AGH University of Science and Technology

# Seminar in *Artificial Intelligence*

## Word embedding

**Marcin Trebunia, Dominik Rygiel, Konrad Adamczyk**

Department of Telecommunications

27.05.2019

# Agenda

1. **Introduction**
   - Necessity for encoding text
   - Different types of encoding

# Introduction

What is word embedding?

# Encoding text

- Machine learning models take vectors (arrays of numbers) as input.
- ....

# Encoding text

What a **lovely** day.
What a **nice** day.

## One hot encoding

$$\begin{aligned}
\text{What} &= [1\ \ 0\ \ 0\ \ 0\ \ 0] \\
\text{a} &= [0\ \ 1\ \ 0\ \ 0\ \ 0] \\
\text{lovely} &= [0\ \ 0\ \ 1\ \ 0\ \ 0] \\
\text{nice} &= [0\ \ 0\ \ 0\ \ 1\ \ 0] \\
\text{day} &= [0\ \ 0\ \ 0\ \ 0\ \ 1]
\end{aligned}$$

# One hot encoding ctd.

- Words completely independent of each other
- Inefficient approach: vector is sparse

Source: .

# One hot encoding ctd.

Example:

- Dictionary of 10,000 words
- One hot encode each word
- Each vector's elements are 99.99% zeros!

# Unique number encoding

$$\begin{aligned}
\text{What} &= [1] \\
\text{a} &= [2] \\
\text{lovely} &= [3] \\
\text{nice} &= [4] \\
\text{day} &= [5]
\end{aligned}$$

# Unique number encoding ctd.

- + Efficient - dense vector
- – Encoding arbitrary - does not catch relationships between words
- – Can be challenging for a model to interpret

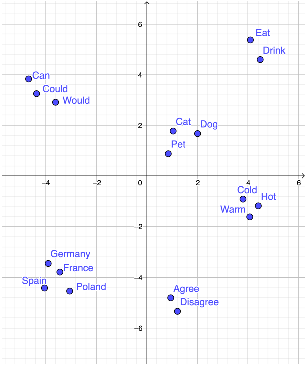
## Word embedding

$$\begin{array}{rcllllr} \text{What} & = & [1.2 & \text{-}0.1 & 4.3 & 3.2] \\ \text{a} & = & [0.4 & 2.5 & \text{-}0.9 & 0.5] \\ \text{lovely} & = & [2.1 & 0.3 & 0.1 & 0.4] \\ \text{nice} & = & [2.0 & 0.4 & 0.3 & 0.5] \\ \text{day} & = & [3.0 & \text{-}0.6 & 3.5 & \text{-}0.8] \end{array}$$

# Word embedding

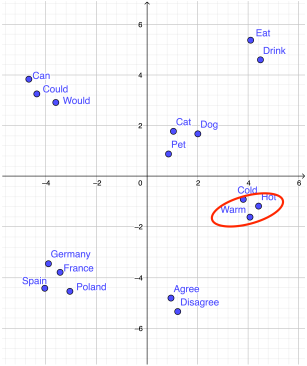
- Words with similar context occupy close spatial positions
- The cosine of the angle between words' vectors should be close to 1 (angle close to 0)

# Word Embedding



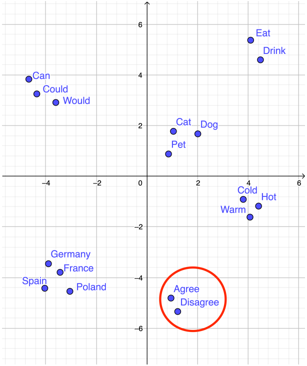
Caption of the figure

# Word Embedding



Words are synonyms

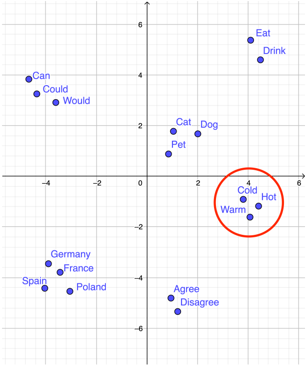
# Word Embedding



Words are antonyms

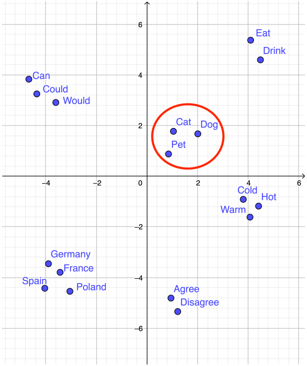
# Word Embedding

## Slide with a Figure from a File



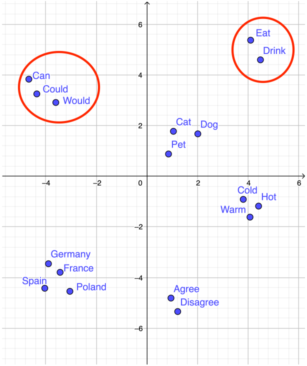
Words are value on a scale

# Word Embedding



Words are hyponym - hypernym

# Word Embedding



Words appear in similar context

# How can we use it?

- If user search for ”Dell notebook battery size” we would like to match it also with ”Dell laptop battery capacity”
- If user search for ”Cracow Motel” we would like to match it also with ”Krakow Hotel”

# Title of the Slide

## Subtitle of the Slide: Use Only if Necessary. . .

- Bulalet point 1.
- Bullet point 2 -e– you can emphasise a text.

# Title of the Slide

## Slide without Bullets

Text:

Blocks are good for important notions.

Alertblocks are even better to catch the attention.

**Title**

You can have the title in the (alert)block.

# Title of the Slide

## Slide with a Figure from a File

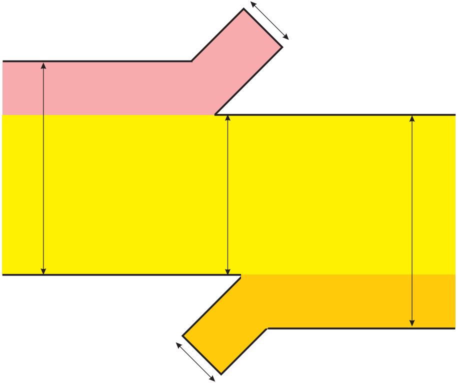
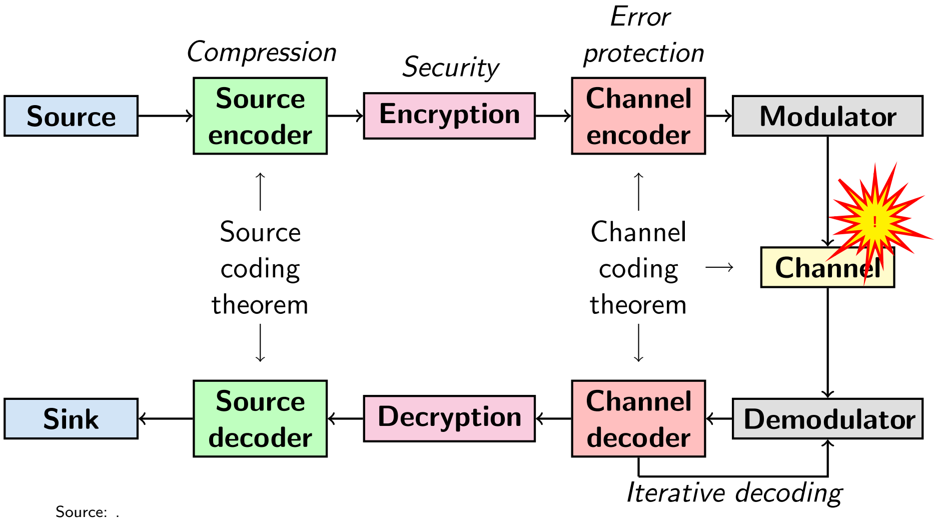


**Figure:** Caption of the figure

I am sorry that the example figures and the table concern the information theory ☺

# Title of the Slide

## Slide with a Figure Drawn with `tikz`



Source: .

# Title of the Slide

## Slide with a Table

**Table:** Caption of the Table

| Data | Entropy |
|---|---|
| Plain text | 4.347 |
| Native executables | 5.099 |
| Packed executables | 6.801 |
| Encrypted executables | 7.175 |

Source: .

# Thank you for your attention!

# Q & A

Intro to word embeddings.
https://www.tensorflow.org/alpha/tutorials/text/word_embeddings.
Accessed: 2019-05-11.

Robert Lyda and James Hamrock.
Using Entropy Analysis to Find Encrypted and Packed Malware.
*IEEE Security & Privacy*, 5(2):40–45, March/April 2007.

Todd K. Moon.
*Error Correction Coding*.
John Wiley & Sons, Inc., Hoboken, NJ, 2005.