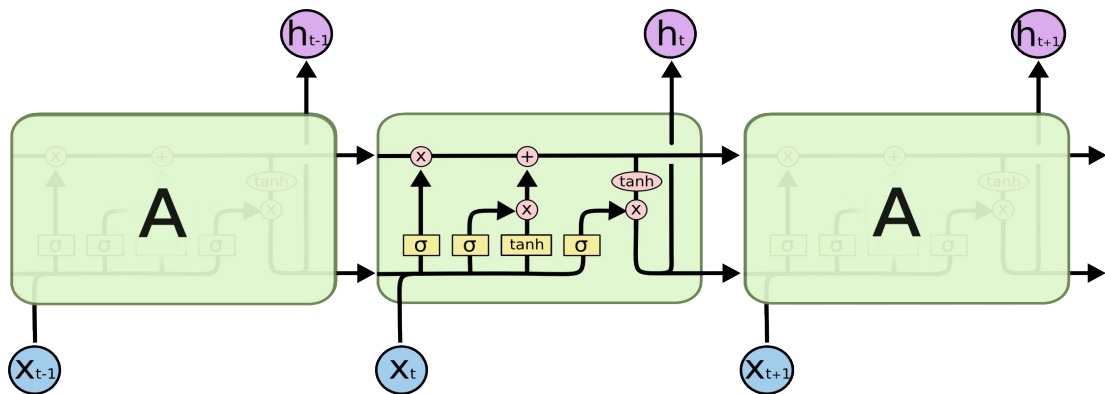| 2012	Bag of words (tf-idf)
2013	RNN / LSTM (word Embeddings - word2vec)
2015	LSTM (word Embeddings + attention)
2018-19	Pre-trained Language Model (ULMFit, BERT) (Transfer Learning in NLP)

RNN - Recurrent Neural Network
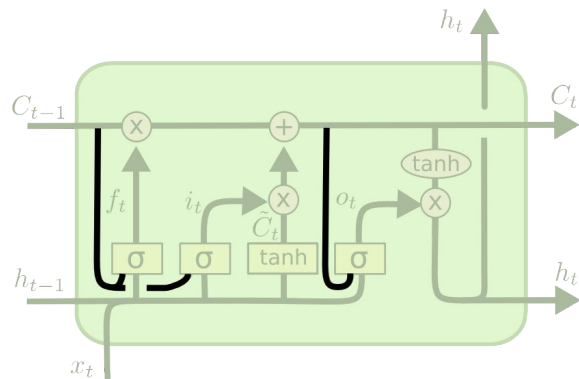# LSTM - Long Short Term Memory

RNN
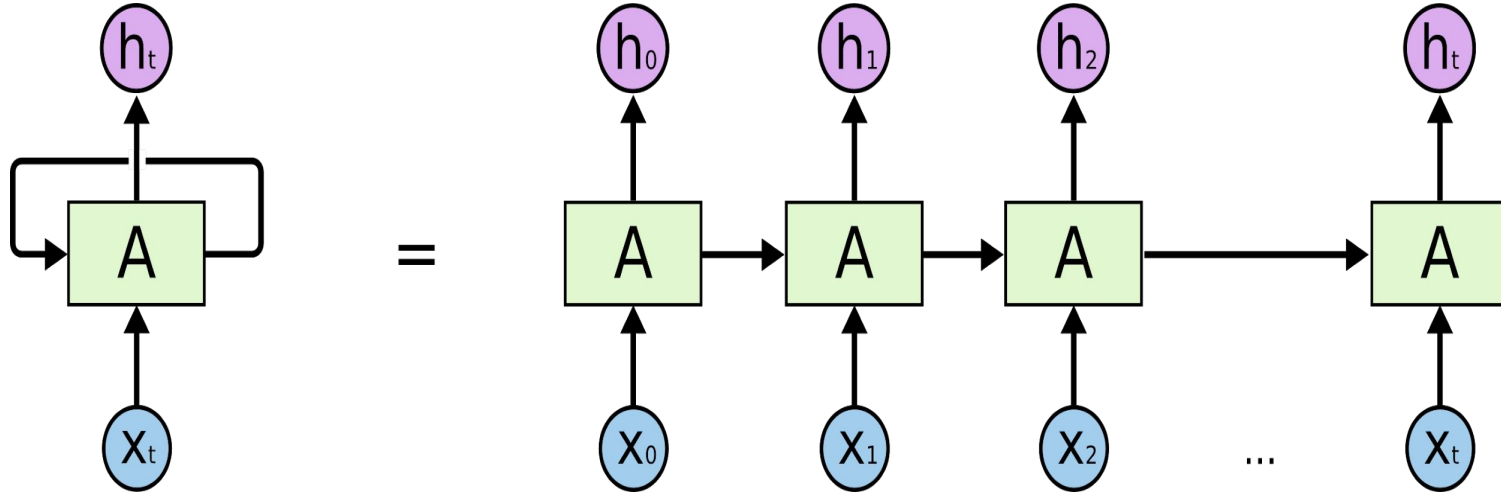
LSTM

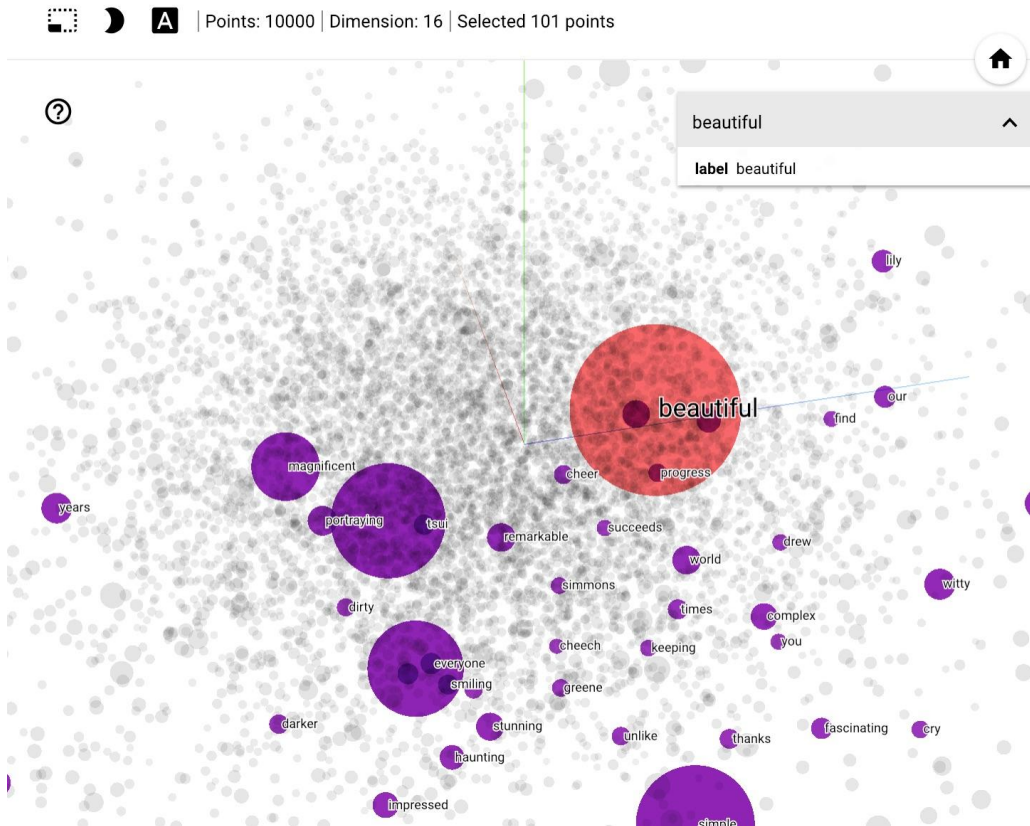$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

$$h_t = o_t * \tanh(C_t)$$

$$f_t = \sigma \left(W_f \cdot [\boldsymbol{C_{t-1}}, h_{t-1}, x_t] \ + \ b_f \right)$$

$$i_t = \sigma \left(W_i \cdot [\boldsymbol{C_{t-1}}, h_{t-1}, x_t] \ + \ b_i \right)$$

$$o_t = \sigma \left(W_o \cdot [\boldsymbol{C_t}, h_{t-1}, x_t] \ + \ b_o \right)$$

Reference

Word Embeddings

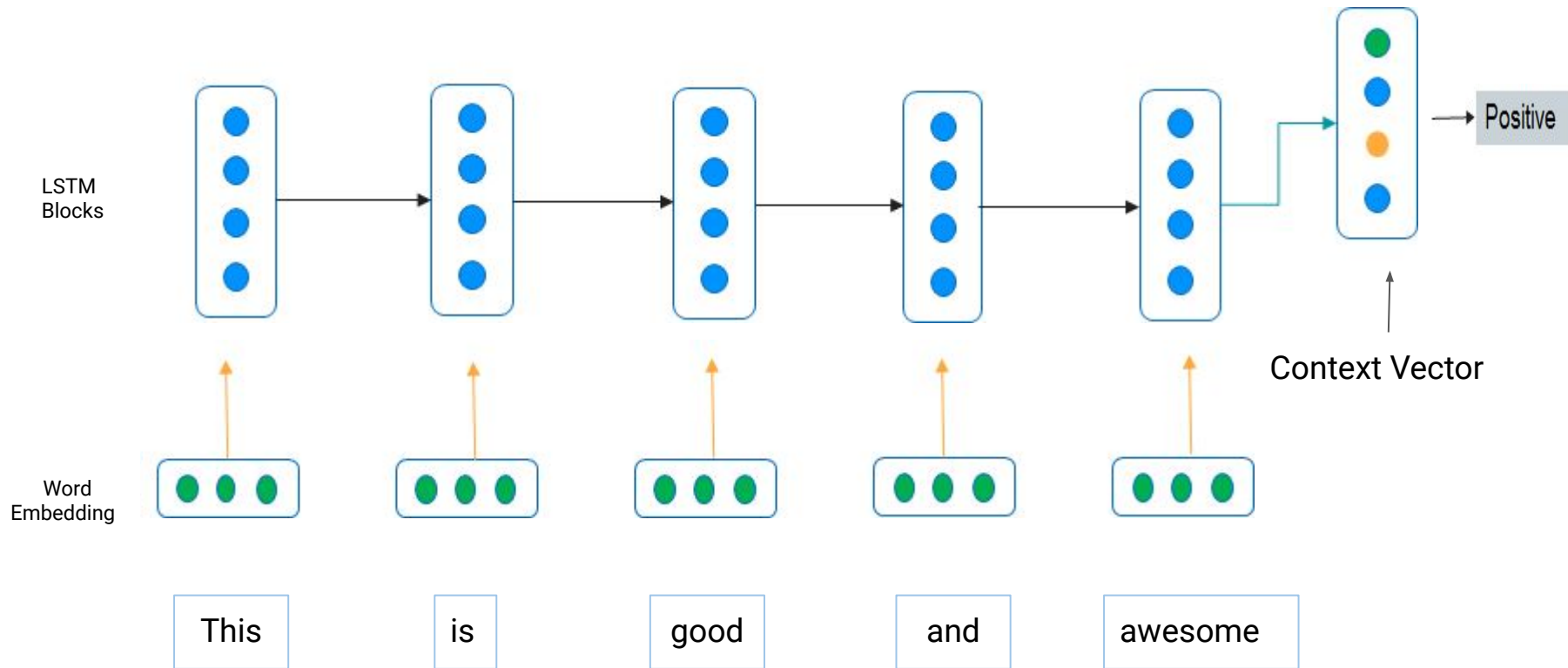# Word2Vec
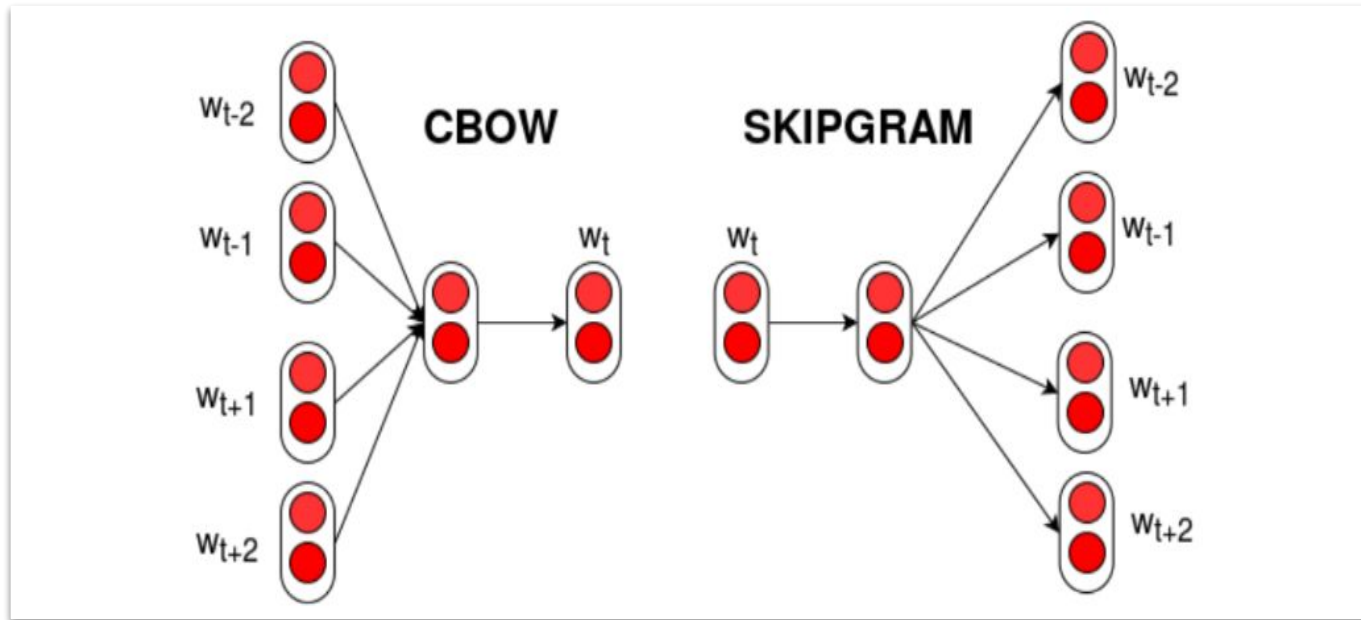
Other Embeddings:
- *Pennington et al. (2014)*: GloVe
- *Bojanowski et al. (2017)*: fastText

LSTM Text Classification

LSTM Blocks

Word Embedding

This

is

good

and

awesome

Positive

Context Vector

LSTM (word Embeddings + attention)

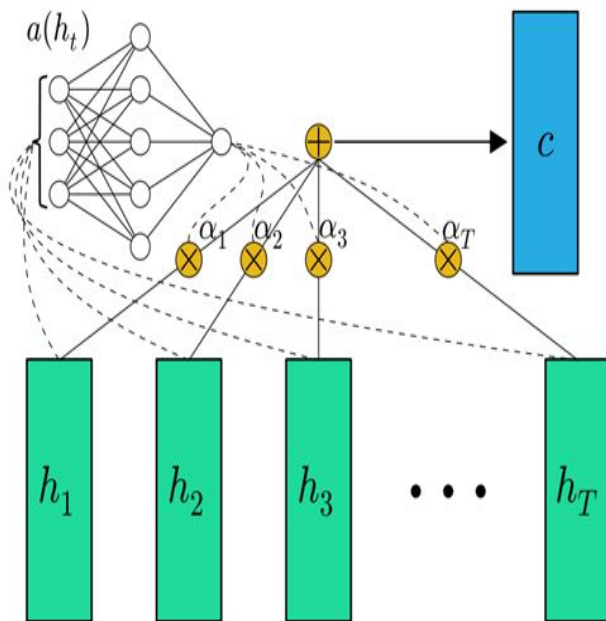| 2012	Bag of words (tf-idf)
2013	LSTM (word Embeddings - word2vec)
2015	LSTM (word Embeddings + attention)
2018-19	Pre-trained Language Model (ULMFit, BERT) (Transfer Learning in NLP)

Why Attention?

One of his best and most shocking films ! divine is the most filthy person ever ! mink stole also delivers a superb performance

This is terrible movie , don't waste your money on it, don't even watch it for free , that is what i have to say.

One of his best and most shocking films ! divine is the most filthy person ever ! mink stole also delivers a superb performance

$$u_i = \tanh(W_s h_i + b_s),$$

$$\alpha_i = \frac{\exp(u_i^\top u_s)}{\sum_i \exp(u_i^\top u_s)},$$

$$v = \sum_i \alpha_i h_i,$$

Reference

Language Model

2012	Bag of words (tf-idf)
2013	LSTM (word Embeddings- word2vec)
2015	LSTM (word Embeddings + attention)
2018-19	Pre-trained Language Model (ULMFit, BERT) (Transfer Learning in NLP)

Language Model: Frequency based

Chain Rule:

- P(A,B,C,D) = P(A)P(B|A)P(C|A,B)P(D|A,B,C)

- *p(B|A) = P(A,B)/P(A)*

P(**"its water is so transparent"**) =P(its) × P(water|its) × P(is|its water) × P(so|its water is) ×

P(transparent|its water is so)

Bi-gram Model

Library - **kenlm**

P("**its water is so transparent**") = P(its) × P(water|its) × P(is|water) × P(so|is) × P(transparent|so)

Spell - correction Example

versatile **acress** whose

P("versatile actress whose")   >   P("versatile across whose")

SwiftKey

Language Model

Jane   is   visiting

Decoder
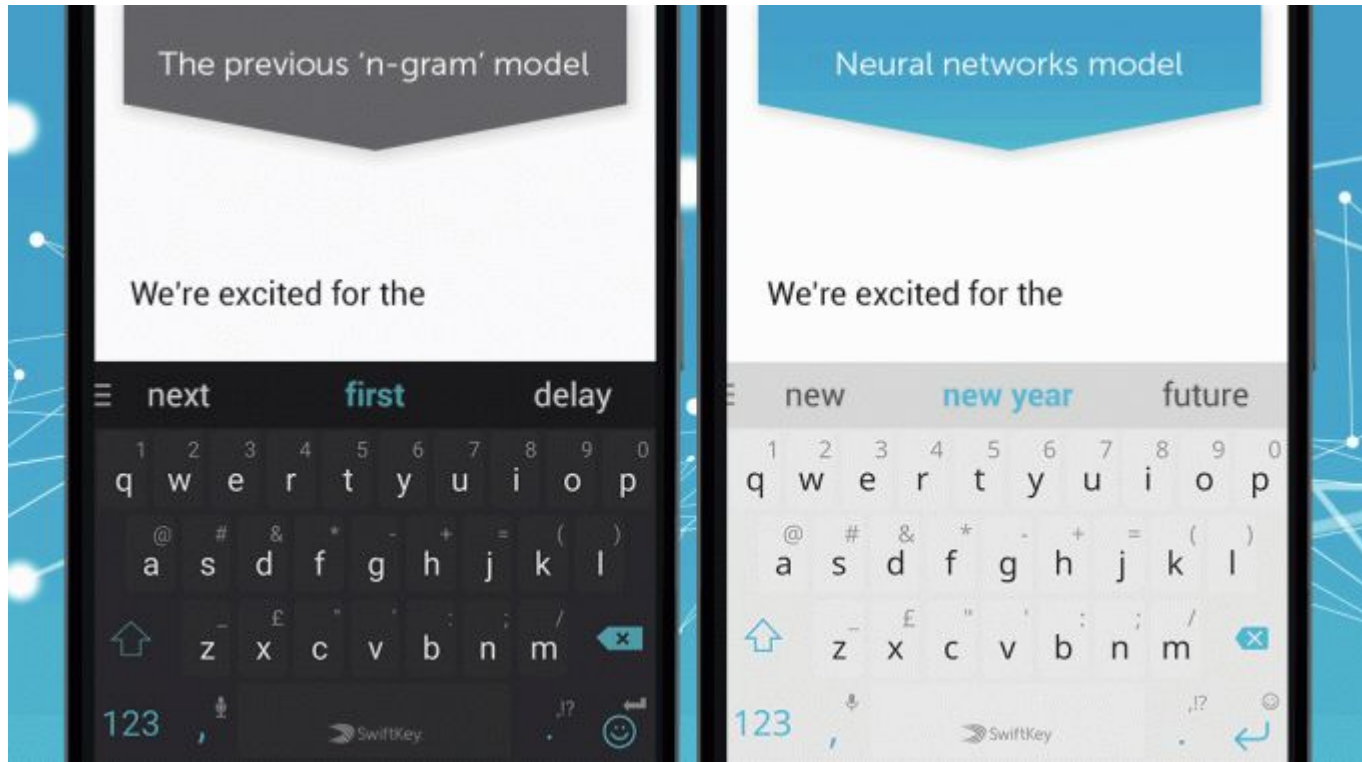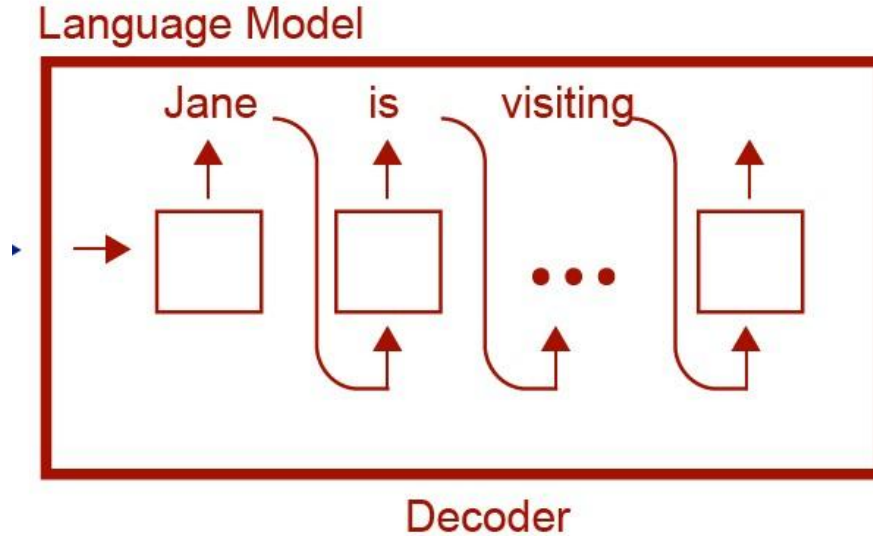
- generate most likely output depending on input

RNN / LSTM Block

GPT2 Language Model

Give the model a prompt, or start telling a story.

Kaggle Kernels

Welcome to Kaggle Kernels

The best place to explore data science results and share your own work

Code
Skip the download. Kernels is preloaded with the most common data science languages and libraries.

Learn
Gain exposure to new tools and techniques. The "hottest" kernels showcase the best work on Kaggle.

Mentor
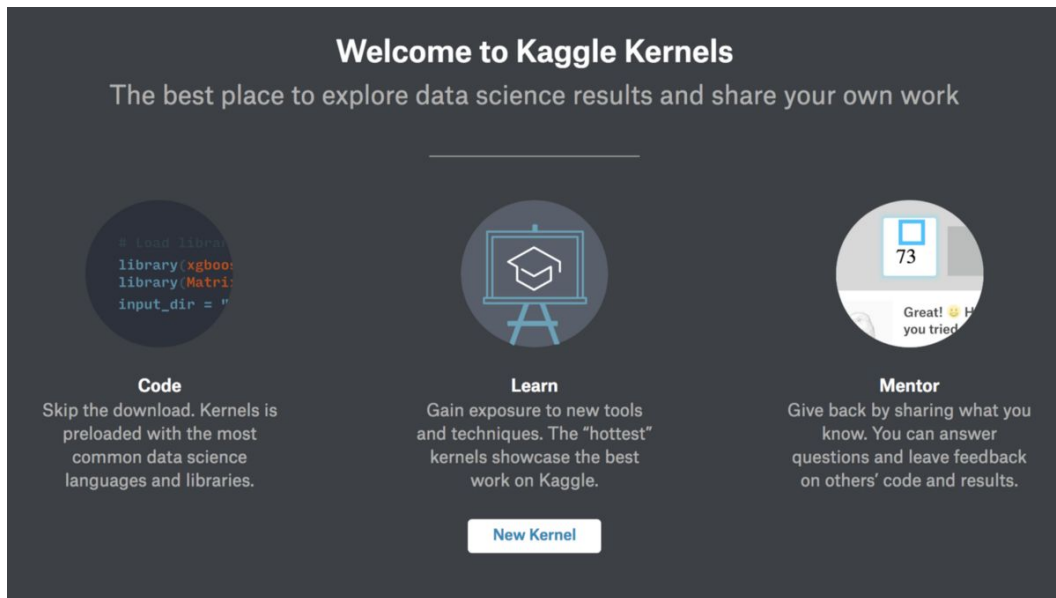Give back by sharing what you know. You can answer questions and leave feedback on others' code and results.

New Kernel

Documentation

- a free cloud based platform to run code in the browser (Scripts / Jupyter Notebooks)
- can write scripts in R as well as Python
- Install any package and use any dataset for experiment
- Free GPU (~ 10 hrs for a kernel)
- can run more than 8 kernels in parallel

Transfer Learning (ULMFiT, BERT)

2012	Bag of words (tf-idf)
2013	LSTM (word Embeddings- word2vec)
2015	LSTM (word Embeddings + attention)
2018-19	Pre-trained Language Model (ULMFiT, BERT) (Transfer Learning in NLP)

ImageNet Challenge
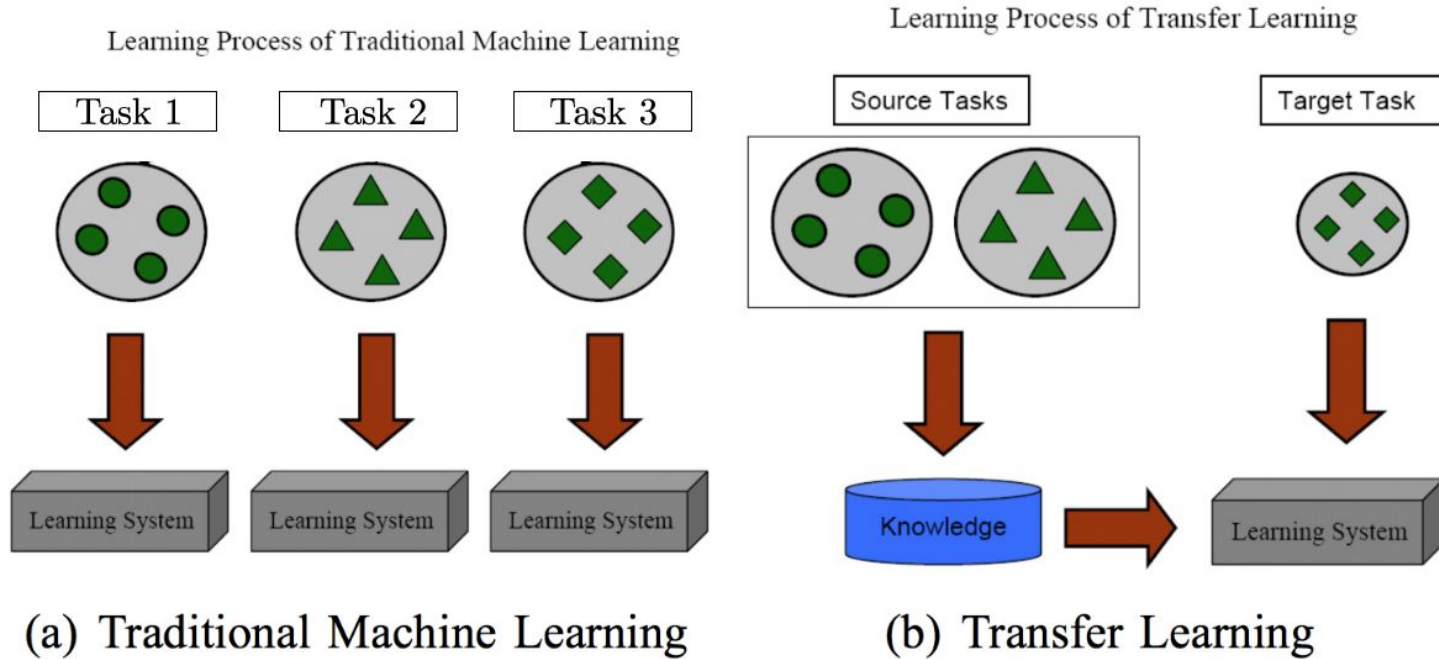
IMAGENET

- 1,000 object classes (categories).
- Images:
  - 1.2 M train
  - 100k test.

Where does Transfer Learning work ?

Learning Process of Traditional Machine Learning

Learning Process of Transfer Learning

Task 1 | Task 2 | Task 3

Source Tasks | Target Task

Learning System | Learning System | Learning System

Knowledge | Learning System

(a) Traditional Machine Learning

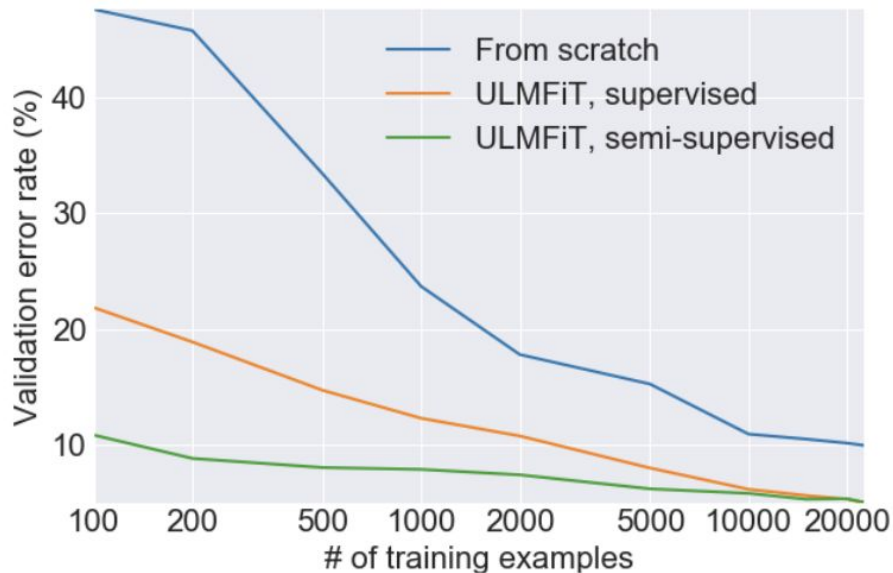(b) Transfer Learning

Reference

Transfer Learning: NLP

ULMFIT: Universal Language Model Fine tuning
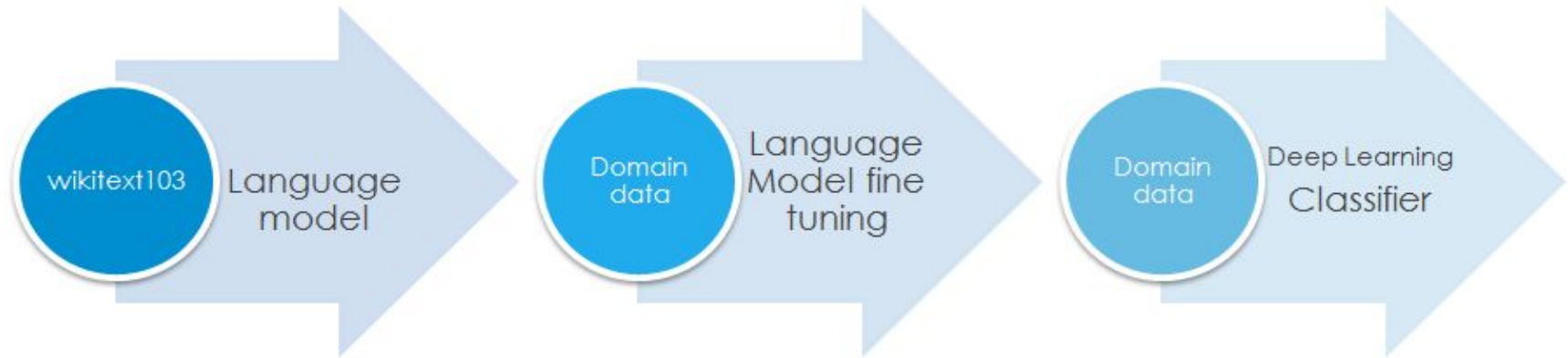
ULMFiT Accuracy

SOTA results for various text classification dataset

BERT: **ImageNet in NLP**

Bidirectional Encoder Representations from Transformers

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

Model:

BERT

Dataset:

WIKIPEDIA
Die freie Enzyklopädie

Objective: Predict the masked word (langauge modeling)

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

Supervised Learning Step

Classifier → 75% Spam
25% Not Spam

Model:
(pre-trained in step #1)

BERT

Dataset:

Email message	Class
Buy these pills	Spam
Win cash prizes	Spam
Dear Mr. Atreides, please find attached...	Not Spam

BERT: Multi- Task learning
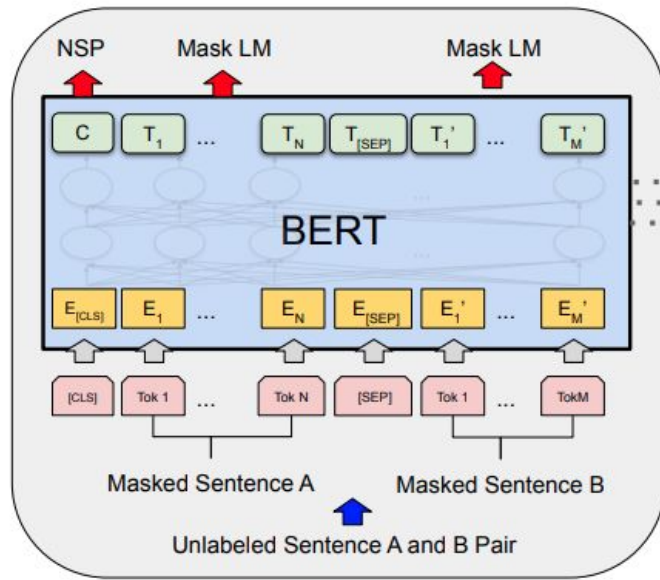
Pre-training
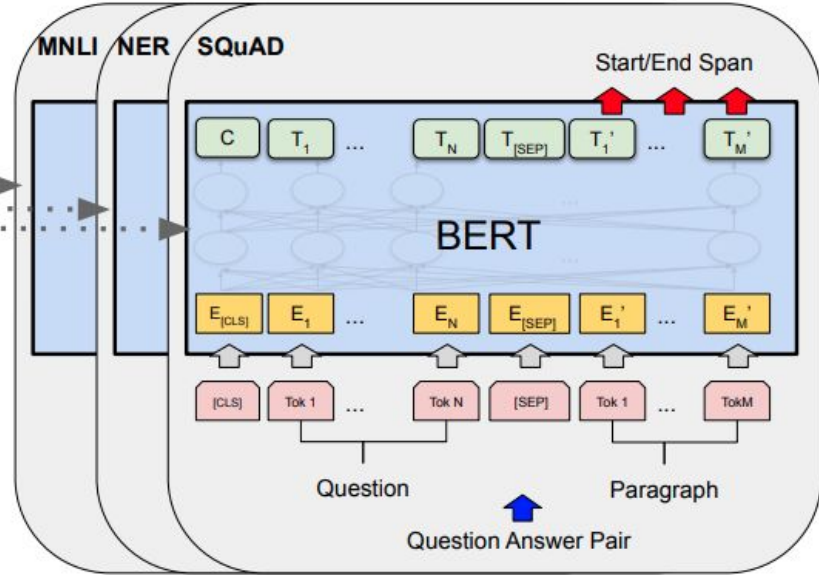
Fine-Tuning

SOTA for different tasks like Text classification, NER, Question - Answer

How to get started in NLP ?

Courses
- Andrew Ng's Deep Learning Specialization
- Deep Learning for NLP with Pytorch
- CS224n: Natural Language Processing with Deep Learning
- Fastai course

Blogs / Competitions
- Kaggle Competitions
- Sebastian Ruder blog on NLP
- Medium blog - Hugging Face

GROFERS   Google   LOGICAi   kaggle   CODING NINJAS
                                                           www.codingninjas.in

Conclusion

- Best place to learn state of the art method in NLP, Computer Vision & Classical ML techniques
- Free GPU for any experiment
- Join the competition early when it launches
- Try to learn from kaggle kernels
- Create proper train validation split
- Read discussion forums carefully and try to implement some of the ideas
- Apply learnings from kaggle to your projects

Happy learnings & Never Give up !!

References

1. https://sigmoidal.io/machine-learning-terminology-explained-top-8-must-know-concepts/
2. https://towardsdatascience.com/introducing-natural-language-processing-nlp-series-3-covering-essentials-of-nlp-phase-2-38839ff846de
3. https://hackernoon.com/beam-search-attention-for-text-summarization-made-easy-tutorial-5-3b7186df7086
4. https://rajpurkar.github.io/
5. https://mlwhiz.com/blog/2019/01/17/deeplearning_nlp_preprocess/
6. https://colah.github.io/posts/2015-08-Understanding-LSTMs/
7. https://nlp.stanford.edu/projects/glove/
8. http://nlp.fast.ai/classification/2018/05/15/introducting-ulmfit.html
9. http://ruder.io/
10. https://venturebeat.com/2019/02/14/openai-let-us-generate-text-with-an-ai-model-that-achieves-state-of-the-art-performance-in-several-nlp-tasks/
11. https://blog.swiftkey.com/swiftkey-debuts-worlds-first-smartphone-keyboard-powered-by-neural-networks/
12. https://lilianweng.github.io/lil-log/2018/06/24/attention-attention.html
13. https://arxiv.org/abs/1607.04606
14. https://www.kaggle.com/c/jigsaw-unintended-bias-in-toxicity-classification/overviewb
15. http://jalammar.github.io

Thank you !!

Q/A Session