03/02/2022
24/02/2022

AI Lab: Deep Network for Natural Language Processing

Leandro Paolo De Persiis
leandro.paolo.de.persiis@gmail.com
Master degree in psychology
Student AS-AI 2018-2019
Programmer Analyst at IAD

What is Natural Language Processing?

- Any technique aimed at processing of language which, in some cases, seems to simulate a certain understanding of the language itself.

- Texts Classification
- Sentiment Analysis
- Spam detection
- Information extraction
- Text Summary
- Question Answering
- Captioning
- Translation
- Dialog and Virtual Assistants

NLP Background

The first studies on NLP start from the dawn of computer science.

Symbolic level.

Problems:

- importance of the context,
- difficulty of grasping what is not said,
- implications,
- ambiguities,
- irony and sarcasm,
- triky names
- non-standard language.

NLP Background

Statistical approach.

Machine learning.

Big Data.

SVM, logistic regression and random forest, using scattered vectors as data.

NLP Background

Deep Learning.

Machine learning.

Big Data.

Deep neural network.

Word embeddings.

Text classification

- we have a text t
- a certain set of categories

$\mathrm{C} = \{\mathrm{c}_1, \mathrm{c}_2, \mathrm{c}_3, \ldots \mathrm{c}_n\}$

Task:

Identify the category c to witch t belongs

$$c \in C$$

Text classification

- $X = \{x_1, x_2, x_3, \ldots x_n\}$ (text documents)
- $Y = \{y_1, y_2, y_3, \ldots y_n\}$ (the labels)

$$Y = f(X)$$

Task:

Find f (a deep neural network and and its connections) such that the answers are exactly the ones we want

Assignment

Cataloging the different posts of a blog according to its category:

- Sport
- Politics
- Economics
- Philosopy
- Physics
- Literature
- Psychology

Pipeline

- Collecting and labeling data
- Preprocessing (lowercase, tokenize, remove not words, remove stop words, stemming)
- Build model
- Train
- Test, evaluate, adjust
- Save, use

Collecting and labeling data

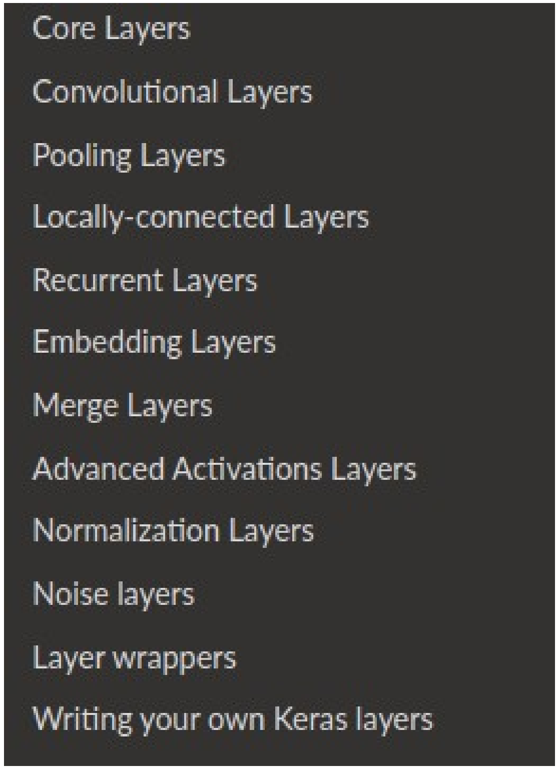
- Ready dataset – Provided to us, purchased or found a ready-made dataset that is right
- Search, where find your documents (wikipedia, newspapers, books, blogs ecc.)
- Write a bot for automatize
- Collect in files
- Produce one or more .csv (one document one record). One file for category or one file for all categories

Preprocessing

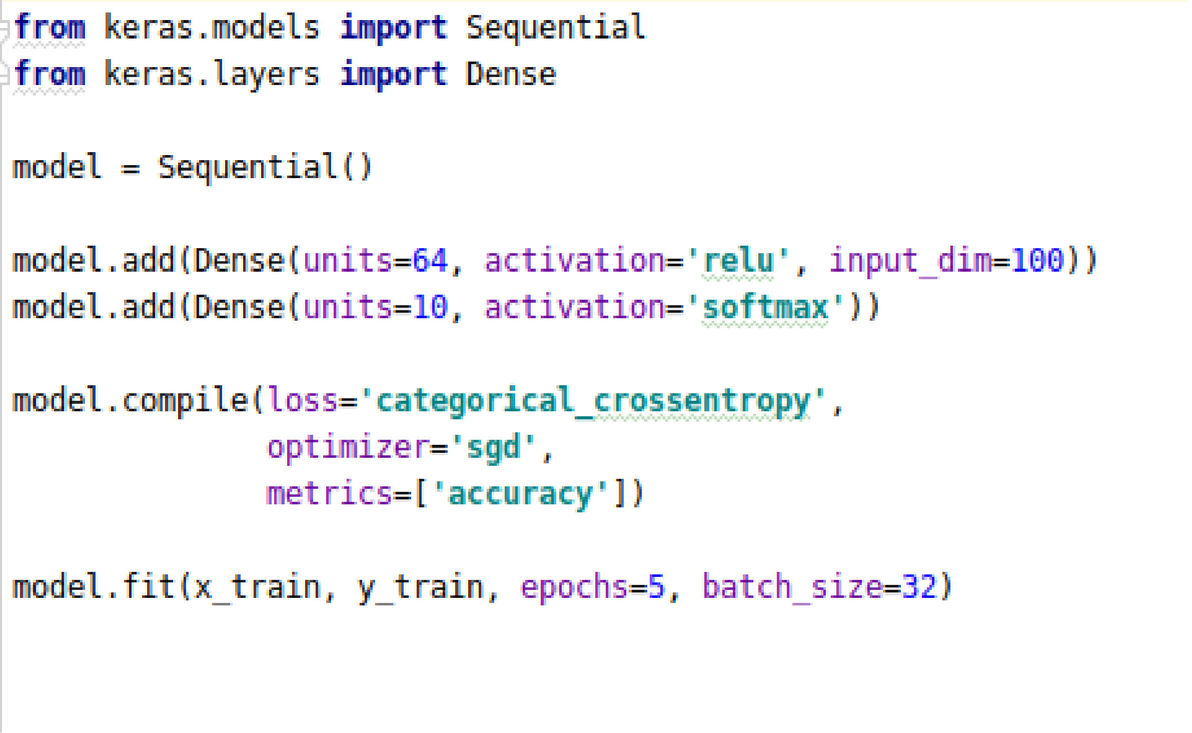
- Lowercasing
- Normalization
- Noise removal
- Tokenization
- Remove not words and stop words
- Lemmatization
- Stemming
- Convert to index
- Convert to vector

Build model

https://keras.io/



- Core Layers
- Convolutional Layers
- Pooling Layers
- Locally-connected Layers
- Recurrent Layers
- Embedding Layers
- Merge Layers
- Advanced Activations Layers
- Normalization Layers
- Noise layers
- Layer wrappers
- Writing your own Keras layers



```
from keras.models import Sequential
from keras.layers import Dense

model = Sequential()

model.add(Dense(units=64, activation='relu', input_dim=100))
model.add(Dense(units=10, activation='softmax'))

model.compile(loss='categorical_crossentropy',
              optimizer='sgd',
              metrics=['accuracy'])

model.fit(x_train, y_train, epochs=5, batch_size=32)
```

One-hot

This is an example sentence with a few words and a single word repeated

['this', 'is', 'an', 'example', 'sentence', 'with', 'a', 'few', 'words', 'and', 'a', 'single', 'word', 'repeated']

['example', 'sentence', 'words', 'single', 'word', 'repeated']

['example', 'sentence', 'word', 'single', 'word', 'repeat']

[1, 2, 3, 4, 3, 5]

example = [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

sentence = [0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

word = [0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

single = [0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

word = [0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

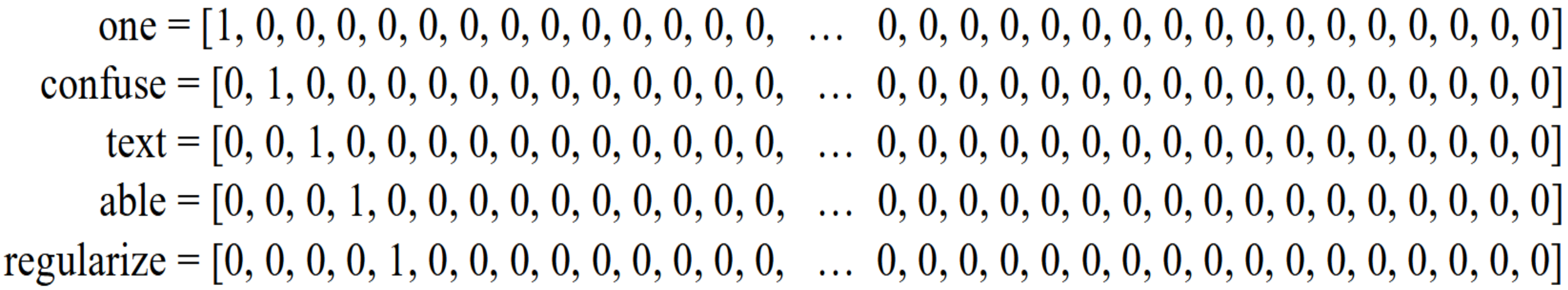
repeat = [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0]

....

... = [0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0]

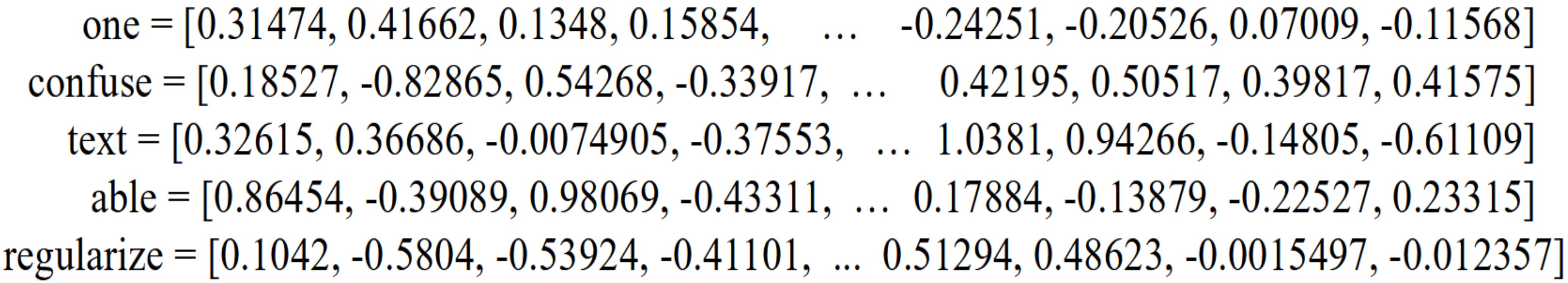
... = [0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0]

Embeddings



```
one = [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,   …   0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
confuse = [0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,   …  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
text = [0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,   …  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
able = [0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,   …  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
regularize = [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,   …  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

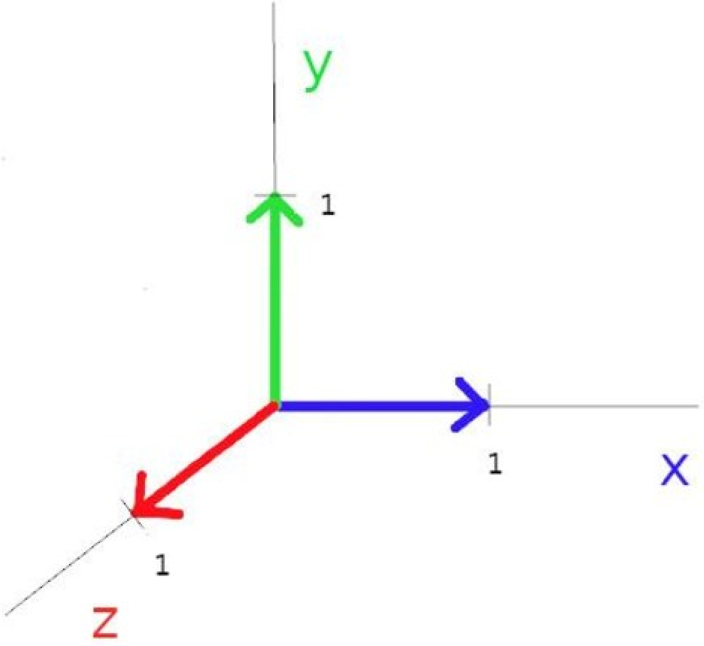
400.000 dimensions

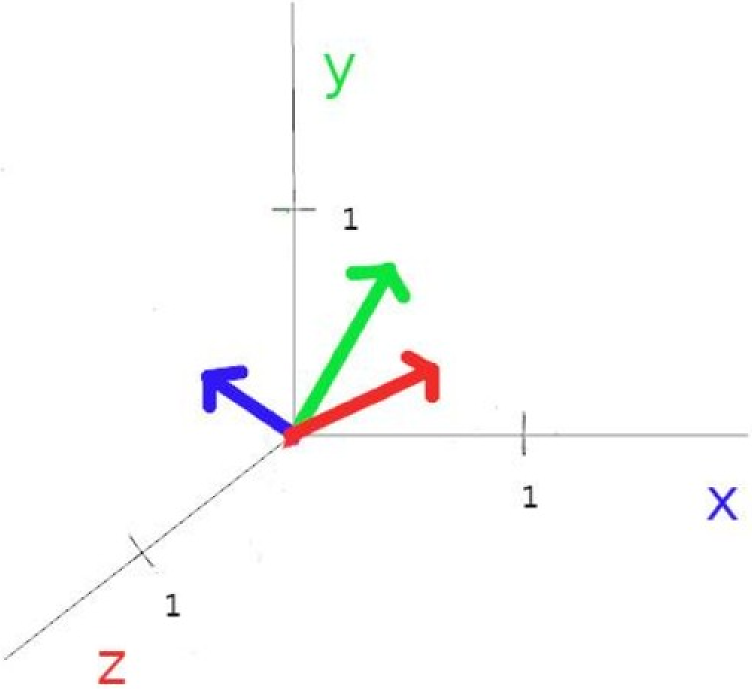


```
one = [0.31474, 0.41662, 0.1348, 0.15854,     …    -0.24251, -0.20526, 0.07009, -0.11568]
confuse = [0.18527, -0.82865, 0.54268, -0.33917,  …     0.42195, 0.50517, 0.39817, 0.41575]
text = [0.32615, 0.36686, -0.0074905, -0.37553,   …  1.0381, 0.94266, -0.14805, -0.61109]
able = [0.86454, -0.39089, 0.98069, -0.43311,  …  0.17884, -0.13879, -0.22527, 0.23315]
regularize = [0.1042, -0.5804, -0.53924, -0.41101,  ...  0.51294, 0.48623, -0.0015497, -0.012357]
```

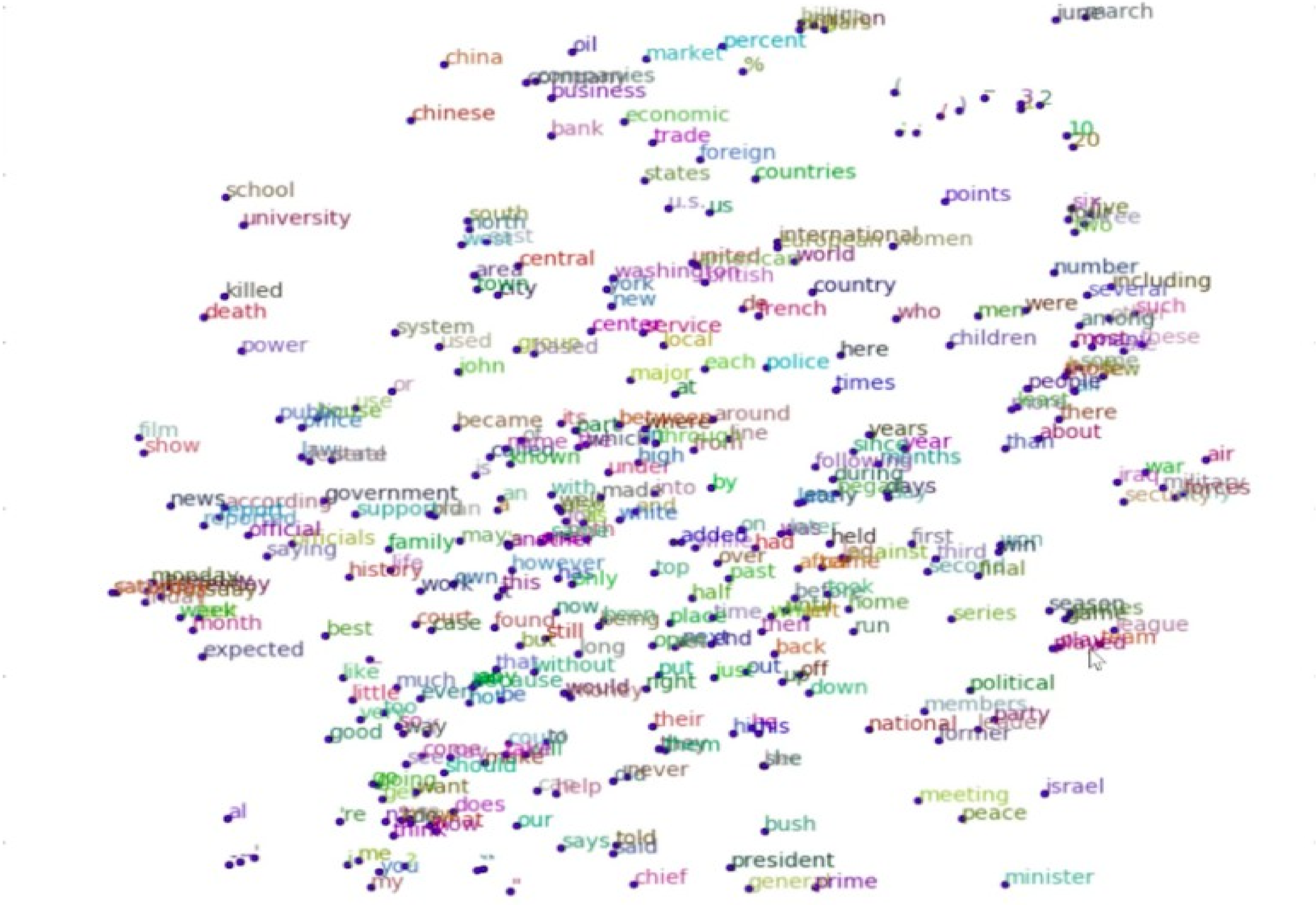
100 dimensions

Embeddings



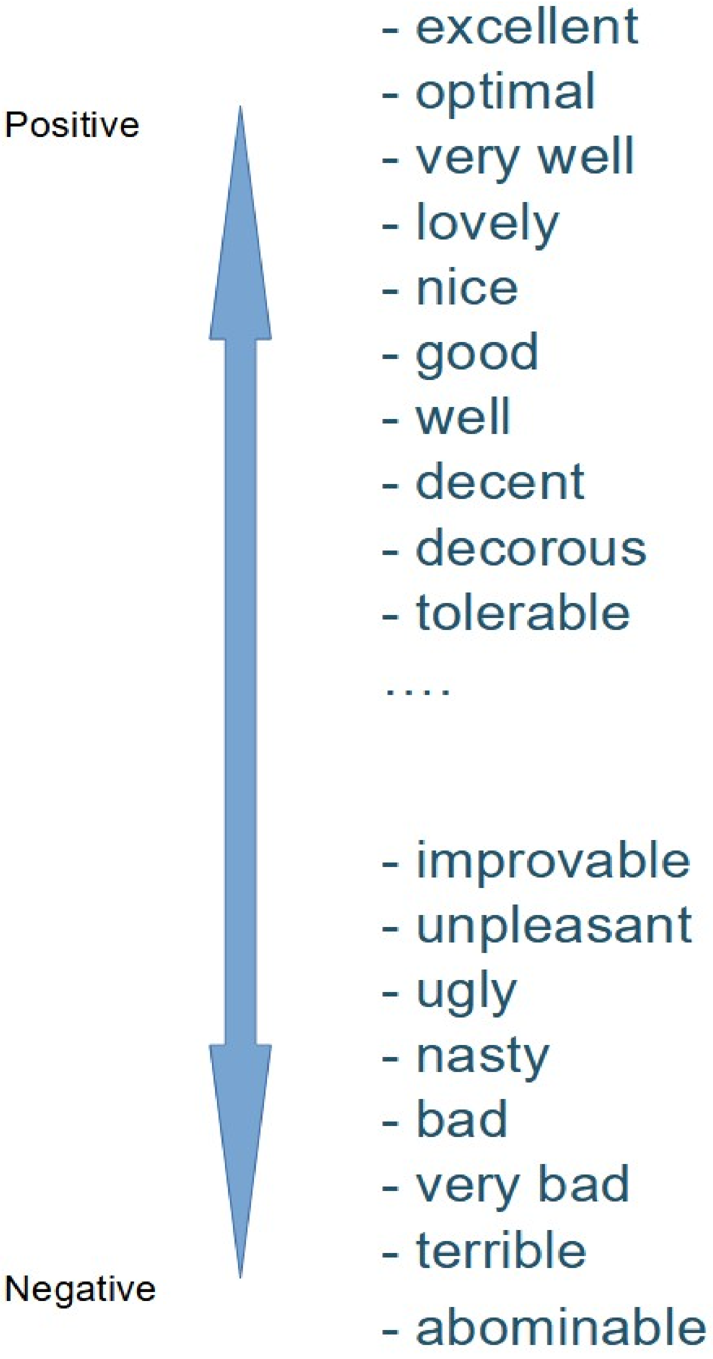


Embeddings



billion
june
march
percent
oil
market
china
%
companies
business
chinese
economic
bank
trade
10
20
foreign
states
countries
school
university
south
north
west
east
points
six
five
three
two
u.s.
us
international
european
women
united
american
world
central
area
washington
british
number
including
killed
death
town
city
york
new
country
several
french
who
men
were
such
among
system
center
service
local
children
most
these
power
used
group
based
here
john
major
each
police
at
times
people
or
public
office
became
its
part
between
where
around
film
show
which
through
line
years
since
year
about
than
there
known
high
under
following
months
during
war
air
news
according
government
with
made
into
by
days
iraq
military
forces
security
report
reported
support
official
officials
family
may
another
added
on
held
first
won
saying
life
over
had
against
third
history
however
has
top
past
second
final
work
own
this
only
half
before
took
home
week
month
now
being
place
time
then
series
season
games
league
best
court
case
found
but
still
long
open
end
run
back
team
played
expected
that
without
just
out
off
like
much
because
put
up
down
political
little
ever
would
right
members
party
good
way
their
his
national
former
leader
come
should
make
they
them
she
never
going
want
help
meeting
israel
al
does
our
peace
think
says
told
said
bush
me
you
my
president
chief
general
prime
minister

Embeddings

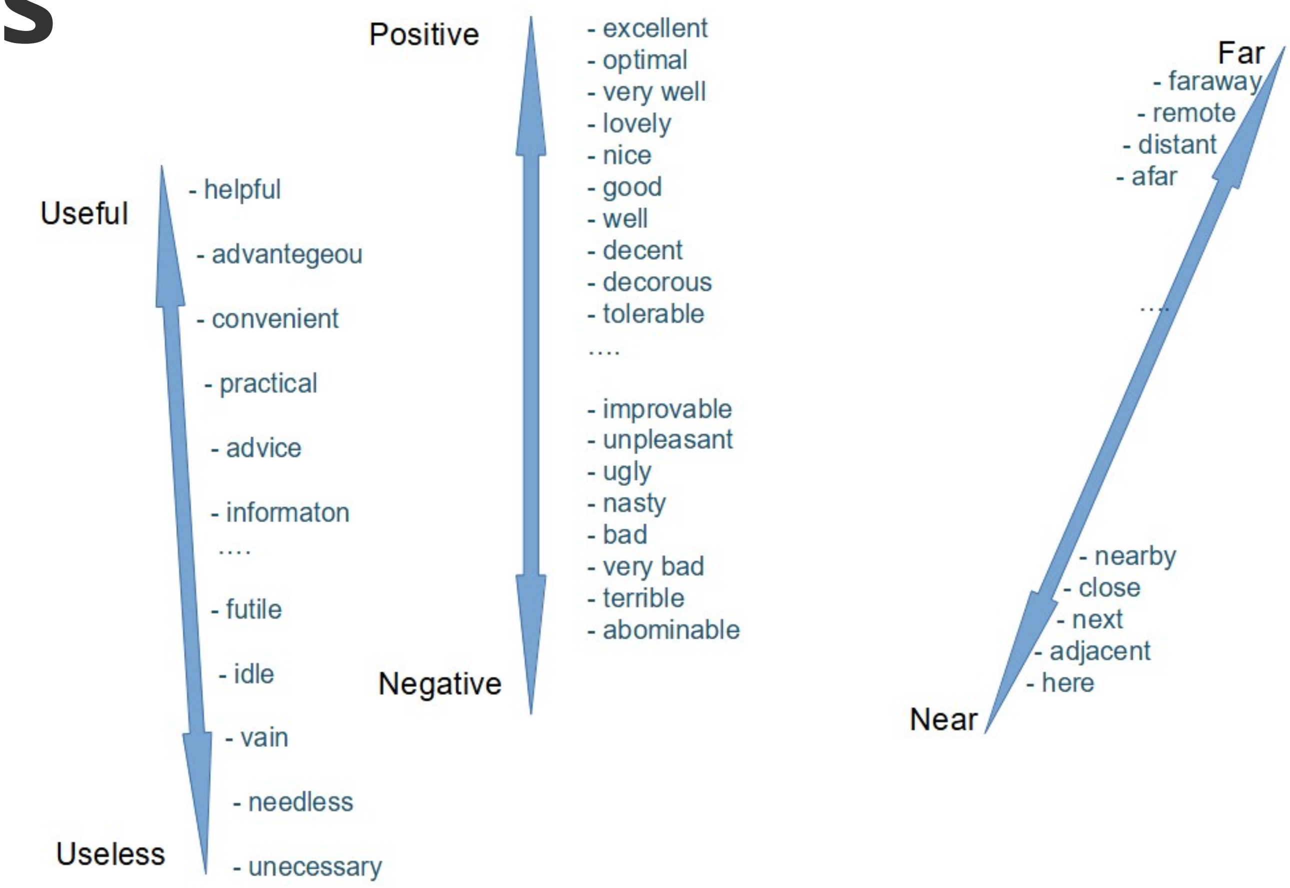


- excellent
- optimal
- very well
- lovely
- nice
- good
- well
- decent
- decorous
- tolerable

....

- improvable
- unpleasant
- ugly
- nasty
- bad
- very bad
- terrible
- abominable

Embeddings



Useful

- helpful
- advantegeou
- convenient
- practical
- advice
- informaton

….

- futile
- idle
- vain
- needless
- unecessary

Useless

Positive

- excellent
- optimal
- very well
- lovely
- nice
- good
- well
- decent
- decorous
- tolerable

….

- improvable
- unpleasant
- ugly
- nasty
- bad
- very bad
- terrible
- abominable

Negative

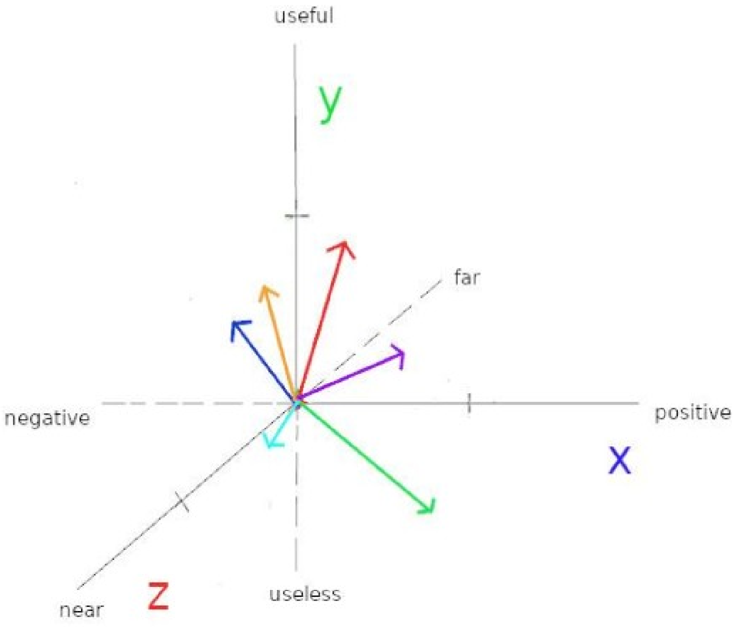
Far

- faraway
- remote
- distant
- afar

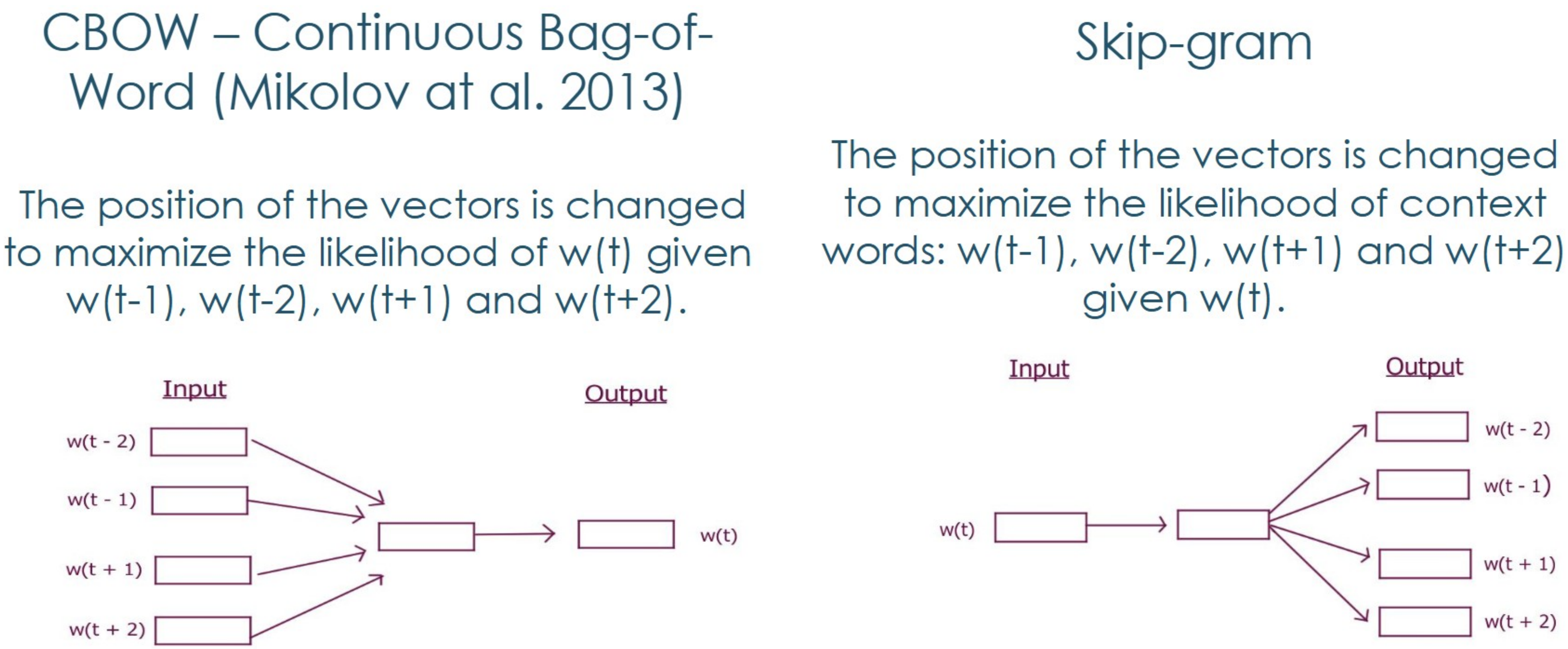
….

- nearby
- close
- next
- adjacent
- here

Near



Word2vec



CBOW – Continuous Bag-of-Word (Mikolov at al. 2013)

The position of the vectors is changed to maximize the likelihood of w(t) given w(t-1), w(t-2), w(t+1) and w(t+2).

Skip-gram

The position of the vectors is changed to maximize the likelihood of context words: w(t-1), w(t-2), w(t+1) and w(t+2) given w(t).

GloVe

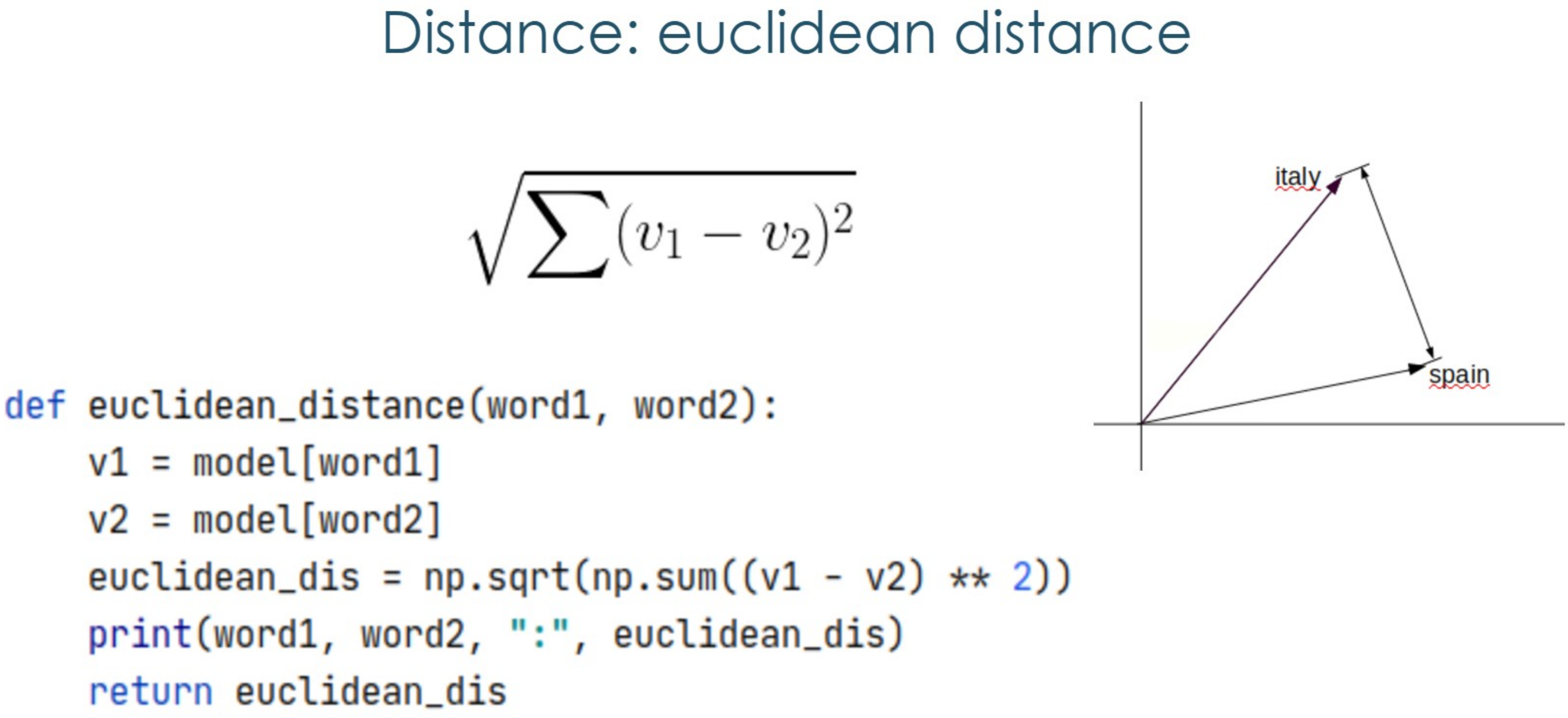
(Jeffrey Pennington, Richard Socher, and Christopher D. Manning 2014)

Weighted least squares

Train on global word-word co-occurrence counts

Minimizes the difference between the dotproduct of the vector of a word and its context word

Evalutation methods



Distance: euclidean distance

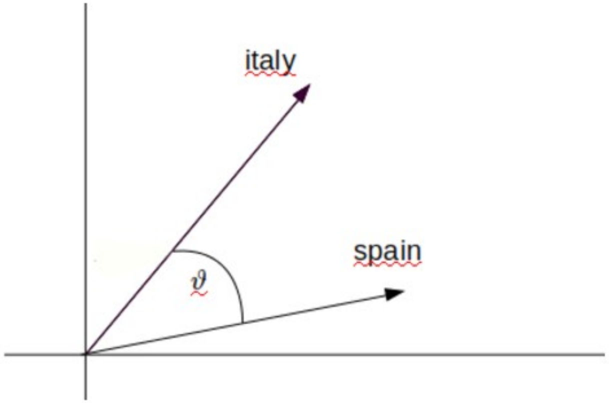
$$\sqrt{\sum (v_1 - v_2)^2}$$

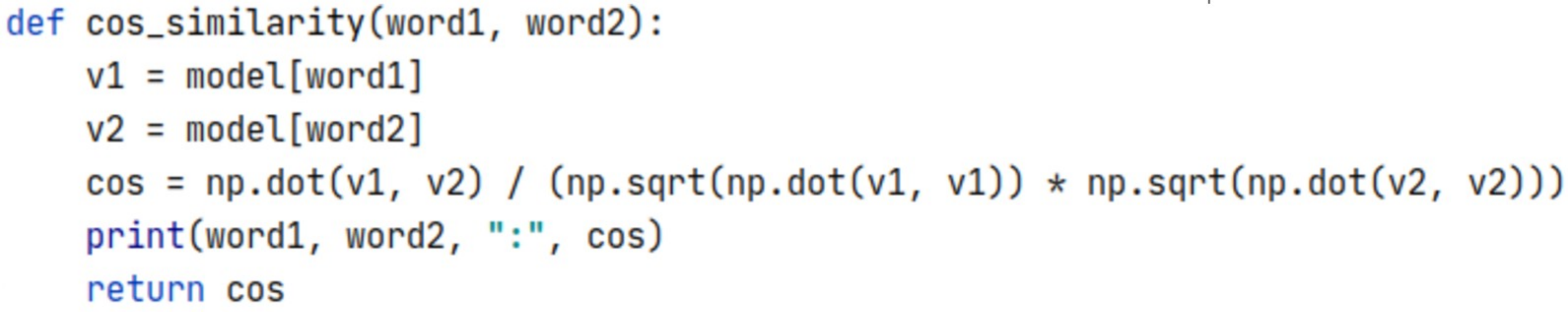
```
def euclidean_distance(word1, word2):
    v1 = model[word1]
    v2 = model[word2]
    euclidean_dis = np.sqrt(np.sum((v1 - v2) ** 2))
    print(word1, word2, ":", euclidean_dis)
    return euclidean_dis
```

Evalutation methods

Distance: angle between pairs of word vectors

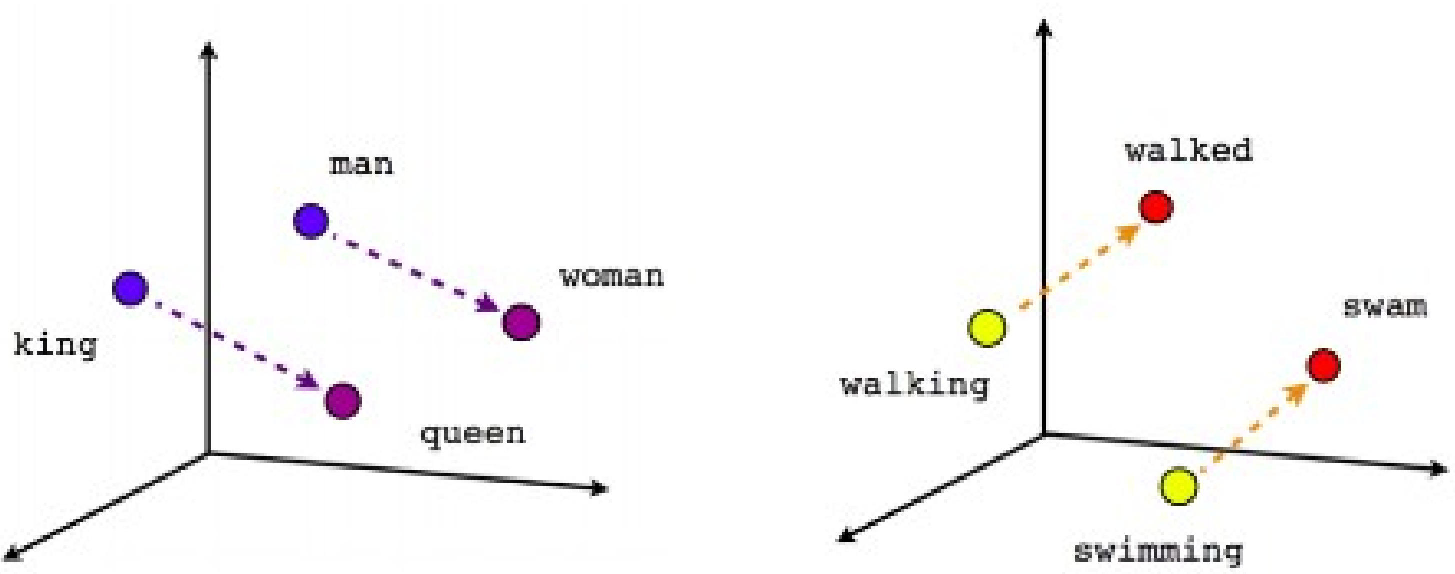
$$\cos(\vartheta) = \frac{v_1.v_2}{\|v_1\|_2 \, \|v_2\|_2}$$





```
def cos_similarity(word1, word2):
    v1 = model[word1]
    v2 = model[word2]
    cos = np.dot(v1, v2) / (np.sqrt(np.dot(v1, v1)) * np.sqrt(np.dot(v2, v2)))
    print(word1, word2, ":", cos)
    return cos
```

Embeddings similarities



Pipeline

- ~~Collecting and labeling data~~
- ~~Preprocessing (lowercase, tokenize, remove not words, remove stop words, stemming)~~
- ~~Build model~~
- Train
- Test, evaluate, adjust
- Save, use

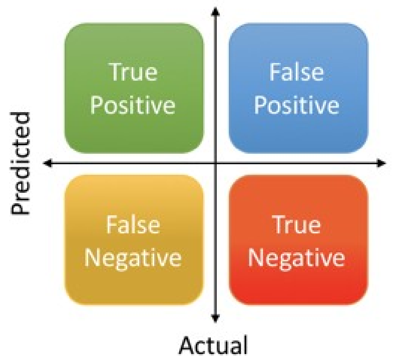
Test, evaluate

- Accuracy
- Precision
- Recall
- F1 score

$$\textbf{Precision} = \frac{\text{True Positive}}{\text{Actual Results}} \text{ or } \frac{\text{True Positive}}{\text{True Positive + False Positive}}$$

$$\textbf{Recall} = \frac{\text{True Positive}}{\text{Predicted Results}} \text{ or } \frac{\text{True Positive}}{\text{True Positive + False Negative}}$$

$$\textbf{Accuracy} = \frac{\text{True Positive + True Negative}}{\text{Total}}$$



$$\textbf{F1 Score} = 2 * \frac{\text{Precision * Recall}}{\text{Precision + Recall}}$$

Character based neural language model

Strengths and Weaknesses

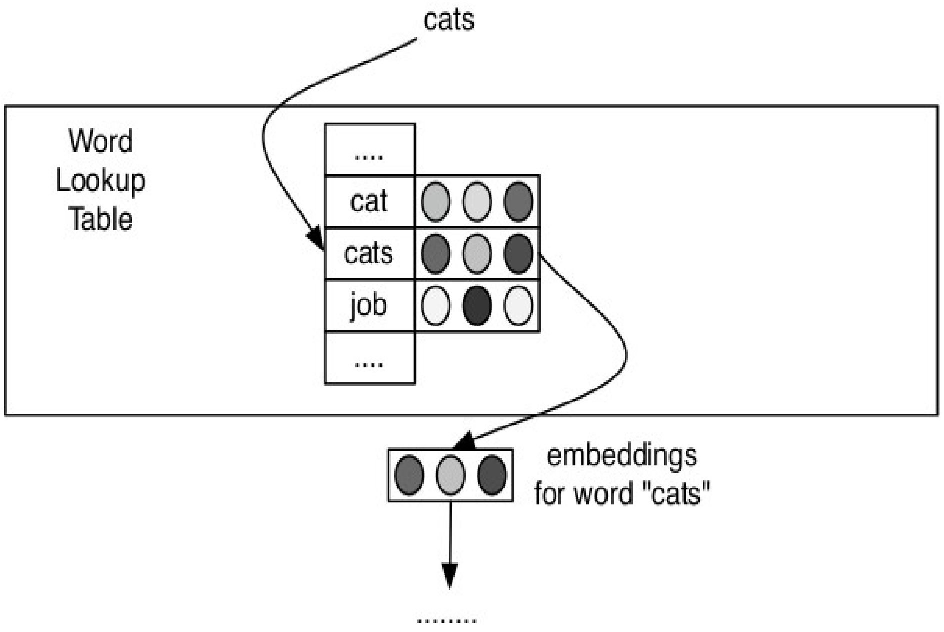
Fors:

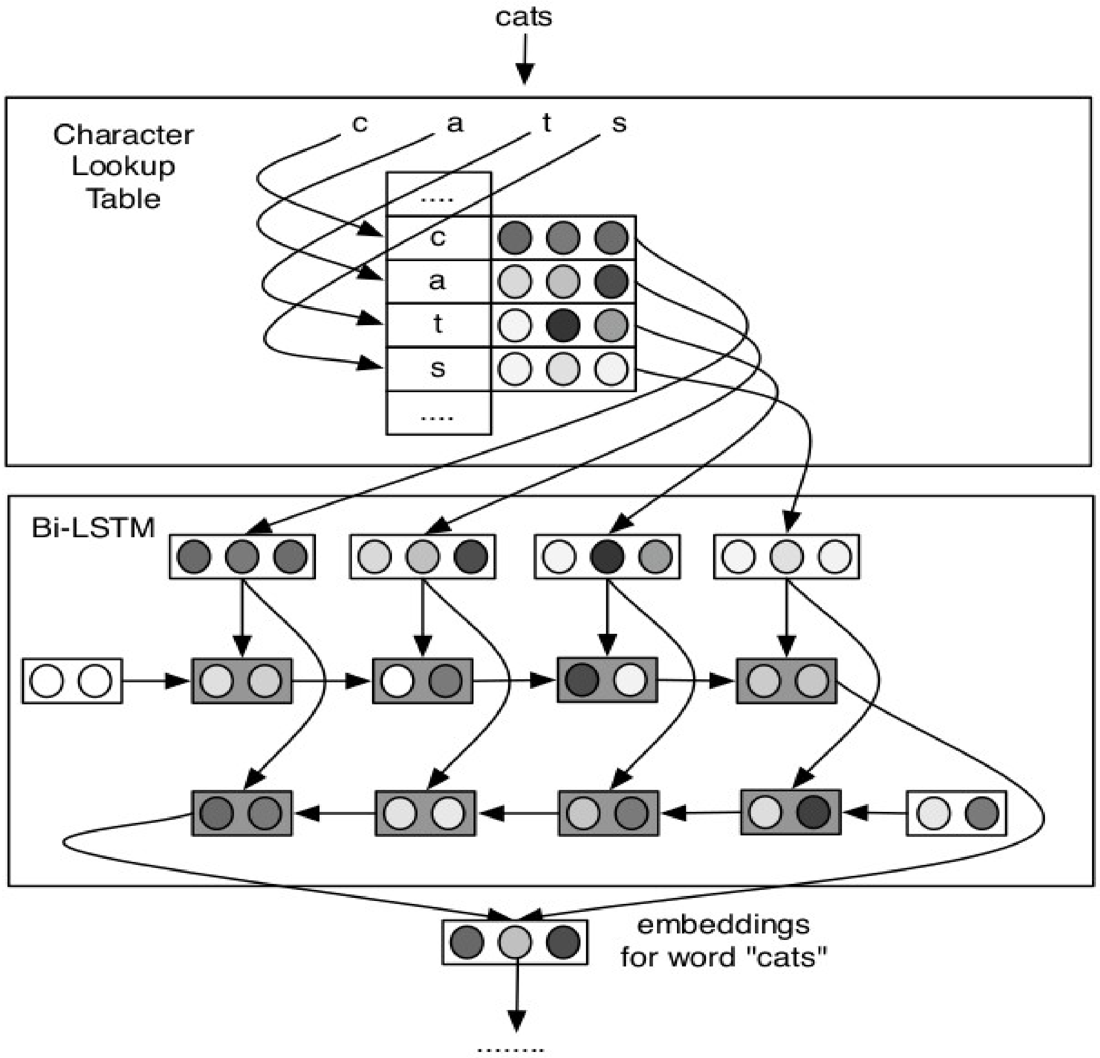
- Smaller first layer of the Neural network,
- The problem of words not present in the vocabulary is overcome,
- It is not necessary to divide correctly into words,
- The best choice for morphologically very rich languages.

Againsts:

- Very slow LSTM networks
- Give up the previously seen benefits of word embedding

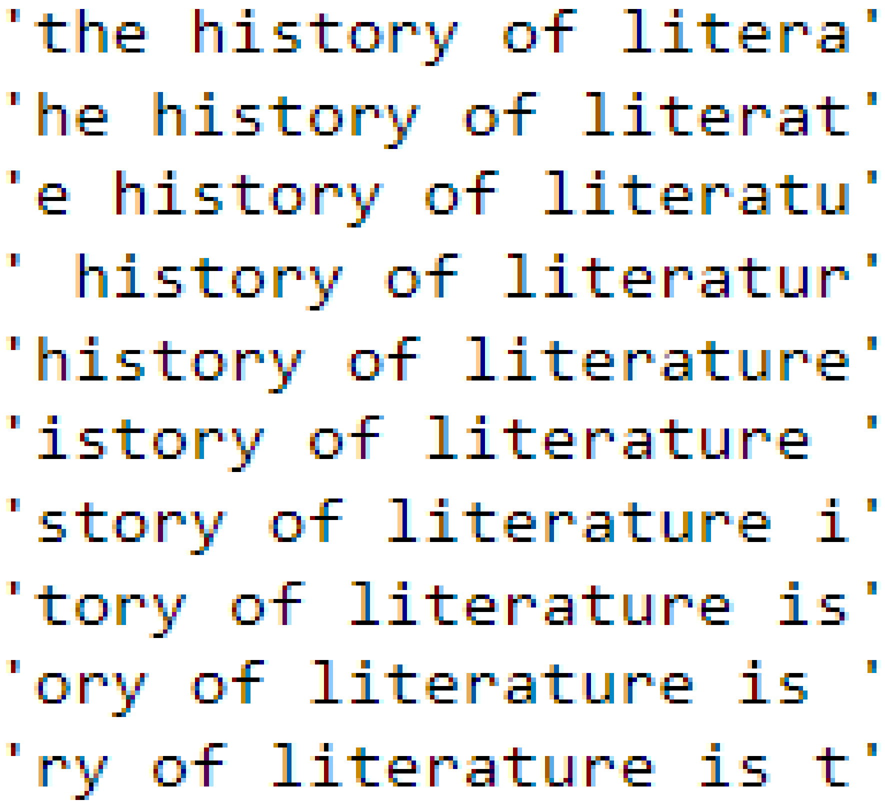
Structure and operation





Images from https://arxiv.org/abs/1508.02096

Sequences example



```
'the history of litera'
'he history of literat'
'e history of literatu'
' history of literatur'
'history of literature'
'istory of literature '
'story of literature i'
'tory of literature is'
'ory of literature is '
'ry of literature is t'
```

Character-Based Neural Language Model Applications

- **Translations**
  Several studies have shown the same or better performance at the state of the art, particularly when morphologically rich or mixed-language. languages are also involved.

- **Word completion and natural language correction**
  It is the most congenial field to this technique, several studies demonstrate the effectiveness of language models obtained in this way in the completion of words and in the correction of what is typed.

- **Named Entity recognition**
  This technique has also been used successfully in this area, particularly in the medical field.

- **Classification and clustering**
  In particular areas with many non-regular terms and mixed-language (e.g. twitter).

- **Text generators**
  For different types of texts also for rhyming verses.

- **Speech recognition**
  Character-based models are especially good at predicting novel word forms that were not seen in the training data.

Thanks for the attention

leandro.paolo.de.persiis@gmail.com