UNIVERSITÄT BONN

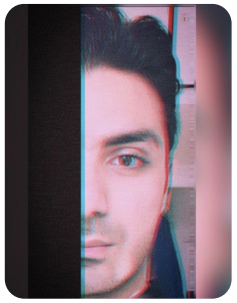
CAISA Lab
https://caisa-lab.github.io/

b-it
Bonn-Aachen
International Center for
Information Technology

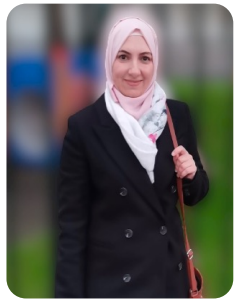
# INTRODUCTION TO NATURAL LANGUAGE PROCESSING

# INSTRUCTORS & TEACHING ASSISTANTS


Prof. Dr. Lucie Flek
Head of CAISA Lab
flek@bit.uni-bonn.de



Vahid Sadiri Javadi
Course Coordinator
vahidsj@bit.uni-bonn.de



Farizeh Aldabbas
Teaching Assistant
farizeh@uni-bonn.de

Ulvi Shukurzade
Teaching Assistant
ulvi@uni-bonn.de


Lectures: **Thursday** 10:15 – 11:45 (B-IT-Max 0.109) (Zoom Link)

Exercises: **Wednesday** **- Group 1 (TA: Vahid):** 14:15 - 15:45 (B-IT-Max 0.109) (Zoom Link)

**- Group 2 (TA: Ulvi):** 16:00 - 17:30 (B-IT-Max 0.109) (Zoom Link)

eCampus Course

# ANNOUNCEMENT

## Announcements:

**- Assignment #2** ✅

- Received 14 submissions until Monday 11:59 PM
- Received 39 submissions until Tuesday 11:59 PM

**- Assignment #3**

**- Deadline:** In two weeks, Tuesday, Dec 5th, 23:59

**- Submission of Problem Formulation (PF):**

**- Deadline:** Tuesday, Nov 28th, 23:59
**- Guideline:** eCampus >> Project >> Problem Formulation (PF) - Guidelines
**- Submission:**
  **- What:** PDF
  **- Where:** eCampus >> Project >> Problem Formulation (PF) - Submissions >> File name: **Team_<num.>**

**- ONLY** one of the team members should upload it!

# ANNOUNCEMENT

Announcements:

**- Q & A:**

- Next week, I'll be here at 2:15 PM to answer your questions about PF & PS.
- Please prepare your questions.

**- Overleaf Templates:**

- [Default Project]

- [Resource Creation Project]

- [Robustness and Reproducibility Project]

# COURSE OUTLINE

## Content of Course:

**Week 1:** 25.10.2023 | Introduction & Python basics

**Feature Engineering:**

**Week 3:** 08.11.2023 | Word operations & Feature extraction using Pandas, Sklearn

**Week 4:** 15.11.2023 | Linear classification using TF - IDF

**Language Processing:**

**Week 5: 22.11.2023 | Word embeddings using spaCy**

**Week 6: 29.11.2023 | Q & A: PF + PS**

**Week 7:** 06.12.2023 | Transformers and Generative Models I

**Week 8:** 13.12.2023 | Transformers and Generative Models II

**Week 9:** 20.12.2023 | POS tagging & HMMs

**Week 10:** 10.01.2024 | Project development (supervision by appointment)

**Week 11:** 17.01.2024 | Project development (supervision by appointment)

**Week 12:** 24.01.2024 | Project development (supervision by appointment)

**Week 13: 31.01.2024 | PROJECT PRESENTATIONS (PP)**

# AGENDA

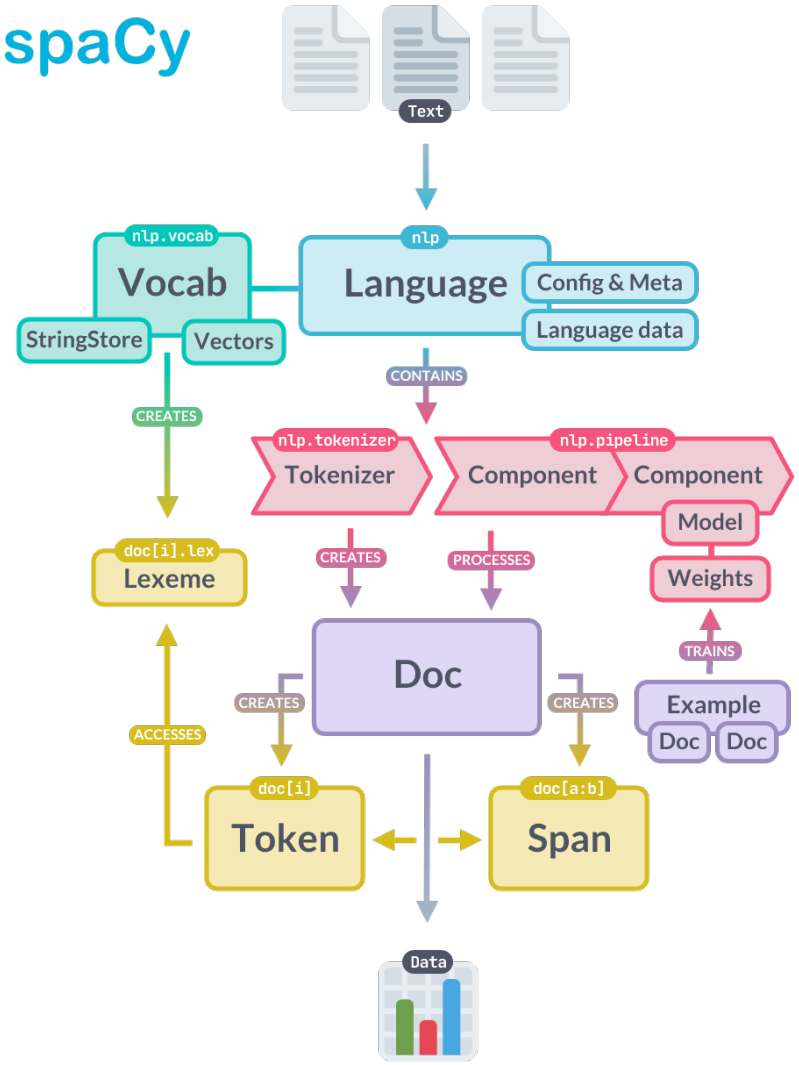
Today, we will talk about:

- **Assignment #2**
- **Word Embedding**

# FEATURE EXTRACTION

## using SpaCy

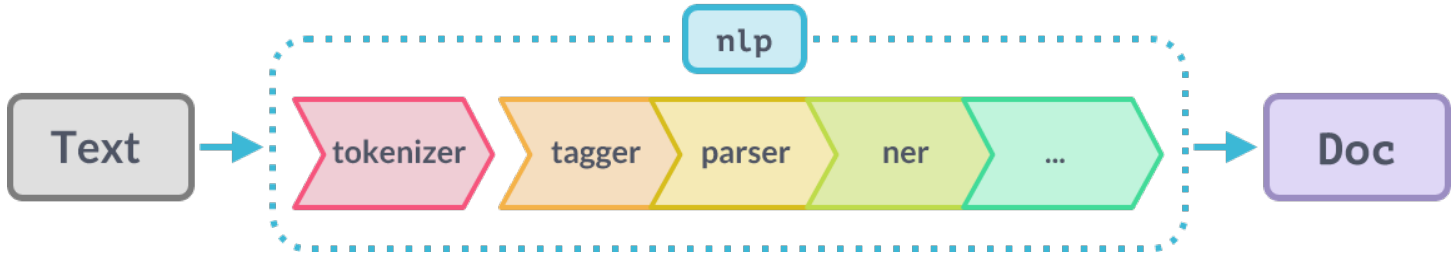
# SpaCy INTERNAL STRUCTURE




What's inside?

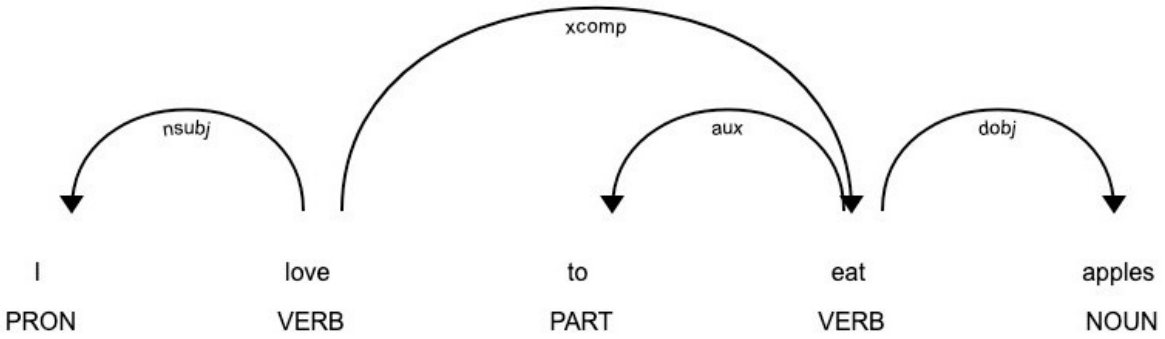
# SpaCy INTERNAL STRUCTURE

## What is a pipeline?

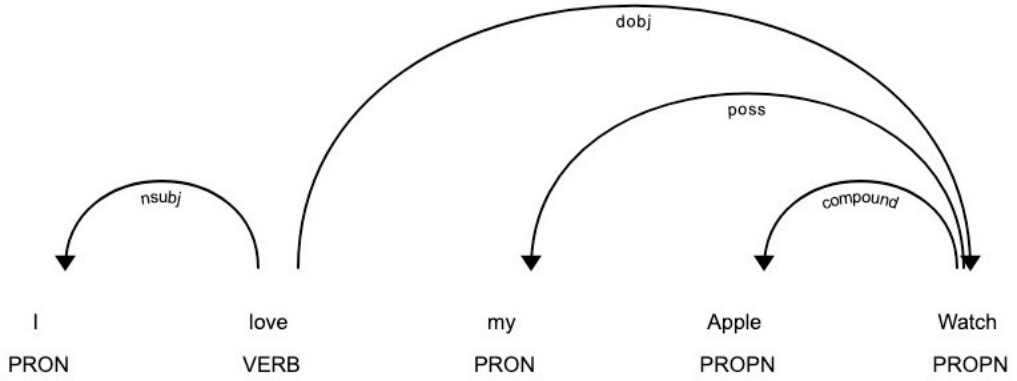


# DEPENDENCIES

I love to eat apples




I love my Apple watch



# DEPENDENCIES & POS CATEGORIES

| Clausal Argument Relations | Description |
|---|---|
| NSUBJ | Nominal subject |
| DOBJ | Direct object |
| IOBJ | Indirect object |
| CCOMP | Clausal complement |
| XCOMP | Open clausal complement |
| **Nominal Modifier Relations** | **Description** |
| NMOD | Nominal modifier |
| AMOD | Adjectival modifier |
| NUMMOD | Numeric modifier |
| APPOS | Appositional modifier |
| DET | Determiner |
| CASE | Prepositions, postpositions and other case markers |
| **Other Notable Relations** | **Description** |
| CONJ | Conjunct |
| CC | Coordinating conjunction |

**Figure 14.2** Selected dependency relations from the Universal Dependency set. (de Marneffe et al., 2014)

| | Tag | Description | Example |
|---|---|---|---|
| Open Class | **ADJ** | Adjective: noun modifiers describing properties | *red, young, awesome* |
| | **ADV** | Adverb: verb modifiers of time, place, manner | *very, slowly, home, yesterday* |
| | **NOUN** | words for persons, places, things, etc. | *algorithm, cat, mango, beauty* |
| | **VERB** | words for actions and processes | *draw, provide, go* |
| | **PROPN** | Proper noun: name of a person, organization, place, etc.. | *Regina, IBM, Colorado* |
| | **INTJ** | Interjection: exclamation, greeting, yes/no response, etc. | *oh, um, yes, hello* |
| Closed Class Words | **ADP** | Adposition (Preposition/Postposition): marks a noun's spacial, temporal, or other relation | *in, on, by under* |
| | **AUX** | Auxiliary: helping verb marking tense, aspect, mood, etc., | *can, may, should, are* |
| | **CCONJ** | Coordinating Conjunction: joins two phrases/clauses | *and, or, but* |
| | **DET** | Determiner: marks noun phrase properties | *a, an, the, this* |
| | **NUM** | Numeral | *one, two, first, second* |
| | **PART** | Particle: a preposition-like form used together with a verb | *up, down, on, off, in, out, at, by* |
| | **PRON** | Pronoun: a shorthand for referring to an entity or event | *she, who, I, others* |
| | **SCONJ** | Subordinating Conjunction: joins a main clause with a subordinate clause such as a sentential complement | *that, which* |
| Other | **PUNCT** | Punctuation | ; , () |
| | **SYM** | Symbols like $ or emoji | $, % |
| | **X** | Other | asdf, qwfg |

**Figure 8.1** The 17 parts of speech in the Universal Dependencies tagset (Nivre et al., 2016a). Features can be added to make finer-grained distinctions (with properties like number, case, definiteness, and so on).

# ATTRIBUTES

I love to eat apples

| | idx | text | pos_ | dep_ | head |
|---|---|---|---|---|---|
| 0 | 0 | I | PRON | nsubj | love |
| 1 | 2 | love | VERB | ROOT | love |
| 2 | 7 | to | PART | aux | eat |
| 3 | 10 | eat | VERB | xcomp | love |
| 4 | 14 | apples | NOUN | dobj | eat |

I love my Apple watch

| | idx | text | pos_ | dep_ | head |
|---|---|---|---|---|---|
| 0 | 0 | I | PRON | nsubj | love |
| 1 | 2 | love | VERB | ROOT | love |
| 2 | 7 | my | PRON | poss | Watch |
| 3 | 10 | Apple | PROPN | compound | Watch |
| 4 | 16 | Watch | PROPN | dobj | love |

# WORD EMBEDDING

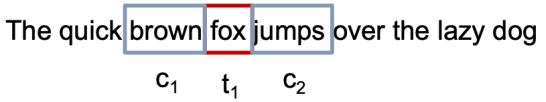
## using spaCy

# WORD EMBEDDING

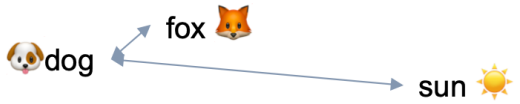
## Word Embedding?

- A numerical feature vector representation of a word (the “meaning of a word“)


- Learning words from context




- Express relations for (frequent) words




- Existing pre-trained embeddings:

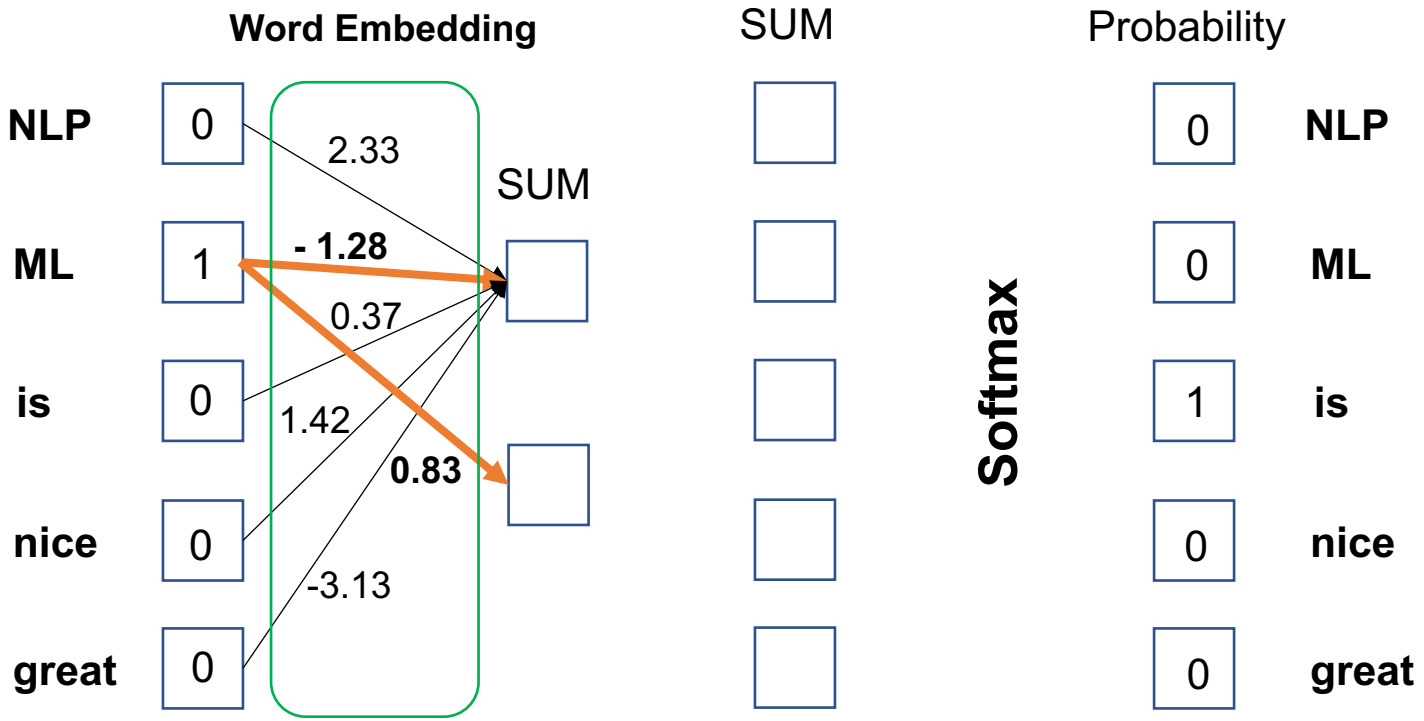
  word2vec, GloVe, fasttext, BERT

# WORD EMBEDDING

## Word Embedding?

**Training Data:**

- NLP is nice!
- ML is great!



# WORD EMBEDDING

## Word2Vec?

- Predicting next word doesn't provide a lot of context to understand each one.
- 2 Strategies that **word2vec** uses to increase the context:
  - Continous Bag of Words: uses surrounding words to predict what occurs in the middle.
  - Skip Gram: uses the word in the middle to predict the surrounding words.

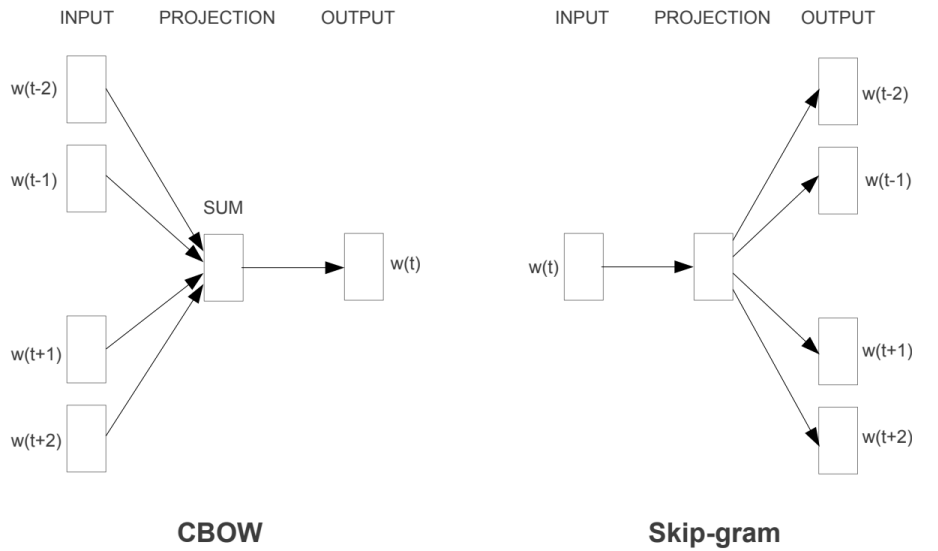



Figure 1: New model architectures. The CBOW architecture predicts the current word based on the context, and the Skip-gram predicts surrounding words given the current word.

# PRACTICAL SESSION ON JUPYTER NOTEBOOK

# See you next Wednesday!