Artificial Intelligence based Model for Ancient Text Analysis

Sergio Hernández González

Introduction

Previous study

Model building

Results

Future directions

Introduction

Rationale

Find unknown translations and adaptations from Ancient Language texts

Challenges

- Identify semantic relationships
- Text written in structurally different languages
- Low resource Ancient Languages

Steps

State of the art	Selection	Implementation
State of the art study on the field of NLP	Specific technique and tools selection	Coded implementations and model training

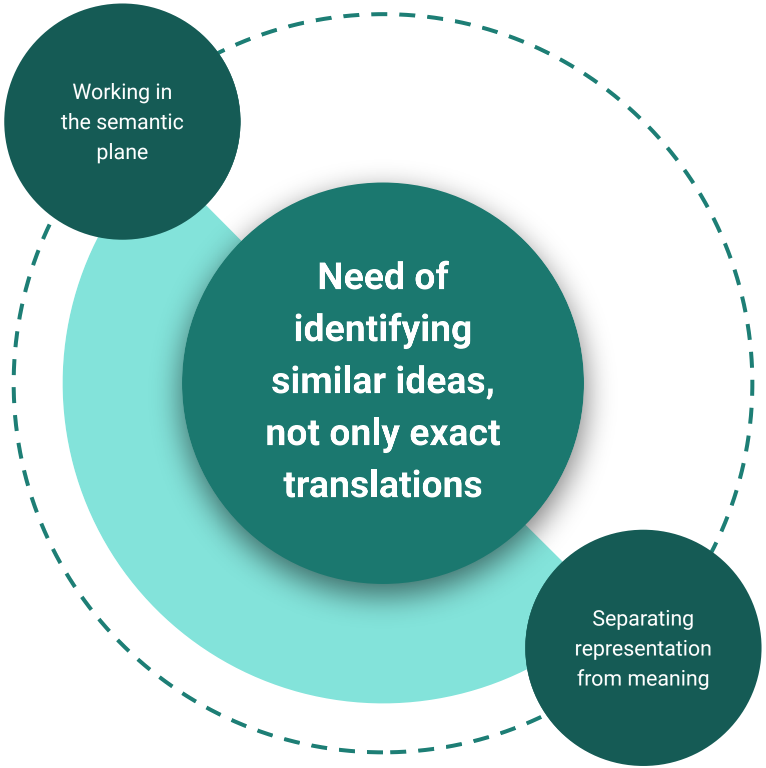
Previous study

Semantic similarity

Quantification of likeness or relatedness between texts

Based on meaning rather than surface form

We need a way to measure and compare meaning



Word embeddings

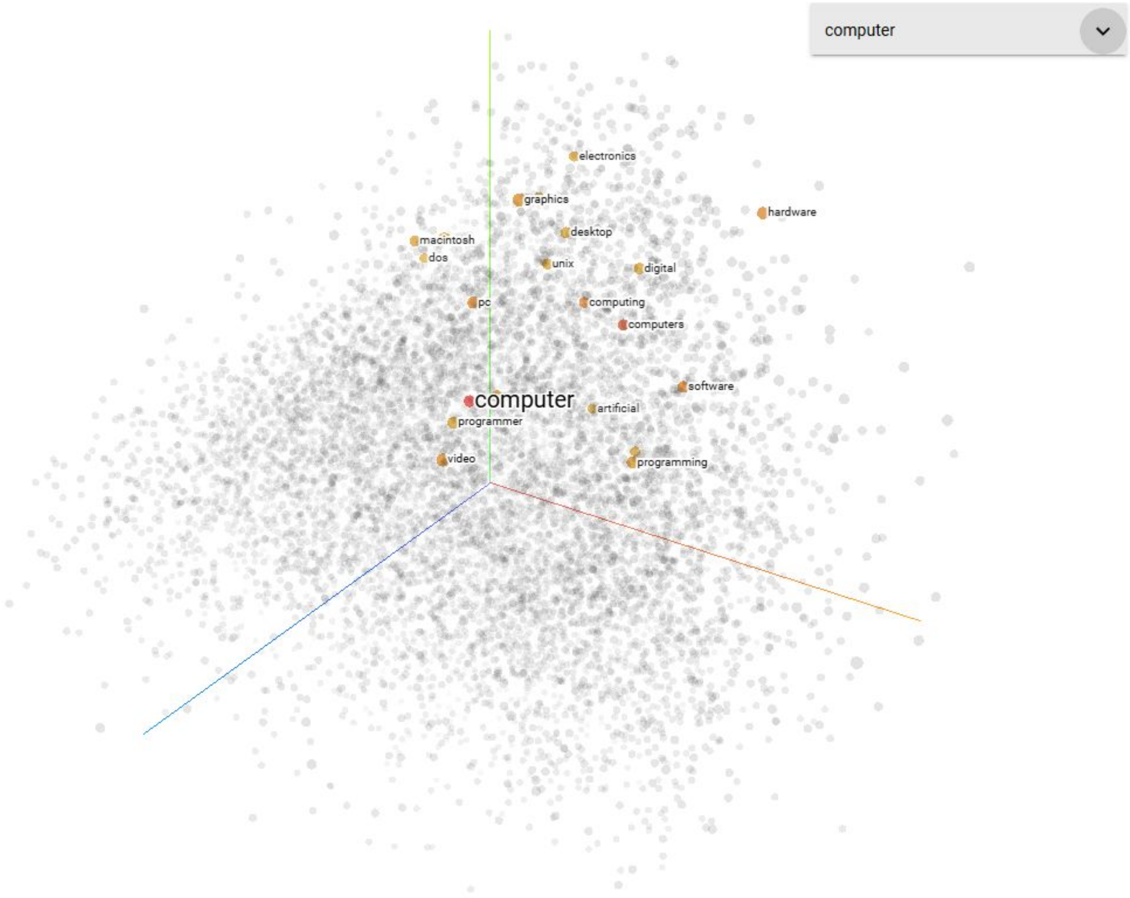
Numerical representation for a word's meaning

Dense vectors

Multidimensional vectors

Easy to handle for machines

Able to represent the reality in a multidimensional space



Word embeddings

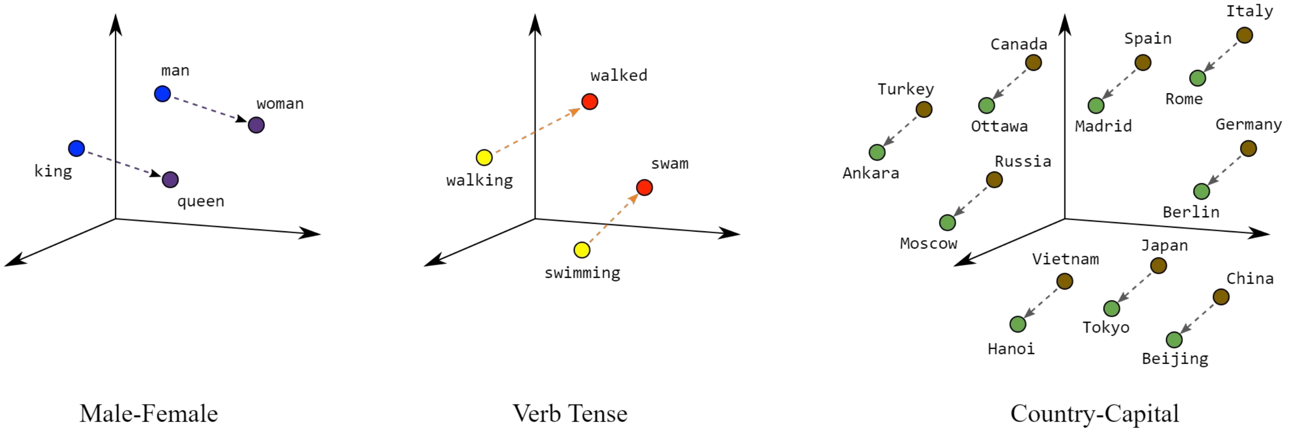


Image from: Google for developers ML course, https://developers.google.com/machine-learning/crash-course/embeddings/translating-to-a-lower-dimensional-space. Under Creative Commons 4.0 license. **CC BY 4.0**.

Word2Vec

Presented by Google in 2013

Neural Network based architecture

Learns from raw text, by examining the context where words appear

Relies on the hypothesis that neighboring words in text have semantic similarities with each other

https://projector.tensorflow.org/

Model building

Tools

python™

GENSIM
topic modelling for humans

NLTK

/regex/

Two models

Test case

English:

Easy evaluation

More resources

- Data
- Tools

Architecture

Target case

Middle High German (MHG)

Data collection

English:

Collection of 7.8 million sentences from the August 2018 English Wikipedia dump

MHG:

The “Reference Corpus of Middle High German” Corpus of diplomatically transcribed and annotated texts from Middle High German (1050-1350) with a size of around 2 million word forms

M001: Ad equum errehet	Normalisierter Lesetext
66_02,0 {10r,13}	[!] [!] erræhet.
66_02,1 {10r,13}	man gienc after wege,
66_02,2 {10r,14}	zôch sîn ros in handen.
66_02,3 {10r,14}	dô begegente ime mîn truhtîn
66_02,4 {10r,14}	mit sînere êrengrehte:
66_02,5 {10r,15}	“wes man gês dû?
66_02,6 {10r,15}	zuo iu ne rîtes dû?
66_02,7 {10r,15}	”waz mac ich rîten?
66_02,8 {10r,15}	mîn ros ist erræhet.
66_02,9 {10r,16}	“nû ziuch ez dâ bî viere,
66_02,10 {10r,16}	dû rûne ime in daz oere,
66_02,11 {10r,17}	trit ez an den zeswen vuoz;
66_02,12 {10r,17}	sô wirdet ime des erræhet buoz.
66_02,13 {10r,18}	[!] [!], [!] [!] [!] [!] [!] [!] [!]:”alsô schiere werde
66_02,14 {10r,19}	diseme, [!] [!] [!], rôt, swarz, blanc, vale, grîsel, vêch,
66_02,15 {10r,20}	rosse des erræheten buoz, same deme got dâ selbe buozte.

https://www.linguistics.ruhr-uni-bochum.de/rem/

Text processing

1. **Tokenization**
   Sentence and word tokenization
2. **Cleaning and normalization**
   Handling special characters and numbers lowercasing...
3. **Stopword removal**
   Removing common words with low semantic information
4. **Stemming and lemmatization**
   Transforming words to their root forms

Training the model

Word2Vec architecture

Provided by Gensim library

```python
from gensim.models import Word2Vec
```

Python

```python
model = Word2Vec(sentences=sentences, vector_size=200, window=4, min_count=1, workers=8)
```

Python

Results

Results: English

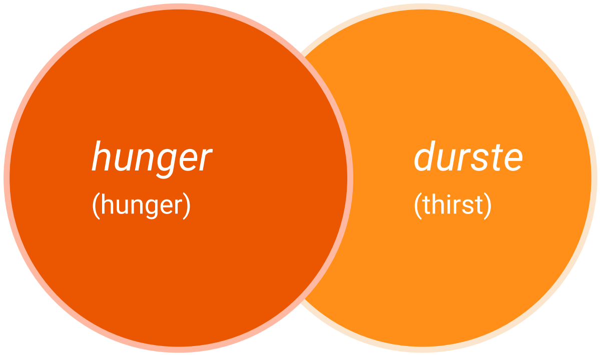
The model shows a good performance when compared to state of the art pre trained models'

Top most similar words to "efficiency":
"reliability", "durability", "responsiveness", "productivity", "effectiveness", …

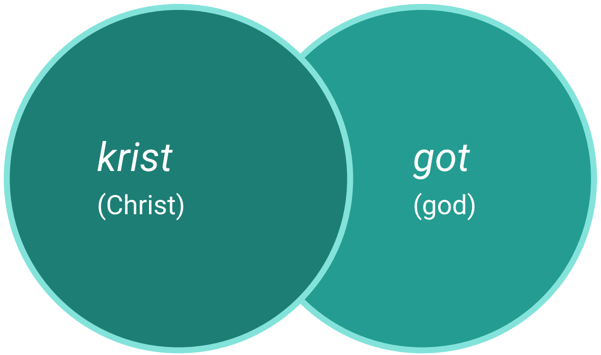
Similarity score for "computer" and "programming":
88%

Word that doesn't match from "dog", "cat", "monkey" and "car":
"car"

Results: MHG

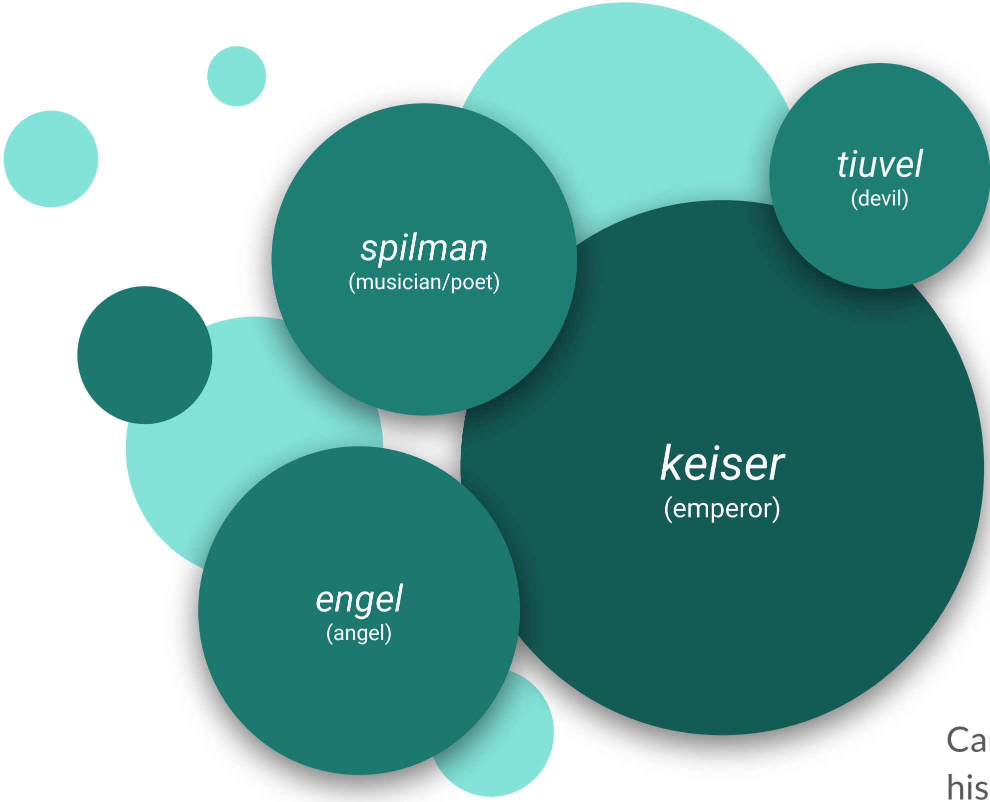


The model is capable of capturing words meaning



Results

Relationships that may seem inaccurate…



Can make sense from an historical perspective

Future directions

Future directions

- Improve current results
  - Address data scarcity
    - Collecting more data
    - Data augmentation
    - Transfer learning
  - Use specialized algorithms
- Explore new technologies
  - Contextual embeddings
  - Sentence embeddings
  - Vectorial spaces alignment

Questions

Thank you