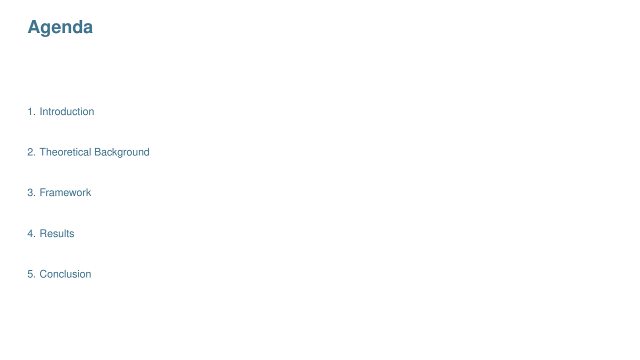
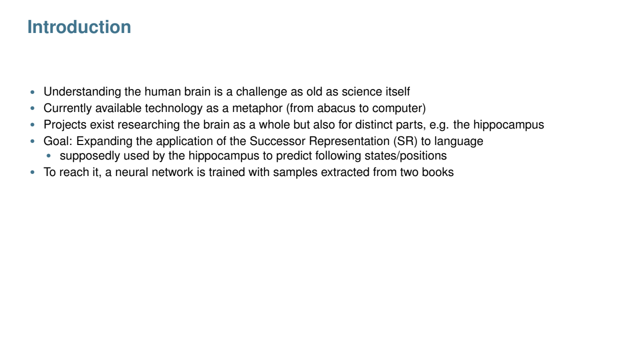
└Agenda



Agenda

1. Introduction
2. Theoretical Background
3. Framework
4. Results
5. Conclusion

ü The agenda consists of five sections: Introduction, the theoretical background, presenting the framework I have developed, my results and a conclusion

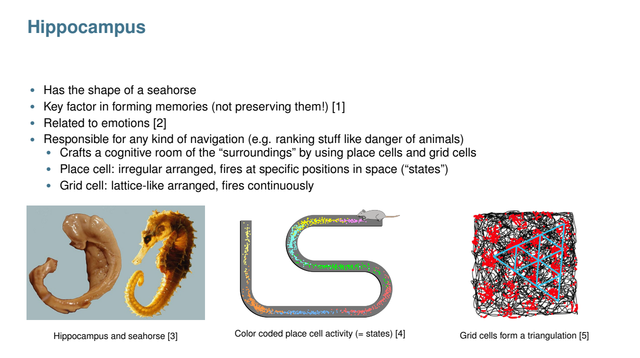


Introduction

- Understanding the human brain is a challenge as old as science itself
- Currently available technology as a metaphor (from abacus to computer)
- Projects exist researching the brain as a whole but also for distinct parts, e.g. the hippocampus
- Goal: Expanding the application of the Successor Representation (SR) to language
  - supposedly used by the hippocampus to predict following states/positions
- To reach it, a neural network is trained with samples extracted from two books

ü I will start with a short overview

- Understanding the human brain is a challenge as old as science itself
- Currently available technology serves as a metaphor (from abacus to computer)
- There exists projects researching the brain as a whole but also for distinct parts, e.g. the hippocampus
- The goal of my thesis is the expansion of the Successor Representation. STICHPUNKT EINBLENDEN Via the SR we can kinda predict future states/positions in an environment, for example our location in a city. The technique was applied beforehand successfully to a spatial environment and shall now be extended to an abstract scenario like language. To put in bluntly, is it possible to achieve proper (long term) word to word predictions by this technique?
- To reach it, a neural network is trained with samples extracted from two books



Hippocampus

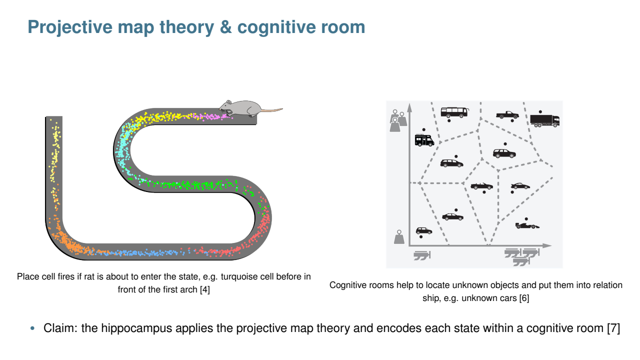
- Has the shape of a seahorse
- Key factor in forming memories (not preserving them!) [1]
- Related to emotions [2]
- Responsible for any kind of navigation (e.g. ranking stuff like danger of animals)
  - Crafts a cognitive room of the “surroundings” by using place cells and grid cells
  - Place cell: irregular arranged, fires at specific positions in space (“states”)
  - Grid cell: lattice-like arranged, fires continuously

Hippocampus and seahorse [3]

Color coded place cell activity (= states) [4]

Grid cells form a triangulation [5]

- The hippocampus has the shape of seahorse
- It plays a key role in forming new memories, not in preserving them...
- ...and is also related to emotions.
- The Hippocampus is responsible for any kind of navigation, for instance ranking stuff like danger of animals
  - Crafts a cognitive room of the “surroundings” by using place cells and grid cells, I will lay out the concept of the cognitive room on the next slide
  - Place cells are irregular arranged and fire at specific positions in space; In the middle picture we can see the blue dots, which encode the firing of place cell. That means it is tied to this specific aisle of the maze. The following orange place cell is active in the region of the last arch. Both resemble one state each.
  - Whereas grid cells are lattice-like arranged and are continuously active [FALLS JEMAND FRAGT: They fire throughout the rat walks around in the square, not just if it is in the center]



Projective map theory & cognitive room

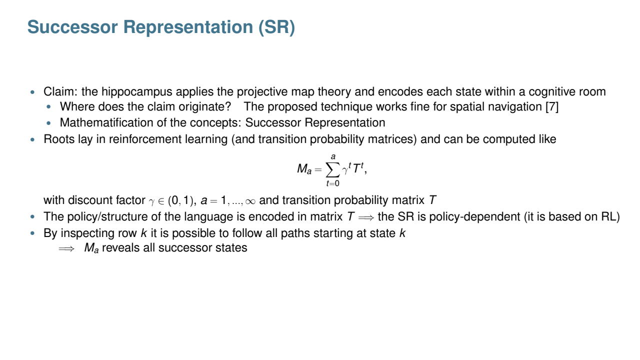
Place cell fires if rat is about to enter the state, e.g. turquoise cell before in front of the first arch [4]

Cognitive rooms help to locate unknown objects and put them into relation ship, e.g. unknown cars [6]

- Claim: the hippocampus applies the projective map theory and encodes each state within a cognitive room [7]

Ü Coming now to the projective map theory and the cognitive room

- Again taking a look at the maze picture. By interpreting the image via the projective map theory a firing place doesn't mark the current state but the immediate successor one. For instance marks the turquoise place cell the first arch and will be active if the rat is in front of it.
- An abstract cognitive room can be seen in 7. Here it shows an already established map of cars depending on weight and engine power we all might have in our minds and each car represents one state. If we now read about an unknown car, for instance an off-road vehicle, we can place it easily in the cognitive room and derive its appearance to some extent.
- These premises lead to the claim: the hippocampus applies the projective map theory and encodes each state within a cognitive room



Successor Representation (SR)

- Claim: the hippocampus applies the projective map theory and encodes each state within a cognitive room
  - Where does the claim originate?   The proposed technique works fine for spatial navigation [7]
  - Mathematification of the concepts: Successor Representation
- Roots lay in reinforcement learning (and transition probability matrices) and can be computed like

$$M_a = \sum_{t=0}^{a} \gamma^t T^t,$$

  with discount factor $\gamma \in (0,1)$, $a = 1, ..., \infty$ and transition probability matrix T
- The policy/structure of the language is encoded in matrix $T \implies$ the SR is policy-dependent (it is based on RL)
- By inspecting row k it is possible to follow all paths starting at state k
  $\implies$ M_a reveals all successor states

ü Having an understanding of the projective map theory and cognitive rooms, we can talk about the SR now. I have left he claim on the slide becauseit serves as good transition.

- Where does the claim originate?
- Stachenfeld et al. tested different environments with classical navigational tasks (not abstract ones like placing cars), for instance the maze depicted earlier. They even had comparative data from rats and humans acting in their cognitive rooms and achieved promising results by applying the theory I am presenting.
- the SR is kinda the Mathematification of these concepts

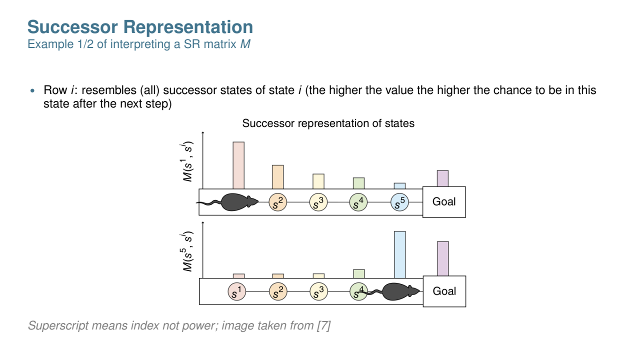
- The roots of the SR lay in reinforcement learning (and transition probability matrices) and can be computed by the sum of exponentiated transition probability matrices

i By γ you can control how influential further apart states are

i By multiplying T with itself n-times you receive the probabilities of being in an arbitrary position after n steps. So, the SR is just a “layered” transition probability matrix T.

- The policy/structure of the language is encoded in matrix T, so the SR is policy-dependent (it is based on RL)
- By inspecting row k it is possible to follow all paths starting at state k
  - M reveals all successor states

– Dadurch dass die vielen Übergangsmatrizen „übereinander“ gelegt werden, erhöhen sich die Werte in jedem Feld mit jedem Summanden. Ist ein Wert nun hoch, bedeutet das, man war „in vielen Zwischenständen“ dort.



Successor Representation

Example 1/2 of interpreting a SR matrix *M*

- Row *i*: resembles (all) successor states of state *i* (the higher the value the higher the chance to be in this state after the next step)

Successor representation of states

Superscript means index not power; image taken from [7]

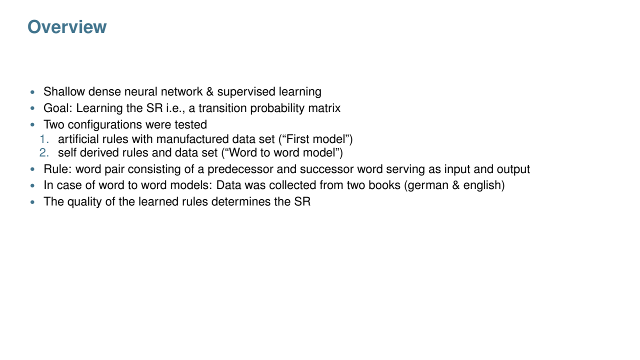
ü It is possible to interpret the rows and columns of the SR, which we will do now. Before we analyze the image, some information about the context in which the values where taken

i The cognitive rooms consists of 6 states in total

i The policy has to actions: going one step or pausing

i The upper half depicts the first row of matrix M and implies the following: Due to the “stay or pause”-policy is the probability for the states s^1, s^2 the largest.

i The same goes for the lower half with state s^5 and `goal`

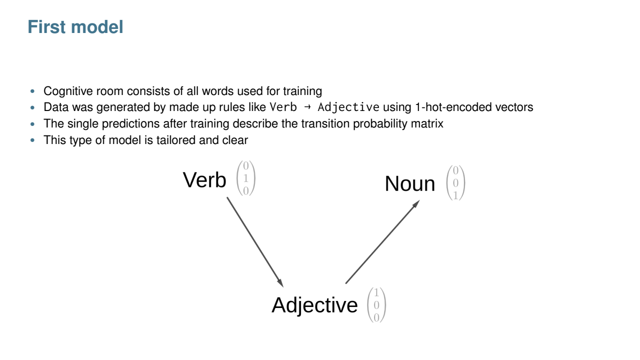


Overview

- Shallow dense neural network & supervised learning
- Goal: Learning the SR i.e., a transition probability matrix
- Two configurations were tested
  1. artificial rules with manufactured data set (“First model”)
  2. self derived rules and data set (“Word to word model”)
- Rule: word pair consisting of a predecessor and successor word serving as input and output
- In case of word to word models: Data was collected from two books (german & english)
- The quality of the learned rules determines the SR

ü Starting with a general overview

- For training a shallow dense neural network with 1 layer was trained with supervised learning.
- The goal was to learn a SR for $t = 0$ which equals a transition probability matrix.
- Two configurations were tested
  - One having artificial rules with a manufactured data set, called“First model”...
  - the other works with self derived rules and data set, called “Word to word model”
- A “Rule” is a word pair consisting of a predecessor and successor word serving as input and output
- In case of word to word models: Data was collected from two books in german & english respectively
- The quality of the learned rules determines the Successor representation



First model

- Cognitive room consists of all words used for training
- Data was generated by made up rules like `Verb → Adjective` using 1-hot-encoded vectors
- The single predictions after training describe the transition probability matrix
- This type of model is tailored and clear

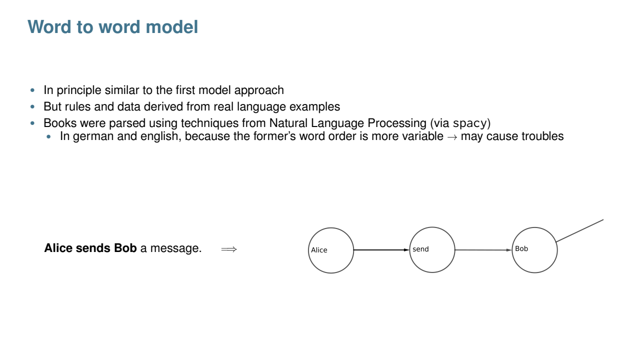
Verb $\begin{pmatrix}0\\1\\0\end{pmatrix}$
Noun $\begin{pmatrix}0\\0\\1\end{pmatrix}$
Adjective $\begin{pmatrix}1\\0\\0\end{pmatrix}$

ü Firstly, I will present the details of the First Model

- The cognitive room consists of all words used for training which is basically a list containing all words
- The data was generated by made up rules like `Verb → Adjective`, the concrete words were chosen randomly and converted to 1-hot-encoded vectors by denoting a one at the index in the cognitive room

i In the picture there is simple scenario depicted. The rules can be combined to a graph. The gray vectors resemble the word and in this case the cognitive room has 3 elements.

- The single predictions after training describe the transition probability matrix
- This type of model is tailored and clear



Word to word model

- In principle similar to the first model approach
- But rules and data derived from real language examples
- Books were parsed using techniques from Natural Language Processing (via spacy)
  - In german and english, because the former’s word order is more variable → may cause troubles

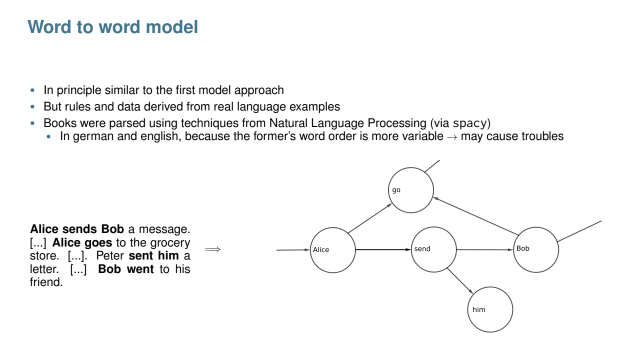
Alice sends Bob a message. ⟹

ü Now, word to word models, which...

- ... are in principal similar to the first model approach
- But rules and data were derived from real language examples
- To do so, books were parsed using techniques from Natural Language Processing and provided via the python module `spacy`. The techniques used will be demonstrated shortly.
  - In german and english, because the former’s word order is more variable, this may cause trouble

i SATZ UND GRAPH ZEIGEN

- As little example: The sentence “Alice sends Bob a message.” is tokenized (this means words become objects with additional information), lemmatized (this stands for mapping conjugated verbs onto infinitives) and finally coupled to have a rule, as we can see in the graph on the right. The middle vertex is annotated (on purpose) with “send” not “sends” due to lemmatization. Each edge encodes a rule with input

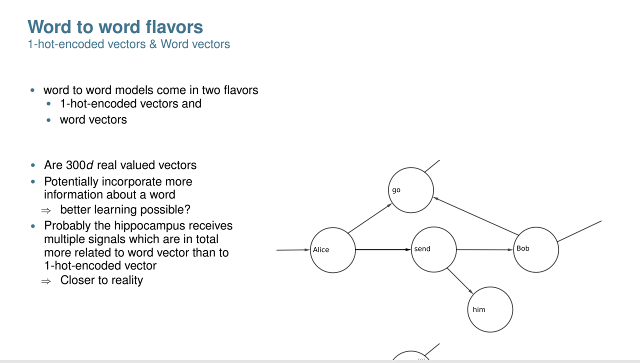


Word to word model

- In principle similar to the first model approach
- But rules and data derived from real language examples
- Books were parsed using techniques from Natural Language Processing (via spacy)
  - In german and english, because the former's word order is more variable → may cause troubles

Alice sends Bob a message. [...] **Alice goes** to the grocery store. [...]. Peter **sent him** a letter. [...] **Bob went** to his friend. ⟹

- In the next steps more and more rules are added



Word to word flavors

1-hot-encoded vectors & Word vectors

- word to word models come in two flavors
  - 1-hot-encoded vectors and
  - word vectors

- Are 300*d* real valued vectors
- Potentially incorporate more information about a word

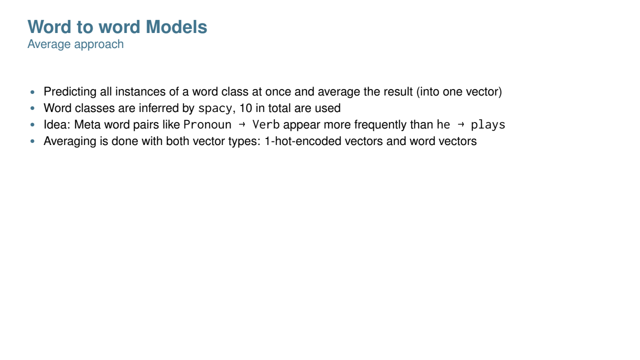
⇒ better learning possible?

- Probably the hippocampus receives multiple signals which are in total more related to word vector than to 1-hot-encoded vector

⇒ Closer to reality

ü KEINE ÜBERLEITUNG

- word to word models come in two flavors
  - equipped with 1-hot-encoded vectors and ...
  - word vectors
- Word vectors can be calculated with `spacy` and are 300*d* real valued vectores as indicated in the graph
- They potentially incorporate more information about a word, STICHPUNKT EINBLENDEN therefore we hope they increase learning quality.
- Probably the hippocampus receives multiple signals which are in total more related to word vector than to 1-hot-encoded vector STICHPUNKT EINBLENDEN



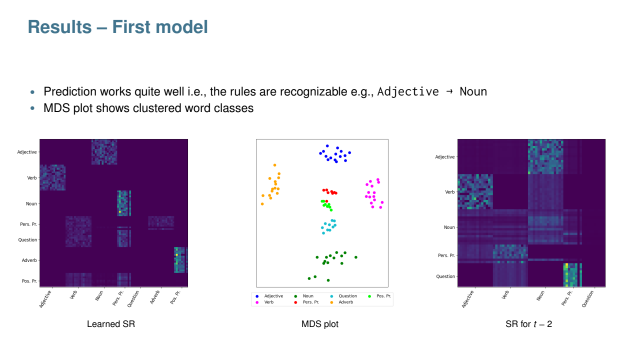
Word to word Models

Average approach

- Predicting all instances of a word class at once and average the result (into one vector)
- Word classes are inferred by `spacy`, 10 in total are used
- Idea: Meta word pairs like `Pronoun → Verb` appear more frequently than `he → plays`
- Averaging is done with both vector types: 1-hot-encoded vectors and word vectors

ü There is also an average approach for word to word models

- It works by predicting all instances of a word class at once and average the result (into one vector)
- The word classes of the 10 highest indices are inferred by `spacy`
- The idea behind averaging was that meta word pairs like `Pronoun → Verb (Pronoun followed by Verb)` appear more frequently than `he → plays (he then plays)`, so a coarser tool may be worth trying
- The Averaging takes place for both vector types: 1-hot-encoded vectors and word vectors



Results – First model

- Prediction works quite well i.e., the rules are recognizable e.g., `Adjective → Noun`
- MDS plot shows clustered word classes

● Adjective	● Noun	● Question	● Pos. Pr.
● Verb	● Pers. Pr.	● Adverb	

Learned SR

MDS plot

SR for $t = 2$

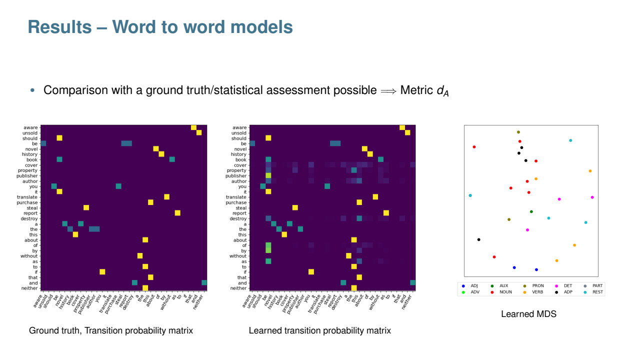
üü Finally, I will present the results.

ü Again, we'll begin talking about the First Model

- Works well because the SR performs best in these well defined scenarios

i To avoid clutter only word classes are labeled, indeed one row corresponds to one word.

- The MDS plot shows clustered word classes, which also means learning was successful. Although not necessary for this type of model, it offers some visual feedback for configurations using a larger data set because their matrix can't be plotted
- LETZTES BILD: SR for t=2, the two step rule `Question → Pers. Pr. → Verb` is visible and it is possible to recognize the following states of a `question word`, here they are `Personal Pronoun` and `Verb`



Results – Word to word models

- Comparison with a ground truth/statistical assessment possible ⟹ Metric d_A

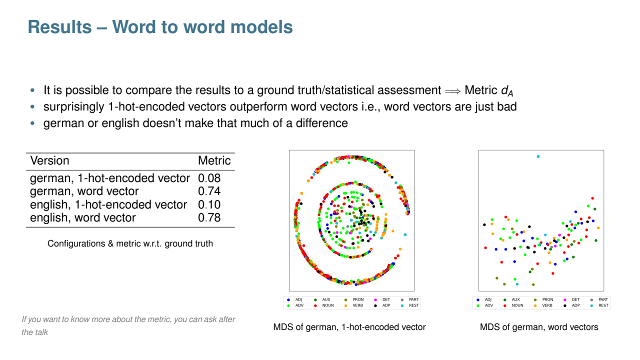
Ground truth, Transition probability matrix

Learned transition probability matrix

Learned MDS

ü word to word models show a different outcome.

- It is possible to compare the results to a ground truth/statistical assessment, hence the Metric d_A comes into play
- On the left (LINKES BILD) is a ground truth depicted and next to it the predictions of the network. Although there is a resemblance visible it has to be assessed cautiously because a tiny data set was used for illustration purposes only to convey an intuition for the results and the procedure.
- The MDS is displayed because matrices won't provide visual feedback anymore and as seen before sufficient learning is also visible in the cluster plot.



Results – Word to word models

- It is possible to compare the results to a ground truth/statistical assessment ⟹ Metric d_A
- surprisingly 1-hot-encoded vectors outperform word vectors i.e., word vectors are just bad
- german or english doesn't make that much of a difference

Version	Metric
german, 1-hot-encoded vector	0.08
german, word vector	0.74
english, 1-hot-encoded vector	0.10
english, word vector	0.78

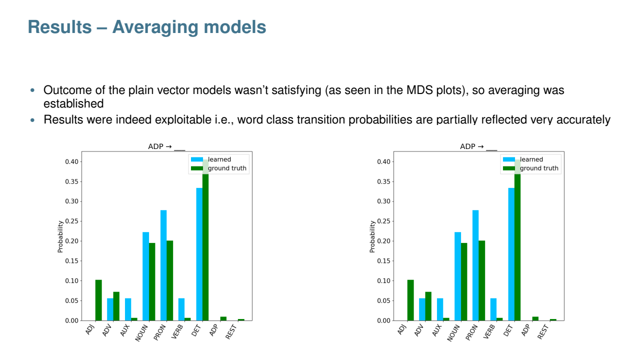
Configurations & metric w.r.t. ground truth

If you want to know more about the metric, you can ask after the talk

MDS of german, 1-hot-encoded vector

MDS of german, word vectors

- In the table we find all four configurations comprising of 1-hot-encoded vectors and word vector each ran with german and english.
  i The metric maps matrices close to 0 if the equal the ground truth and to 1 if not
- Surprisingly 1-hot-encoded vectors perform better than word vectors. Since they encode a word with more than two numbers, this was nothing to reckon with.
- We expected that english performs better than german due to the more static word order which was a fallacy
- In the image on the left is the MDS plot of the SR using 1-hot-encoded vectors illustrated and on the right using word vectors. In both cases the hoped clusters didn't form and the configuration on the right seems to be omit a degenerated output at all.



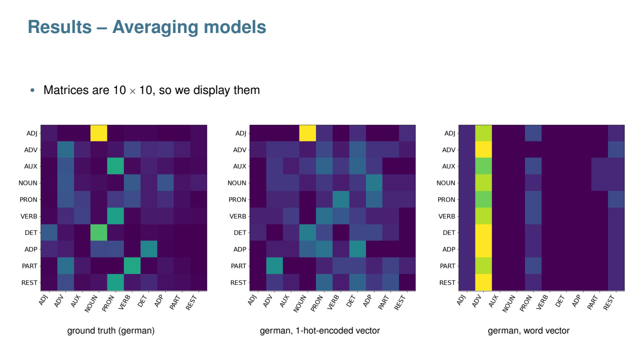
Results – Averaging models

- Outcome of the plain vector models wasn’t satisfying (as seen in the MDS plots), so averaging was established
- Results were indeed exploitable i.e., word class transition probabilities are partially reflected very accurately

ü After all, coming to the Average Approach

- Because the outcome of the plain vector models wasn’t satisfying (as illustarted in the MDS plot beforehand), an averaging approach was developed. Based on the predictions shown beforehand, transition probabilities between word classes were calculated
- The results are indeed exploitable especially using 1-hot-encoded vectors. There, word class transitions are partially reflected very accurately as we can see in both bar plots

i In both figures the learned probabilities and the ground truth probabilities are plotted, in blue and green respectively. The title is referring to the first word class, whereas the x axis shows the particular successor. Both diagrams show good accuracy for the word classes.



Results – Averaging models

- Matrices are 10 × 10, so we display them

ground truth (german)

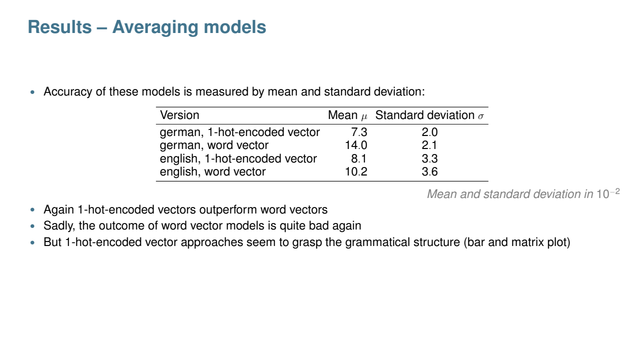
german, 1-hot-encoded vector

german, word vector

- Matrices are 10 × 10, so we display them

i On the left there is the ground truth of the book illustrated, in the middle a 1-hot-encoded vector model and on the right the word vector equivalent. The latter shows no learning at all, whereas similarities to the ground truth are recognizable in the middle.

i The picture is similar for the english book



Results – Averaging models

- Accuracy of these models is measured by mean and standard deviation:

Version	Mean μ	Standard deviation σ
german, 1-hot-encoded vector	7.3	2.0
german, word vector	14.0	2.1
english, 1-hot-encoded vector	8.1	3.3
english, word vector	10.2	3.6

Mean and standard deviation in 10^{-2}

- Again 1-hot-encoded vectors outperform word vectors
- Sadly, the outcome of word vector models is quite bad again
- But 1-hot-encoded vector approaches seem to grasp the grammatical structure (bar and matrix plot)

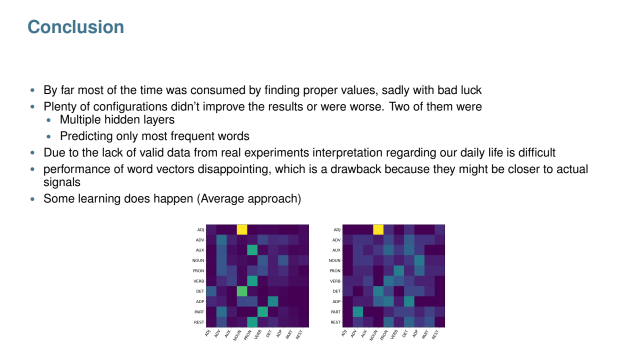
- The accuracy of these models is measured by mean and standard deviation
- Sadly, the outcome of word vector models is not worth mentioning it again. This is in particular unsatisfactory because word vectors might be closer to real signals.
- But 1-hot-encoded vector approaches seem to grasp the grammatical structure, which is justified by the bar and matrix plots

i FALLS JEMAND FRAGT Anderes Maß, es hier explizit die verschiedenen Zeilen mit der Ground truth verglichen wurden, Außerdem hat man weniger Spalten bzw. Zustände im Allgemeinen vorliegen, sodass man bspw. mit der Angabe der Standardabweichung auch etwas anfangen kann

i FALLS JEMAND FRAGT

  i „Mean“ bedeutet: GT - SR, then row-wise mean, finally mean of means

  i „Std. deviation“ GT - SR, then row-wise std. dev., finally std. dev of std. devs.



Conclusion

- By far most of the time was consumed by finding proper values, sadly with bad luck
- Plenty of configurations didn't improve the results or were worse. Two of them were
  - Multiple hidden layers
  - Predicting only most frequent words
- Due to the lack of valid data from real experiments interpretation regarding our daily life is difficult
- performance of word vectors disappointing, which is a drawback because they might be closer to actual signals
- Some learning does happen (Average approach)

ü Coming to the Conclusion

- By far most of the time was consumed by finding well functioning values.
- Plenty of configurations didn't improve the results or were worse. Although it took much space during the months the dead ends aren't illustrated in the presentation but in the thesis. Two of them were
  - Multiple hidden layers
  - Predicting only most frequent words
- Due to the lack of valid data from real experiments interpretation regarding our daily life is difficult
- Unfortunately, performance of word vectors is in general disappointing, which is a drawback because they might be closer to actual signals
- Nevertheless, some learning does happen and paths can be reconstructed (Average approach) ENDE DES VORTRAGS