# Deep Learning (IST, 2021-22)

# Practical 11: Word Embeddings and Large Pretrained Models

Taisiya Glushkova, Rita Ramos, André Martins, Ricardo Rei

## Question 1

In this question you are going to solve some analogy questions using static word embeddings.

1. Install the `torchtext` package. Download pre-trained GloVe vectors:

```
import torch
from torchtext.vocab import GloVe
glove = GloVe(name='6B', dim=50)
```

2. Compute the following word analogies using vector arithmetic. Provide top-5 closest vectors to each analogy:

   analogy('man', 'actor', 'woman')

   analogy('cat', 'kitten', 'dog')

   analogy('dog', 'puppy', 'cat')

   analogy('russia', 'moscow', 'france')

   analogy('obama', 'president', 'trump')

   analogy('rich', 'mansion', 'poor')

   analogy('elvis', 'rock', 'eminem')

   analogy('paper', 'newspaper', 'screen')

   analogy('monet', 'paint', 'michelangelo')

   analogy('beer', 'barley', 'wine')

   analogy('earth', 'moon', 'sun')

   analogy('house', 'roof', 'castle')

   analogy('building', 'architect', 'software')

   analogy('boston', 'bruins', 'phoenix')

   analogy('good', 'heaven', 'bad')

   analogy('jordan', 'basketball', 'woods')

   **Example:** analogy('king', 'man', 'queen')

   **Output:** [king - man + queen = ?]

   (2.8391) woman

(3.3545) girl

(3.9518) boy

(4.0233) her

(4.0554) herself

## Question 2

In this question you are going to experiment with large pretrained models using the Huggingface's `transformers` library.