Course: Intelligent Systems

Unit 4: Language Technologies

Language technologies
Part 3/3

Mariano Rico
2021
Technical University of Madrid

First of all

- Take the satisfaction survey (30 min) http://servicios.upm.es/encuestas
  - Evaluate anonymously your teachers
    - Mari Carmen Suárez/Asunción Gómez
    - Daniel Manrique
    - Martín Molina
    - Mariano Rico

NLP at a glance

- Session 1
  - Encodings
  - Corpus
  - Normalization
  - Hands-on 1
- Session 2
  - Part of Speech
  - Sparsed Vector models
  - TF-IDF
  - Document classification
  - Hands-on 2
- Session 3 (**today**)
  - The neural revolution
  - Transformers
  - BERT / DestilBERT /RoBERTA
  - Language Models 4 NLP tasks
  - Hands-on 3

Table of Contents

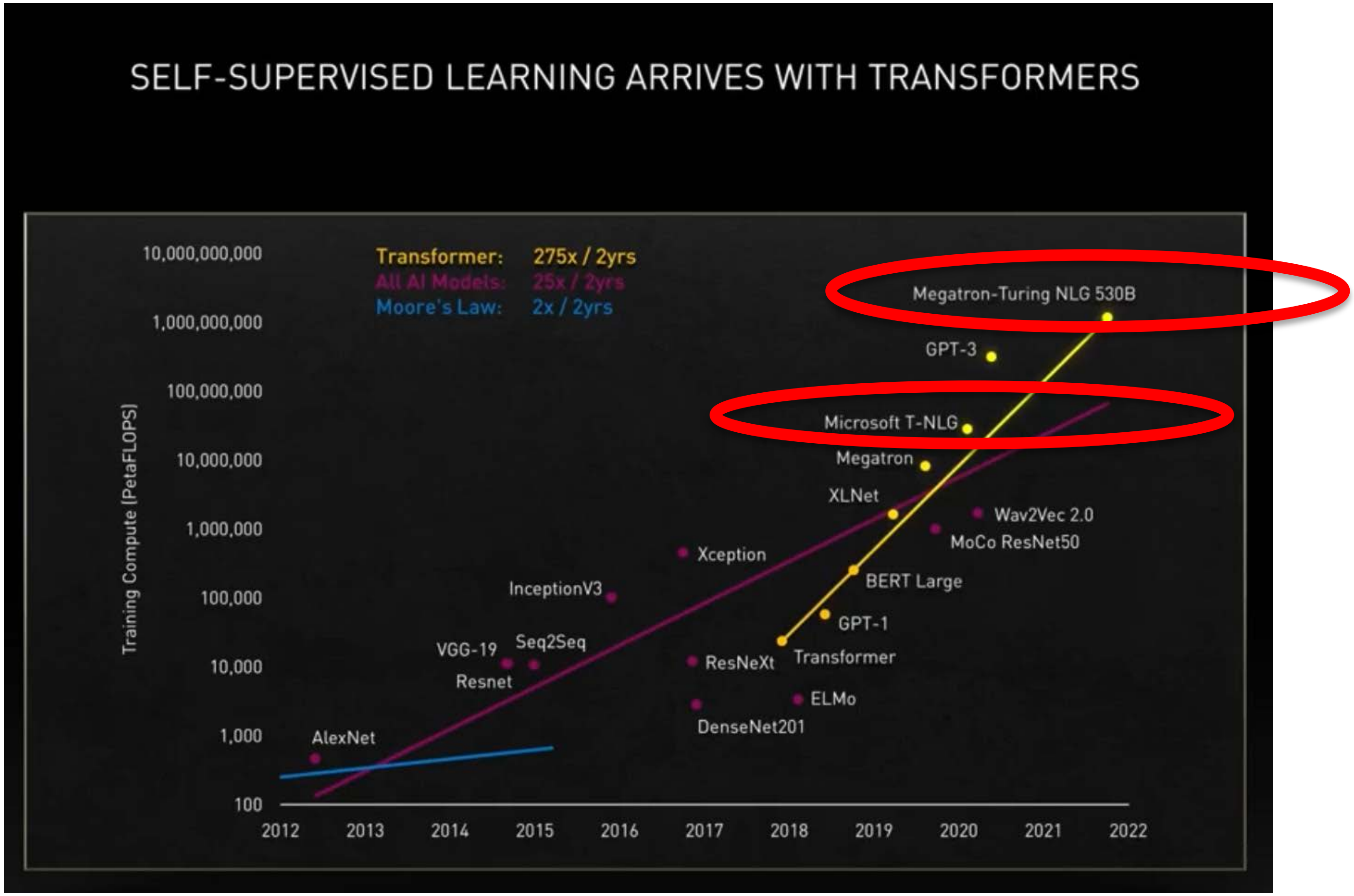
Acknowledgements

- Thanks to Pablo Calleja
  - Many slides in this presentation were made by him

THE NEURAL (R)EVOLUTION

A technological race

- Nov. 2021



NVIDIA [Nov. 2021 GTC Keynote]() by Jensen Huang

A technological race

- Dec. 2021 (less tan 1 month later)

Microsoft Research Blog

Efficiently and effectively scaling up language model pretraining for best language representation model on GLUE and SuperGLUE

Published December 2, 2021

By Jianfeng Gao, Distinguished Scientist & Vice President; Saurabh Tiwary, Vice President & Distinguished Engineer

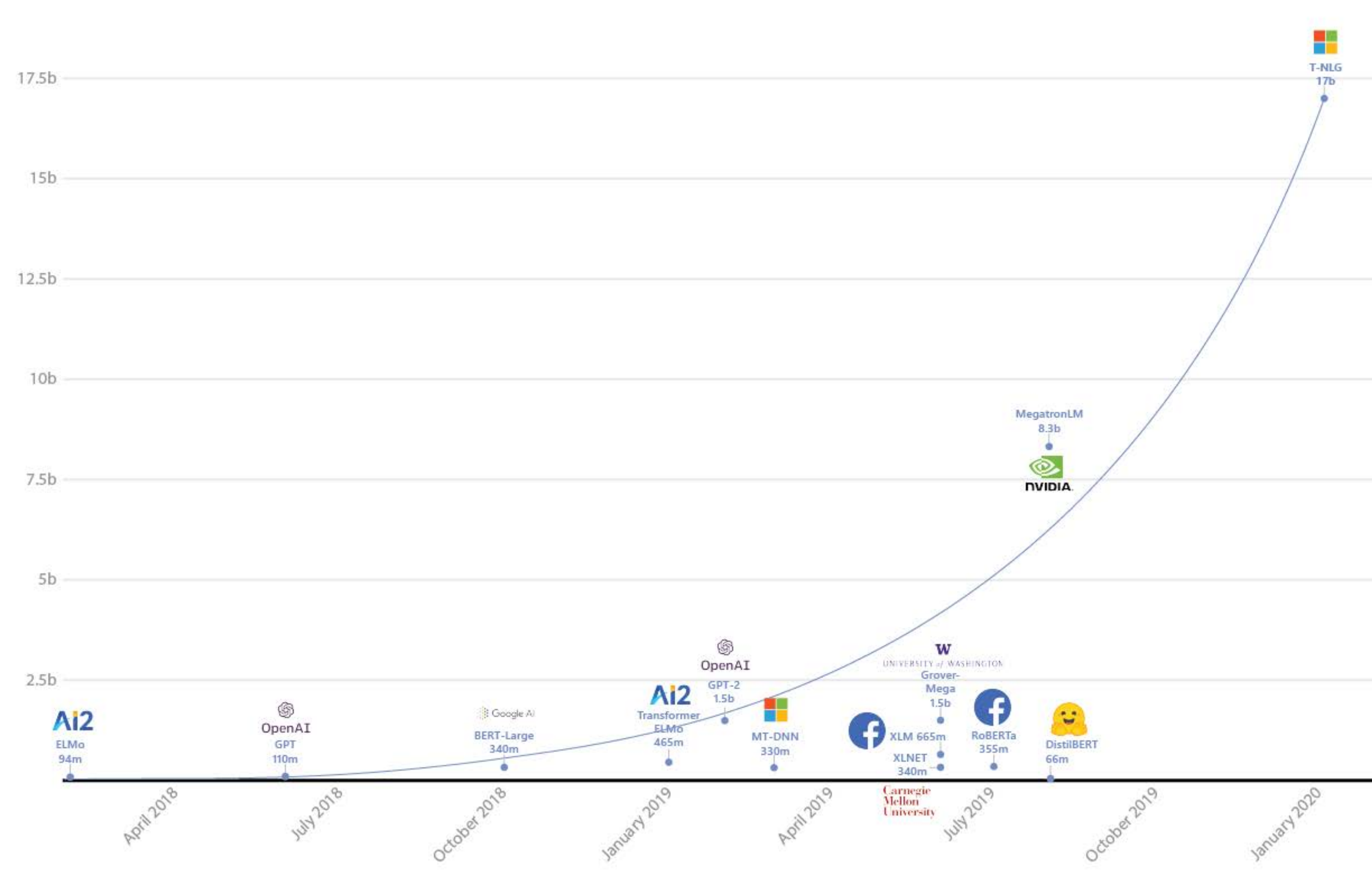
Research Area

Artificial intelligence

MS Research blog Dec. 2021

A technological race

- Evolution: number of parameters and actors



MS Research blog Feb. 2020

A technological race

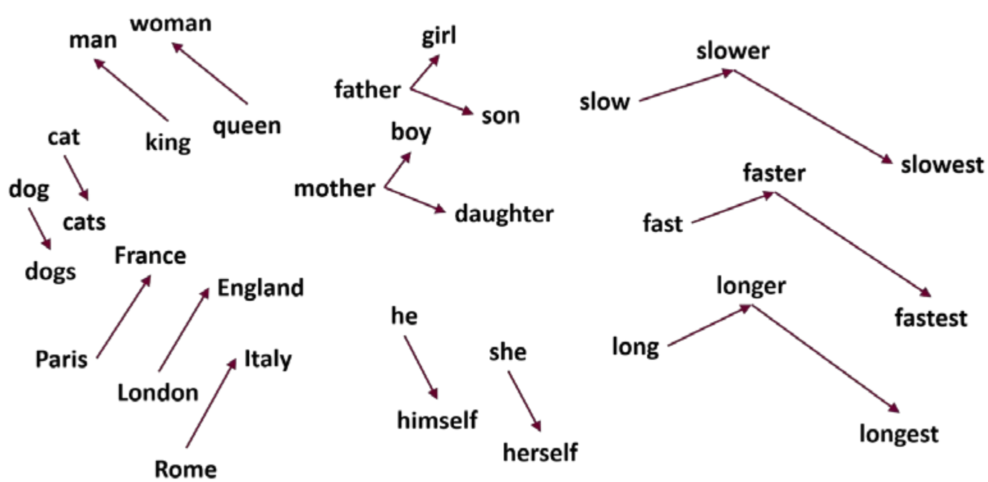
- (R)evolution: things to come
  - Explainable AI
    - Can you trust current AI?. Beyond a black-box model for neural systems
  - Reduction of hardware dependency
    - Do you have hardware to create a neural model?
    - What is the carbon fingerprint of creating a huge model?
    - I am a minority language. How can I get a model for my language?

All started with embeddings

- Distributional Hypothesis (Harris, 1954)

  Words with similar meanings tend to occur in similar contexts

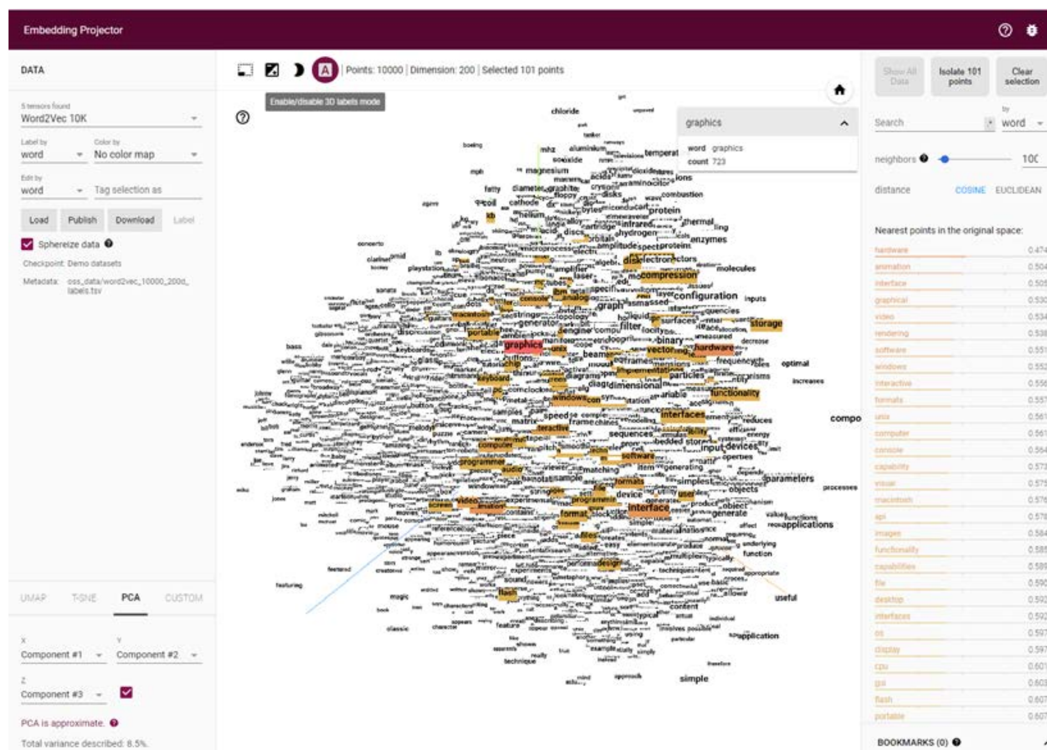
- Word2Vec ([Mikolov 2013](#))
  - Also relations!! ➔ semantic similarity!!



[Source](#)

All started with embeddings

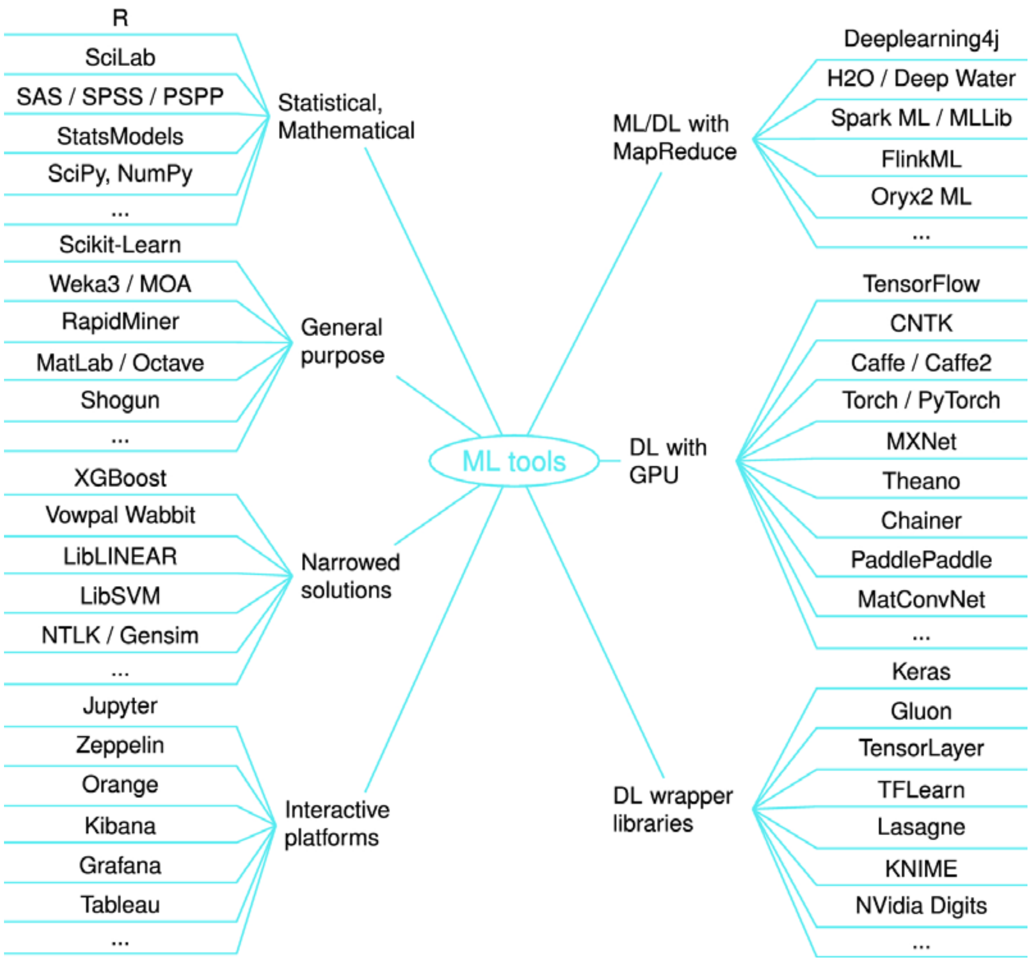
- Embeddings also have limitations
  - If a word becomes a numeric vector, how do we manage polysemy? (“play” → theater?, game?)
  - Fortunately, there are ways to identify the right meaning and then, assign the right embedding.
- Play with them [here](https://projector.tensorflow.org/)



Source: https://projector.tensorflow.org/

Development environments

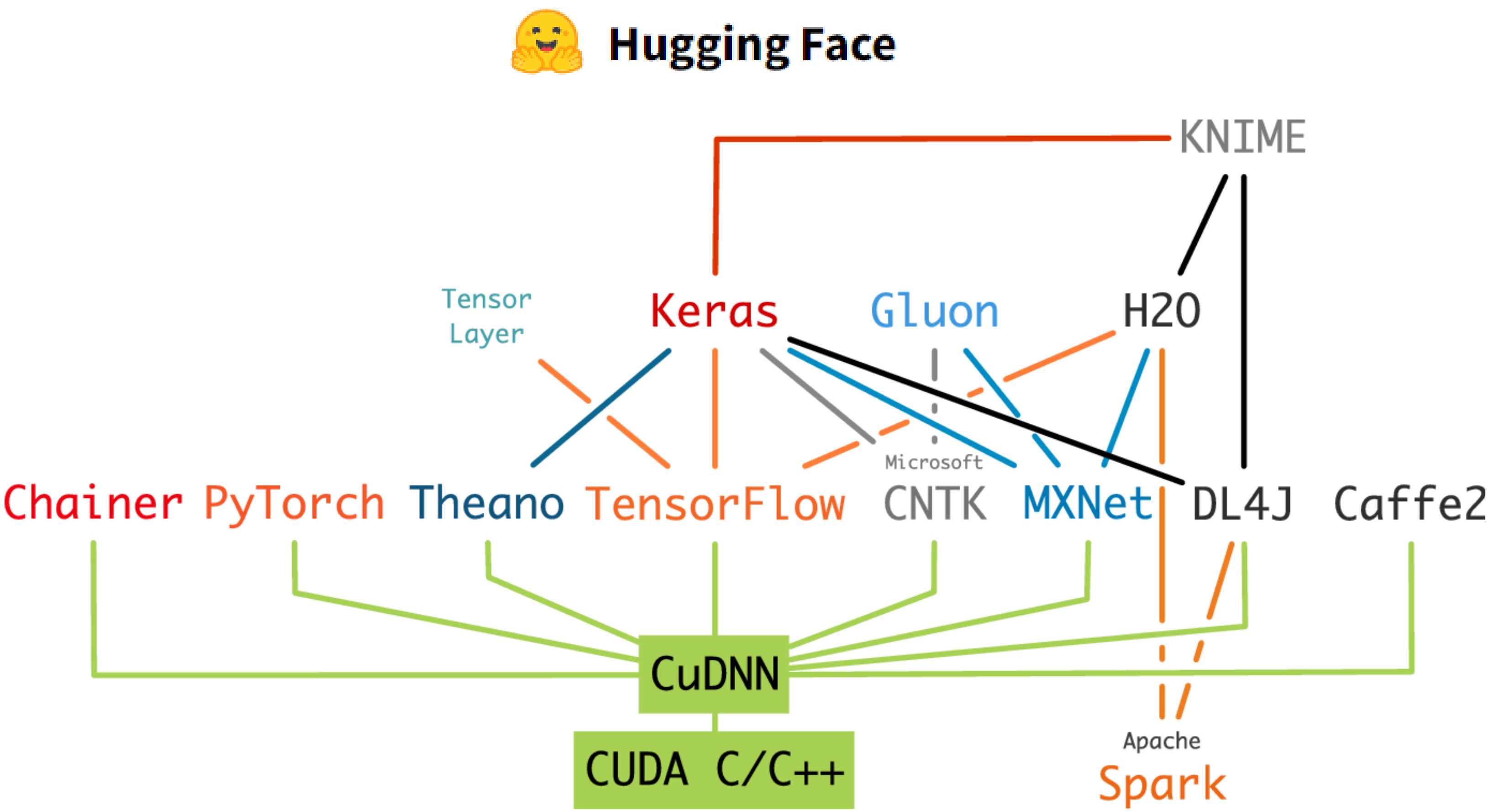
- ML frameworks and libraries



[Springer, Jan. 2019]

Development environments

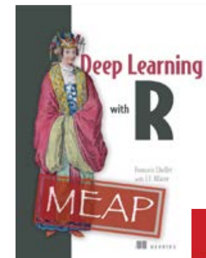
- DL frameworks and libraries



[Springer, Jan. 2019]

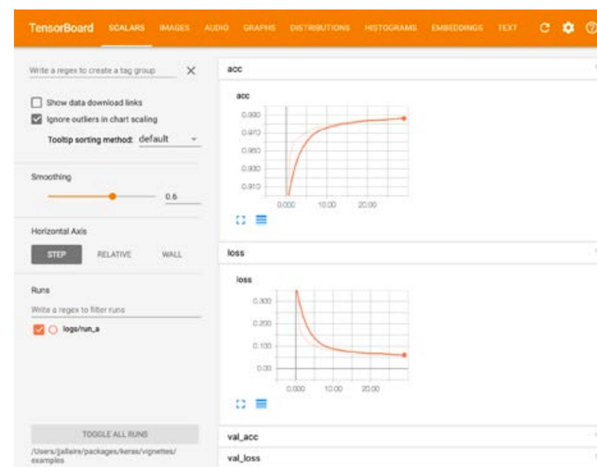
Deep Learning using R

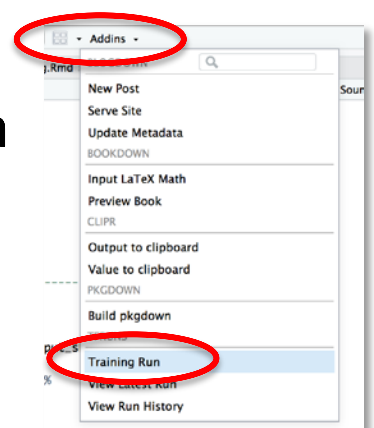
- Although most code is Python there are options for R:
  - Keras from Rstudio (keras.rstudio.com)
    - Cheatsheet (keras 2.1.2, 2017, before TF2)
      - A Spanish version by Carlos Ortega (R Users Madrid)
    - TensorBord: visualizaing the state of the neural net
    - TFruns: track and visualize training runs (integrated with Rstudio s an addin)



2018
(TF1)

2021
(TF2)

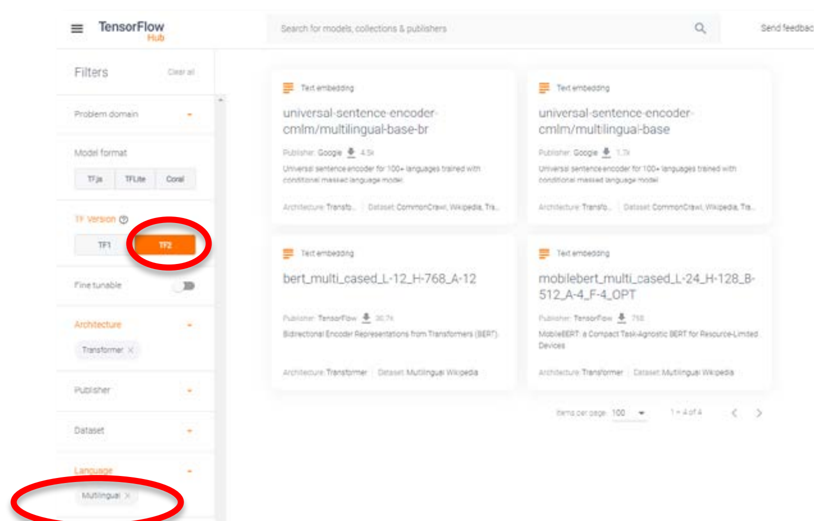




Deep Learning using R

- [Tensorflow](#) (TF1 y TF2)
  - You can usa local GPUs (only NVIDIA) but also cloud GPUs like
    - Google CloudML
    - Cloud Server (Amazon EC2, Google Compute Engine)
    - Paperspace Cloud Desktop (only TF1?)
  - Packge `tfhub`: using models from Tensorflow Hub as a keras layer
    - TF1 and TF2
    - No Spanish models, but there are multilingual (41 languages including es)
    - Many examples
      - » Simple transfer learning
      - » Text classification
      - » Attention (seq2seq almost Transformer)

2019
(pre Transformer)



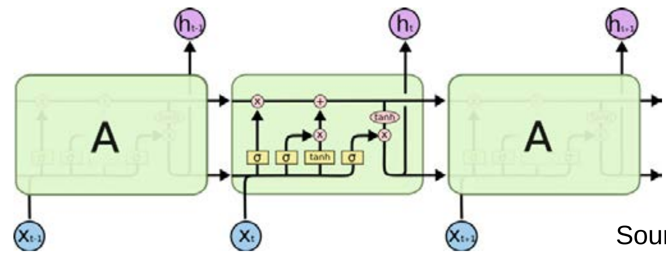
Deep Learning using R

- Using Hugging Face
  - Package `reticulate` can load any Python code
    - Even PyTorch
  - Package [`wrappingtransformers`]
    - Not in CRAN yet
    - Only a few models ☹

TRANSFORMERS

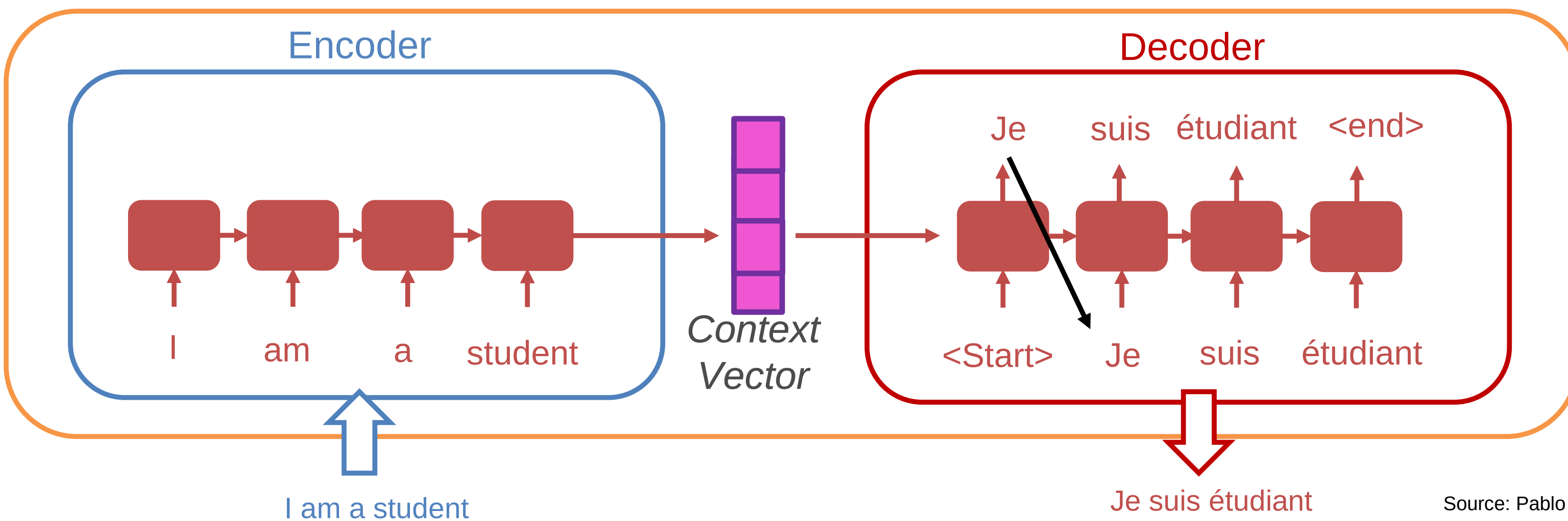
Why transformers

- Evolution of Recurrent Neural Networks (RNN)
  - LSTM
    - Relevant words are lost in long sentences (attention focused on nearby words)



Source: here

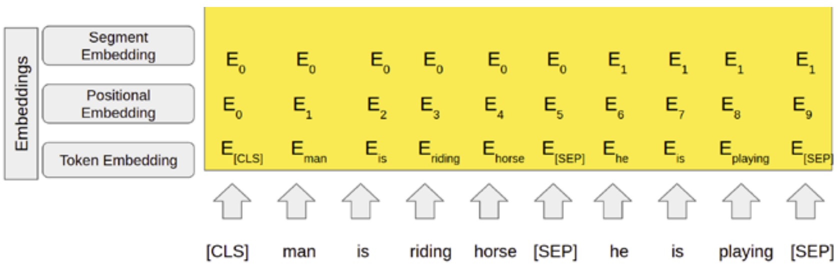
Neural Machine Translation (NMT) system



Source: Pablo Calleja

Why transformers

- Enhances the capture of context information
  - How?: Attention ([is all you need]())
    - Attention mechanism: an alignment score function to quantify the relevance of each token to another token
      - There are several types of attention mechanisms. Transformers use the *scaled dot-product attention*
  - Instead of processing word by Word (a RNNs do), the whole sentence in processed **in parallel**
  - Instead of 1 encoder and 1 decoder (a RNNs do), we have many of them
  - Uses positional embeddings for each token, as well as segment embeddings to separate sentences



- More info (barely math)
  - [Illustrated transformer]()

BERT / DISTIL**BERT** /RO**BERT**A

BERT

- BERT: Bidirectional Encoder Representations from Transformers
  – Encoder: the model uses the encoder part of the transformer
  – Bidirectional means:
    - Pay attention both forward and backwards tokens (transformers only backwards)
    - Achieved with a novel technique named Masked Language Model (MLM)

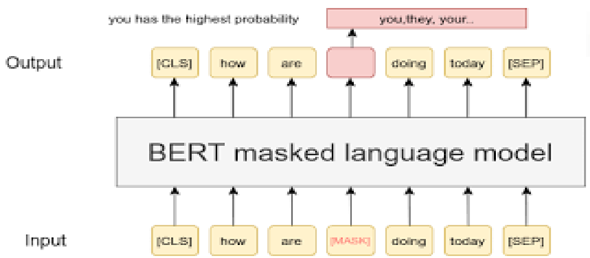
- The paper (v1 Oct. 2018, v2 May 2019)

BERT

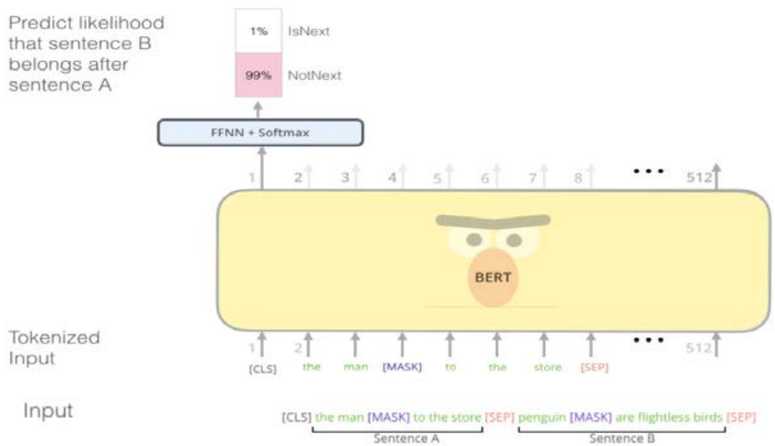
- BERT: Bidirectional Encoder Representations from Transformers
  - Designed to be used as a pre-trained model that can be fine-tuned
    - This pre-trained model can be slightly modified (typically by adding output neural layers) to perform NLP tasks such as:
      - Question answering
      - Sentiment analysis
      - Named entity recognition
      - Text summarization

BERT

- Model training for two different tasks:
  - Masked Language Model (MLM)
    - 15% of tokens in the input are masked
      - 80% replaced with [MASK]
      - 10% with a random Word
      - 10% with the original Word



  - Next Sentence Prediction (NSP)
    - BERT is trained with pairs of sentences and predicts if the second is the subsequent
      - 50% are subsequent pairs and 50% are random
      - Uses special tokens for the classification. [CLS] at the beginning, and [SEP] at the end of each sentence. [CLS] token is used to predict IsNext/NotNext



DistilBERT

- Created by Hugging Face (paper 2020)
- It is a *distilled* BERT
  - 40% smaller
  - 60% faster
  - Retains 97% of the language understanding capabilities
- Methodology
  - BERT is the “teacher” model. DistillBERT is a “student” model with
    - half number of layers (but keeping layer sizes)
    - Without token-type embeddings
    - Without pooling

RoBERTa

- Created by Facebook (paper 2019)
- It is a "Robustly optimized" BERT approach
  - Modifications to the BERT pre-training process:
    - Longer model training times
      - Larger batches and more data
    - Removed one of the two BERT tasks:
      - The *Next Sentence Prediction* (NSP) task
    - Longer sequences for training
    - Changes in the method used for masking the training data

Comparison of BERT-based models

	BERT	**RoBERTa**	**DistilBERT**	**XLNet**
Size (millions)	**Base**: 110 **Large**: 340	**Base**: 110 **Large**: 340	**Base**: 66	**Base**: ~110 **Large**: ~340
Training Time	**Base**: 8 x V100 x 12 days* **Large:** 64 TPU Chips x 4 days (or 280 x V100 x 1 days*)	**Large:** 1024 x V100 x 1 day; 4-5 times more than BERT.	**Base**: 8 x V100 x 3.5 days; 4 times less than BERT.	**Large:** 512 TPU Chips x 2.5 days; 5 times more than BERT.
Performance	Outperforms state-of-the-art in Oct 2018	2-20% improvement over BERT	3% degradation from BERT	2-15% improvement over BERT
Data	16 GB BERT data (Books Corpus + Wikipedia). 3.3 Billion words.	160 GB (16 GB BERT data + 144 GB additional)	16 GB BERT data. 3.3 Billion words.	**Base**: 16 GB BERT data **Large**: 113 GB (16 GB BERT data + 97 GB additional). 33 Billion words.
Method	BERT (Bidirectional Transformer with MLM and NSP)	BERT without NSP**	BERT Distillation	Bidirectional Transformer with Permutation based modeling

What about non English languages?

- Like Spanish
  - MarIA (by RAE+BSC)
    - Github repo with
      - Models (links to 🤗)
        - » RoBERTa (b & L)
        - » GPT2 (b & L)
      - Fine-tuned models for
        - » POS (Part of Speech)
        - » NER (Named Entity Recognition)
        - » QA (Question-Answering)
      - Evaluation results
      - Usage examples (Python)

First massive Artificial Intelligence system in the Spanish language, MarIA, begins to summarize and generate texts

11 November 2021

Launched five months ago, the system expands its capabilities to use the language. Creative and business applications and those related to the digitization of Public Administration increase.

Evaluation ✅

Dataset	Metric	RoBERTa-b	RoBERTa-l	BETO*	mBERT	BERTIN**	Electricidad***
UD-POS	F1	0.9907	0.9898	0.9900	0.9886	0.9898	0.9818
Conll-NER	F1	0.8851	0.8772	0.8759	0.8691	0.8835	0.7954
Capitel-POS	F1	0.9846	0.9851	0.9836	0.9839	0.9847	0.9816
Capitel-NER	F1	0.8960	0.8998	0.8772	0.8810	0.8856	0.8035
STS	Combined	0.8533	0.8353	0.8159	0.8164	0.7945	0.8063
MLDoc	Accuracy	0.9623	0.9675	0.9663	0.9550	0.9673	0.9493
PAWS-X	F1	0.9000	0.9060	0.9000	0.8955	0.8990	0.9025
XNLI	Accuracy	0.8016	0.7958	0.8130	0.7876	0.7890	0.7878
SQAC	F1	0.7923	0.7993	0.7923	0.7562	0.7678	0.7383

** A model based on BERT architecture.*

*** A model based on RoBERTa architecture.*

**** A model based on Electra architecture.*

For the RoBERTa-base

```python
from transformers import AutoModelForMaskedLM
from transformers import AutoTokenizer, FillMaskPipeline
from pprint import pprint
tokenizer_hf = AutoTokenizer.from_pretrained('PlanTL-GOB-ES/roberta-base-bne')
model = AutoModelForMaskedLM.from_pretrained('PlanTL-GOB-ES/roberta-base-bne')
model.eval()
pipeline = FillMaskPipeline(model, tokenizer_hf)
text = f"¡Hola <mask>!"
res_hf = pipeline(text)
pprint([r['token_str'] for r in res_hf])
```

What about non English languages?

- Like Spanish
  - flairNLP (Humbold Univ.)
    - NER models (links to 🤗) for several languages
      - English, German, Dutch, Spanish
      - Top performance
    - Also other models for POS
    - It is a development framework (Python + PyTorch)
      - With tutorials and an enthusiastic community

State-of-the-Art Models

Flair ships with state-of-the-art models for a range of NLP tasks. For instance, check out our latest NER models:

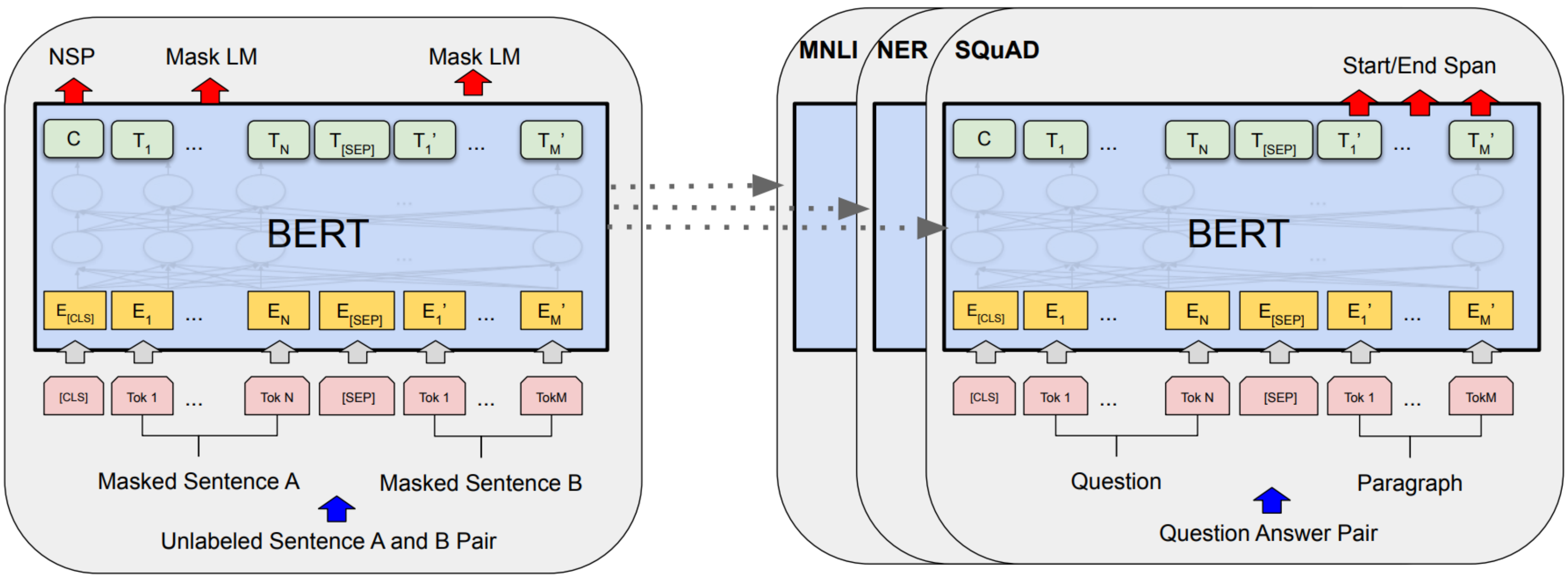
Language	Dataset	Flair	Best published	Model card & demo
English	Conll-03 (4-class)	**94.09**	*94.3 (Yamada et al., 2020)*	Flair English 4-class NER demo
English	Ontonotes (18-class)	**90.93**	*91.3 (Yu et al., 2020)*	Flair English 18-class NER demo
German	Conll-03 (4-class)	**92.31**	*90.3 (Yu et al., 2020)*	Flair German 4-class NER demo
Dutch	Conll-03 (4-class)	**95.25**	*93.7 (Yu et al., 2020)*	Flair Dutch 4-class NER demo
Spanish	Conll-03 (4-class)	**90.54**	*90.3 (Yu et al., 2020)*	Flair Spanish 4-class NER demo

The state of the art in NER: here

LANGUAGE MODELS 4 NLP TASKS

Fine-tuning BERT

Standing on the shoulders of giants



Pre-training

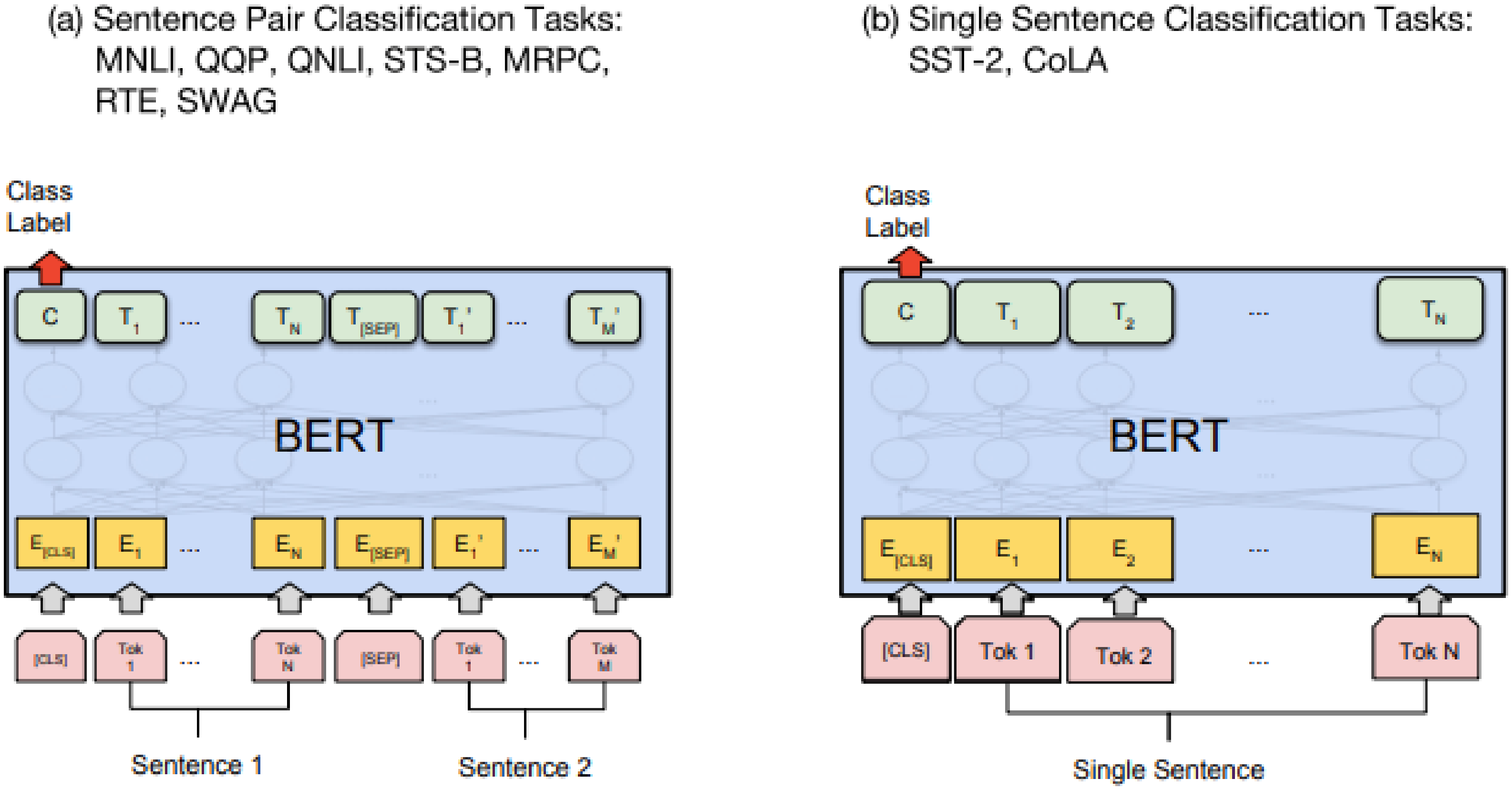
Fine-Tuning

There is also training, but less than the computational effort to create the pre-training model

Source: [BERT paper (2018-9)]

Fine-tuning BERT

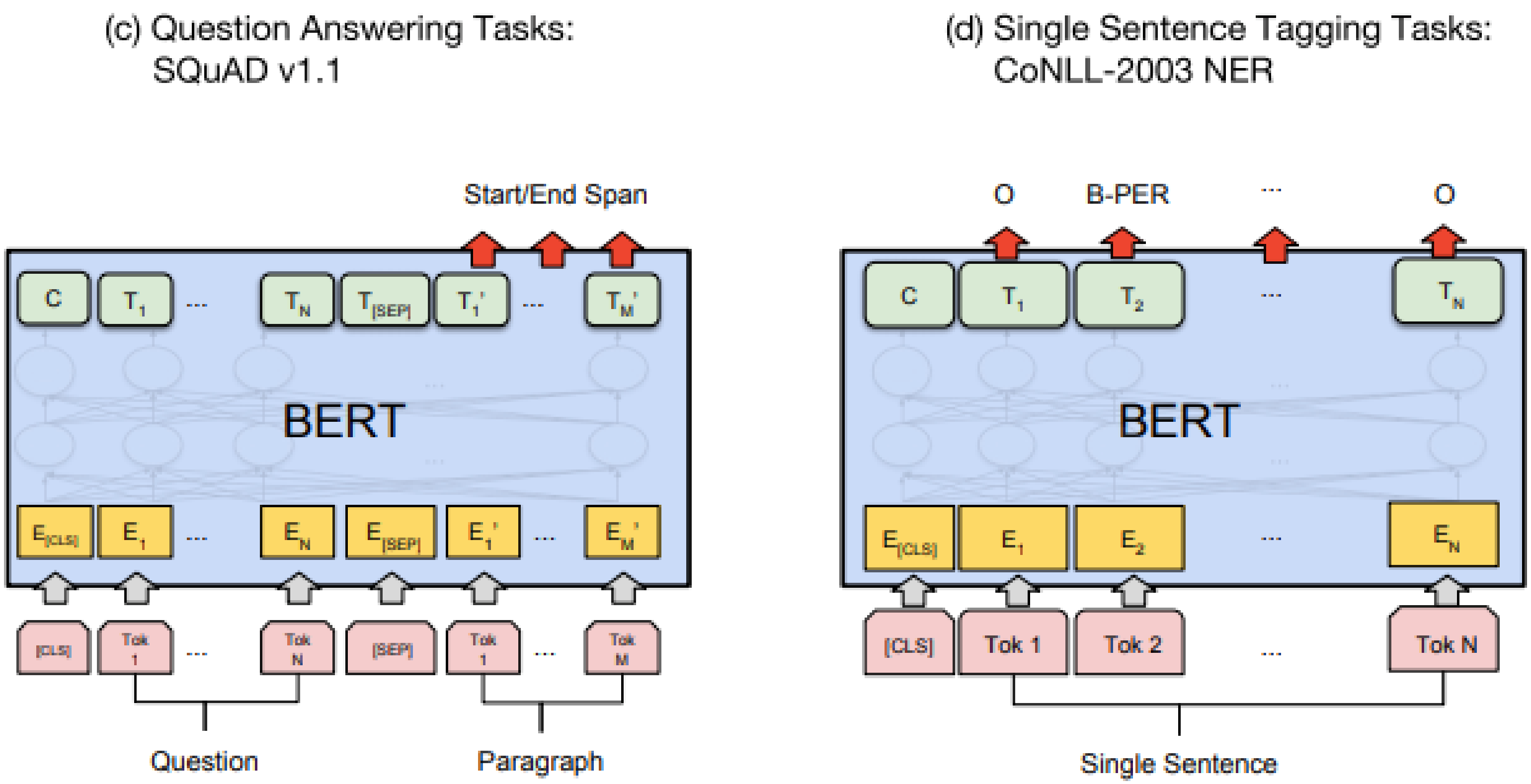
- BERT can be adapted for NLP tasks such as



Source: [BERT paper (2018-9)]

Fine-tuning BERT

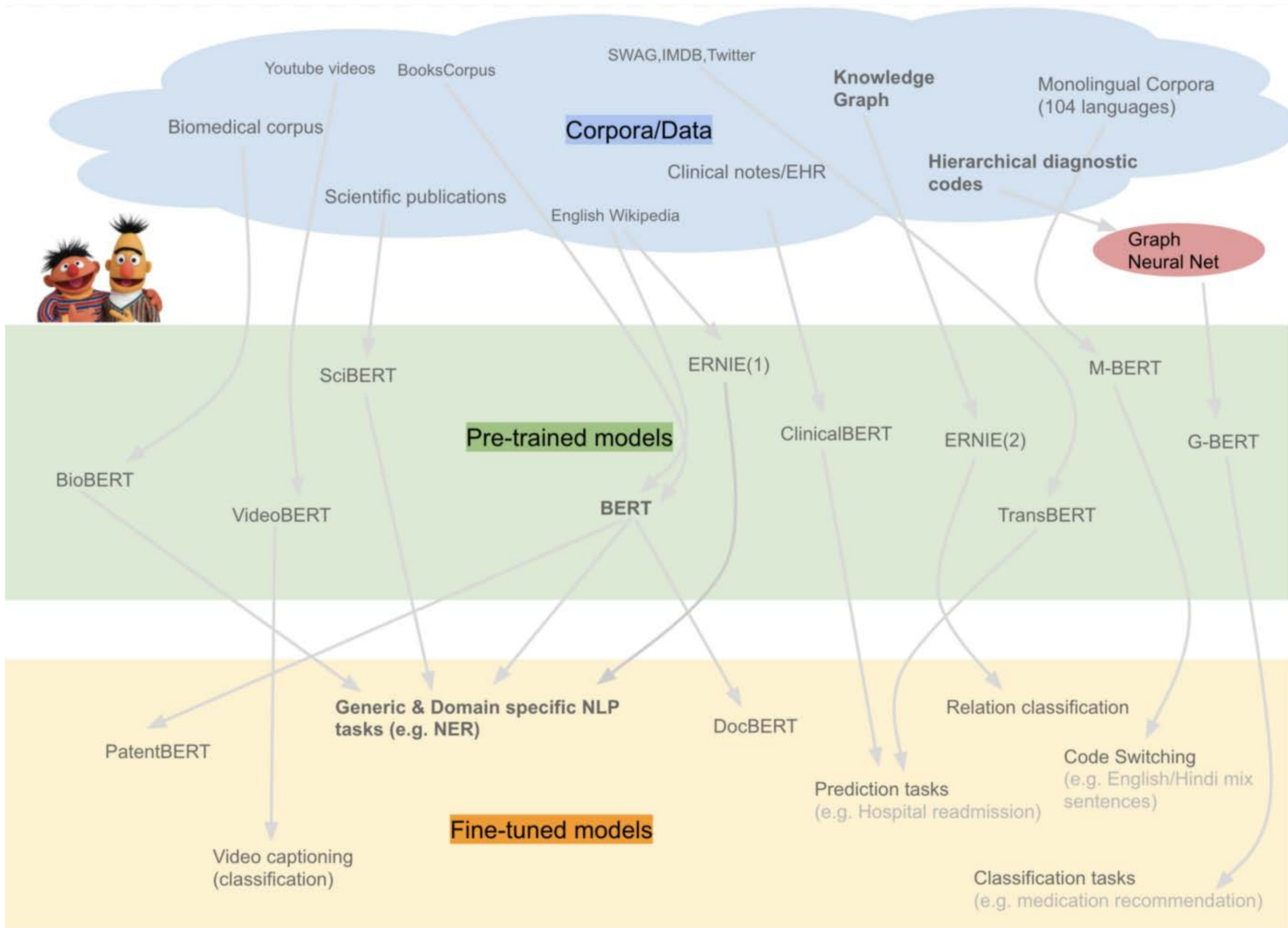
- BERT can be adapted for NLP tasks such as



Source: [BERT paper (2018-9)]()

Fine-tuning BERT

- Evolution and dependencies



Source: here

HANDS-ON 3

Hands-on 3

- **Goal 1**: to practice named entity recognition using lexical-syntactic patterns
  - **Materials:**
    - R script at: http://rpubs.com/rgcmme/IS-HO3
    - Gold standard in course's Moodle (NE Gold Standard.csv)
  - **Tasks:**
    - *Recognize entities*
      - Define lexical-syntactic patterns using regular expressions to detect person names
    - *Assess approach*
      - Check the results of your patterns using metrics
      - Which patterns have worked better? Why?
- **Goal 2**: Compare WordNet versus FrameNet
  - Tasks:
    - Given a set of terms (noun: bank, verb: run), compare both approaches (similarities, differences, pros, cons)
    - Study the RDF (ontology, instances) that can be generated with both approaches.
- **Goal 3**: Use neural methods for NLP tasks
  - Hands-on: to appear

Hands-on 3.
Practice Named Entity Recognition (NER) using lexical-syntactic patterns

GOAL 1

Identify persons in documents

films adapted from comic books have had plenty of success , whether they're about superheroes (batman , superman , spawn) , or geared toward kids (casper) or the arthouse crowd (ghost world) , but there's never really been a comic book like from hell before .

for starters , it was created by **alan moore** (and **eddie campbell**) , who brought the medium to a whole new level in the mid '80s with a 12-part series called the watchmen .

to say **moore** and **campbell** thoroughly researched the subject of **jack** the ripper would be like saying **michael jackson** is starting to look a little odd .

the book (or " graphic novel , " if you will) is over 500 pages long and includes nearly 30 more that consist of nothing but footnotes .

in other words , don't dismiss this film because of its source .

if you can get past the whole comic book thing , you might find another stumbling block in from hell's directors , **albert** and **allen hughes** .

Annotation guidelines

- If a person name is composed of two or more words, annotate the whole name
- If a person name is repeated with a different set of words, annotate it
- If a person name is repeated with the same set of words, do not annotate it
- Do not include qualifiers in the annotation
- Annotate fictitious persons
- Do not annotate fictitious names
- Keep misspellings in annotations

`created by` **`alan moore`** `(`
`and` **`eddie campbell`** `)`

`to say` **`moore`** `and` **`campbell`**

`last time` **`moore`** `and` **`campbell`**

`went to visit` **`dr`** **`jackson`**

`donald sinclair` `(` **`john cleese`** `)`

`(` **`batman`** `,` **`superman`** `,` **`spawn`** `)`

`schwartznager`

Gold standard

- Created a gold standard for 1,000 documents

	A	B	C	D	E	F
1	cv000_29590.txt	alan moore	eddie campbell	moore	campbell	jack
2	cv001_18431.txt	matthew broderick	reese witherspoon	george washington carver	tracy flick	paul
3	cv002_15918.txt	ryan	hanks	tom hanks	joe fox	meg rya
4	cv003_11664.txt	john williams	steven spielberg	spielberg	williams	martin b
5	cv004_11636.txt	herb	jackie chan	barry sanders	sanders	jackie
6	cv005_29443.txt	raoul peck	lumumba	patrice lumumba	eriq ebouaney	helmer p
7	cv006_15448.txt	tony kaye	edward norton	norton	derek vinyard	danny
8	cv007_4968.txt	betsy	molly ringwald	alan alda	ringwald	alda
9	cv008_29435.txt	lumumba	janssens	rudi delhem	moise tshombe	pascal n
10	cv009_29592.txt	schwartznager	stallone	van damme	rongguang yu	wong fei

goldStandard.csv

Comments on the gold standard

- We have a gold standard
  - Not validated
  - Normalized
    - No term repetition
    - No punctuation (".", ",". ";", ":", """, "'", "(", ")")
    - Trimmed whitespace
    - Lower case
- Things to note
  - Annotation guidelines
  - Inter-annotator agreement
- Any feedback on the quality of the gold standard will be appreciated!

Questions?

Course: Intelligent Systems

Unit 4: Language Technologies

Language technologies Part 3/3

Mariano Rico
2021
Technical University of Madrid