# EMBEDDING NLP ON USER DEVICE WITH TENSORFLOW.JS

## Konica Minolta @ Masaryk University – a day with industrial partners


**Lorenzo Beccaro**
Fullstack Developer

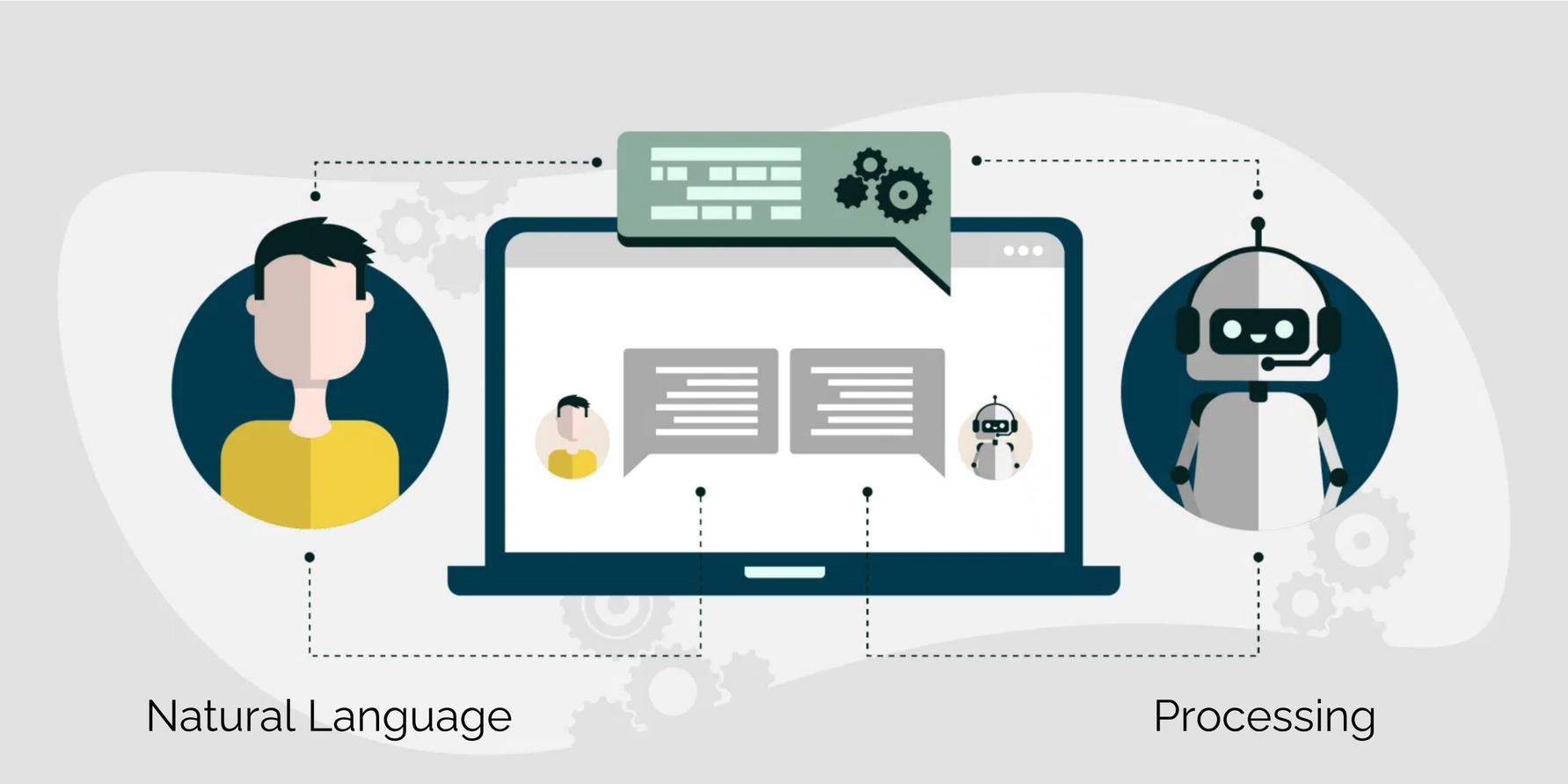
**Andrea Longhi**
Cognitive Services Developer

KONICA MINOLTA

# AGENDA

- What is NLP?
- Road to Transformers
- Question Answering on Digital Documents
- Transformers: using them with Tensorflow.js
- Hands-on Lab

# What is NLP?



# Road to Transformers

Higher

Human Intervention

Lower

- 1950-1980: **Symbolic** NLP
- 1990-2010: **Statistical** NLP
- 2010+: **Neural** NLP
- 2018: **BERT** (Google); a Transformer based on Attention mechanism

A **Transformer** is a way of solving sequence-to-sequence tasks, opening to **Transfer Learning** and relying on (self) **Attention**: a way to focus on relevant parts of the input

It can be used for **representation learning** too.

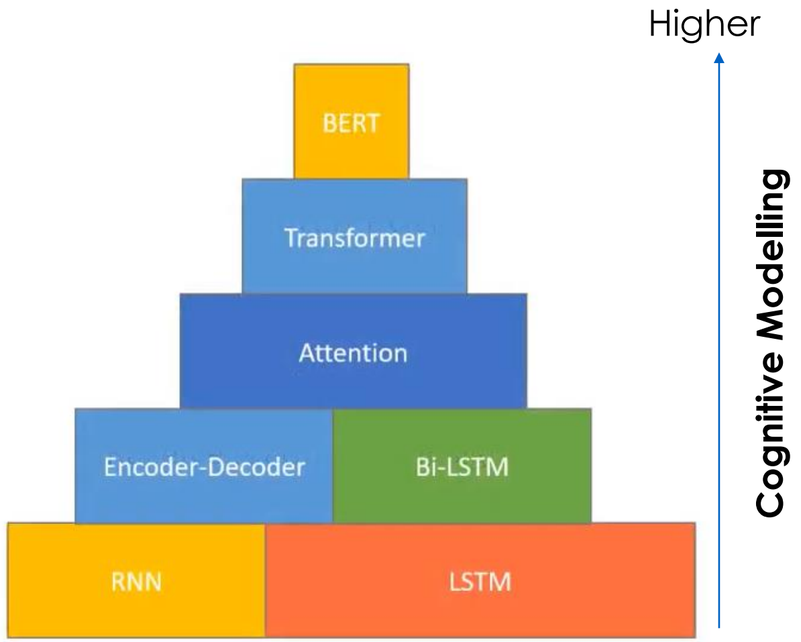
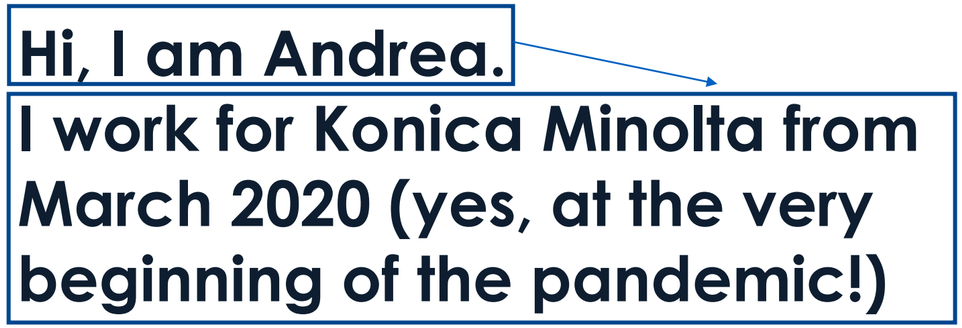


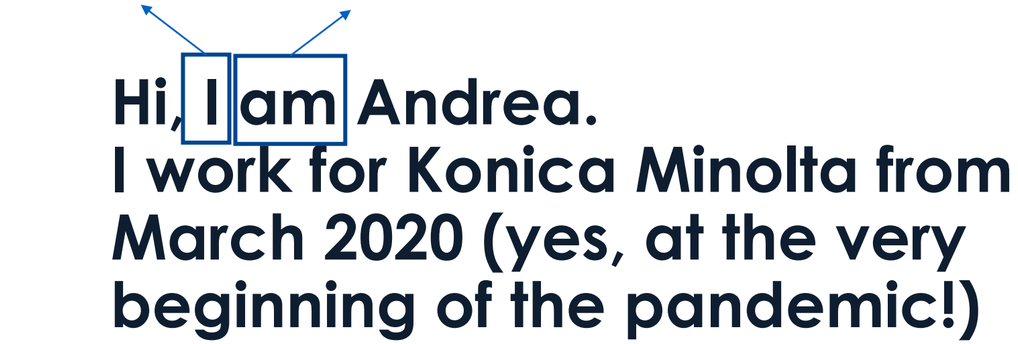

Image :from Chris McCormick

# Example

**Sequence-to-sequence**
What is the next sentence?




**Representation learning**
I'm a «subject»! I'm a «verb»!




**Self-attention**
I'm really linked to «Andrea» word!

# Question Answering on Digital Documents

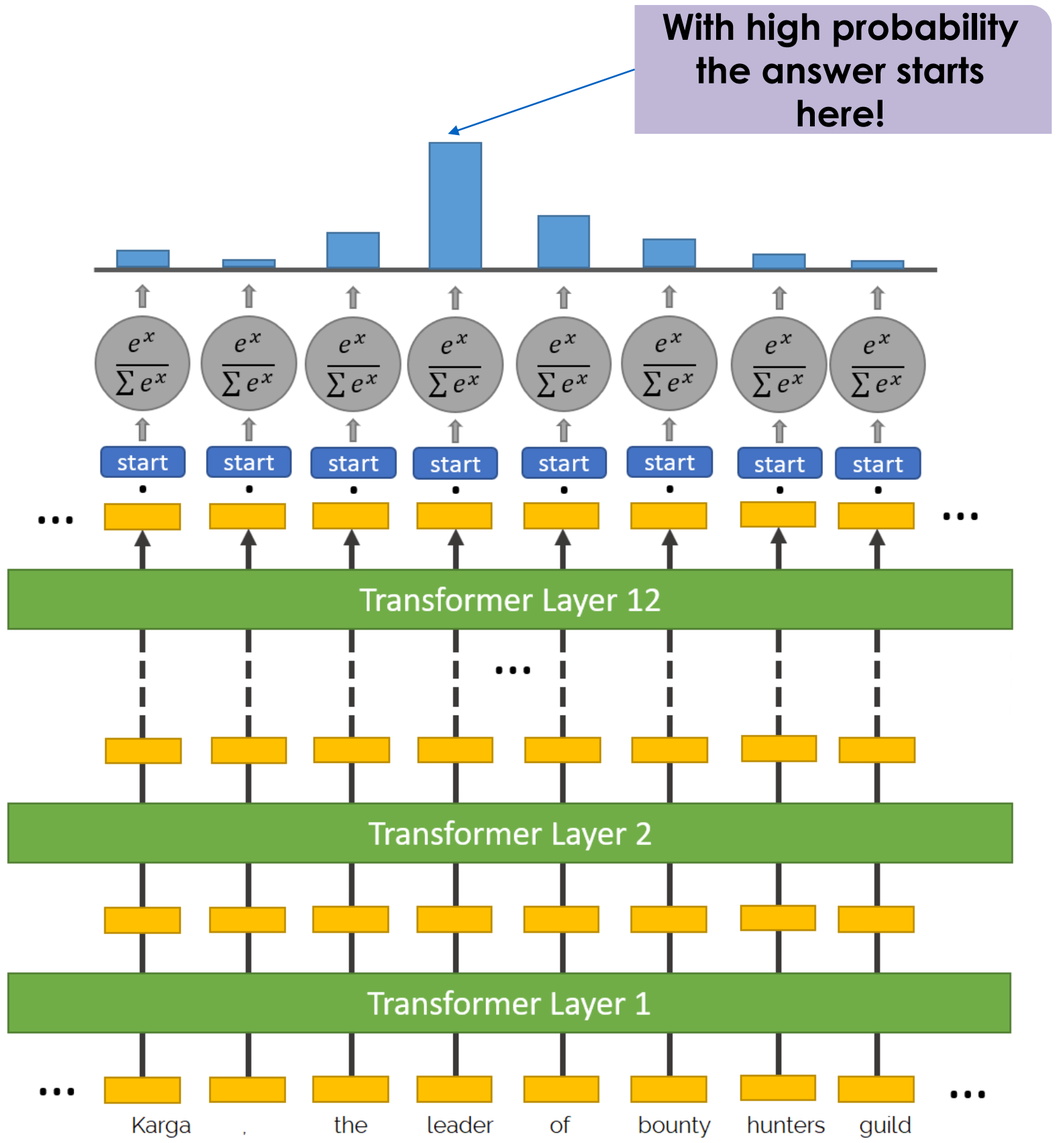
**Task:** given a question and a context, identify start – end span in the context that is the answer.

–**Question**: Who is Greef Karga?

–**Reference**: […] He meets Greef Karga, the **<answer_start>**leader of the bounty hunters guild**<answer_end>**, but he only offers low-paying bounties that will not cover travel expenses.

## General intuition:

- Put **Question** and **Reference** into BERT, separated by [SEP] tag
- Assign **Sentence A** to Question, **Sentence B** to Reference
- Train two classifiers: **Start token** and **End token** predictor using the softmax of the dot-product between last BERT hidden layer and the start/end weights value



# How to use Transfomers with Tensorflow.js

KONICA MINOLTA

- Fine-tune the BERT model on Q&A. Or use the already fine-tuned ☺

- Import from tensorflow-models

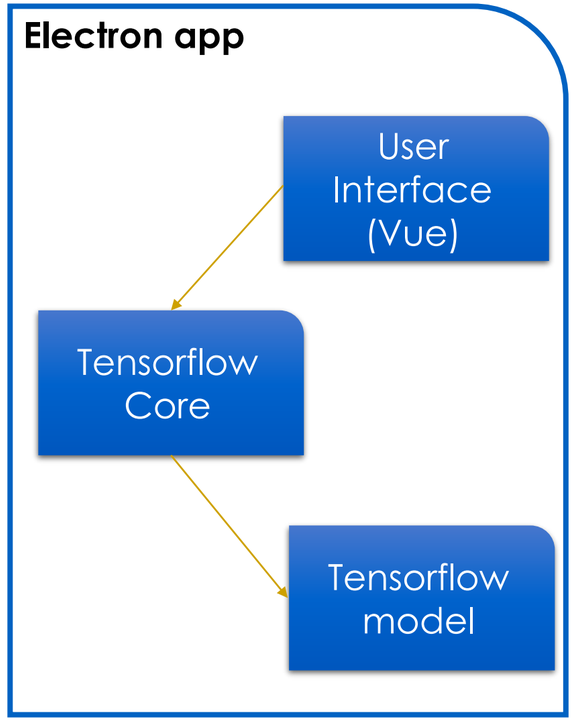
```
// install the peer dependencies for tfjs-core and tfjs-converter.

const qna = require('@tensorflow-models/qna');

// Load the model.
const model = await qna.load();

// Finding the answers
const answers = await model.findAnswers(question, passage);

console.log('Answers: ');
console.log(answers);
```



# Hands-on Lab

KONICA MINOLTA

Let's dive in

Prerequisites:

- Node.js (16 LTS)
- Git
- IDE (IntelliJ/VS code)

Start from this scaffold:

```
git clone git@github.com:spaghiajoeojo/floppy-qa.git
npm install
```


KONICA MINOLTA