Data Science Academy

Data Science for Business – Becoming a Data Science Expert (D)

Pilot Presentation:
for participants of and use in the pilot only

14.09.2021

Boehringer Ingelheim

Agenda week three

	Introduction
1	Recap Basic Machine Learning and Python
2	Complex Models
3	Model Evaluation
4	Hyperparameters
5	Unsupervised Learning
6	Gradient Descent
7	**Deep Learning and Image Recognition**
8	**Deep Learning and Natural Language Processing**
9	Repetition
10	Bias and Ethics in Machine Learning
11	Introduction to Data Science with AWS

Schedule week three

Week 3			
	Day 1 **Monday, 13.09.2021**		**Day 2** **Tuesday, 14.09.2021**
Start: 12:00	Recap	Start: 12:00	Recap
	7 – DL: Image Recognition (Part 1)		8 – DL: NLP (Part 1)
14:00 – 15:00	Break	14:00 – 15:00	Break
	7 – DL: Image Recognition (Part 2)		8 – DL: NLP (Part 2)
End: 18:00	Q&A and Feedback	End: 18:00	Q&A and Feedback

We will also have several short coffee breaks in between.

Feedback for pilot training

We aim to provide a great training experience for you and are looking forward to receiving your feedback!

You will have three different ways to give us your feedback on each training day:

1. We will have an **anonymized** feedback collection **after the last session** of each day per **Myforms.**
2. We will have an **open feedback round and discussion** at the **end of each training day.**
3. Please also **take notes** regarding your ideas during the sessions: **locally or via the Mural Board** which you can reach via LINK.

Quiz: Recap Image Recognition

Please join at slido.com with #031 077.

Let’s go through some questions together.

Let’s see what you think. All answers will be anonymous.

Module 8

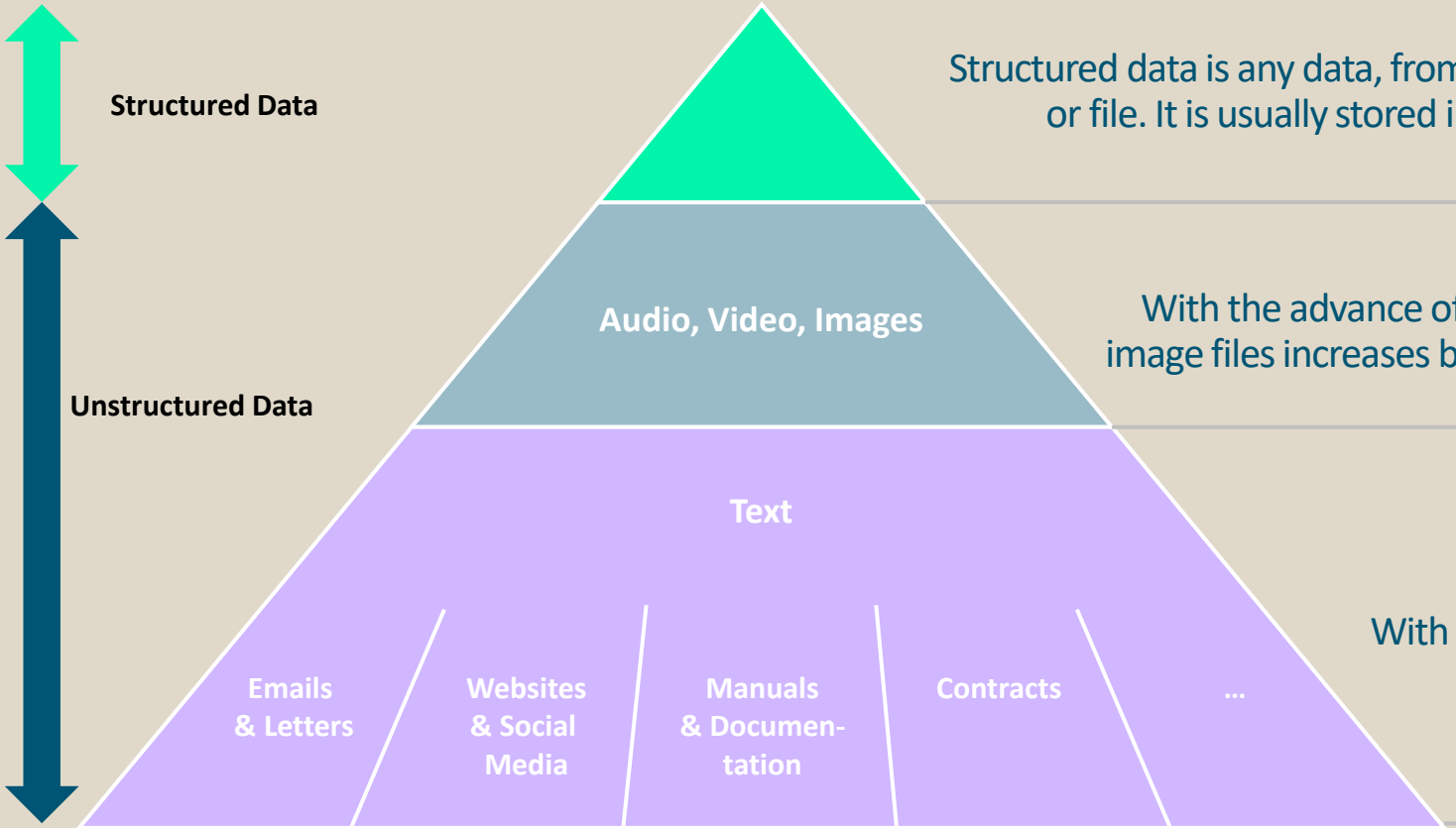
Deep Learning and Natural Language Processing

Agenda

	Introduction
1	Recap Basic Machine Learning and Python
2	Complex Models
3	Model Evaluation
4	Hyperparameters
5	Unsupervised Learning
6	Gradient Descent
7	Deep Learning and Image Recognition
8	**Deep Learning and Natural Language Processing**
9	Repetition
10	Bias and Ethics in Machine Learning
11	Introduction to Data Science with AWS

Vast amounts of unstructured data can be made accessible with Natural Language Processing

Up to 80% of data within any enterprise is unstructured and usually not considered for analytics purposes.



Structured data is any data, from numbers to letters, located in a fixed field of a record or file. It is usually stored in a database and can be used for business intelligence matters right away.

The content of media formats is unorganized. With the advance of artificial intelligence, the analysis of audio, video, and image files increases but it is still focused on few and well-defined use cases.

While the amount of text is skyrocketing, it becomes more and more impossible for a human being to read and digest this data. With using **Natural Language Processing (NLP)**, meaningful insights can be derived from any textual source.

To become truly data-driven, an enterprise needs to leverage all its data assets, including unstructured data.

Step-by-step guide for conducting a successful NLP project

Identify use case

- Select use case with low complexity and one language.
- Look for business processes which heavily rely on text and are frequent and standardized.

Collect data

- Data sample should be sufficiently big and representative.
- If possible, provide domain-specific dictionaries.

Create gold standard

- Annotate (label) data samples to mark relevant segments or to classify each sample.
- Ensure high quality of annotation as this impacts the quality of NLP.

Divide data

- 80% of data samples are used for training purposes.
- 20% of data samples are used for quality testing.

Set up NLP pipeline

- Analytical pipeline consists of data cleansing, linguistic pre-processing and machine learning algorithms.

Assess NLP quality

- Accuracy of NLP pipeline is calculated by processing test data samples and comparing the results to the gold standard.

Refine NLP approach

- Usually, a second iteration of training is done to improve the NLP quality.

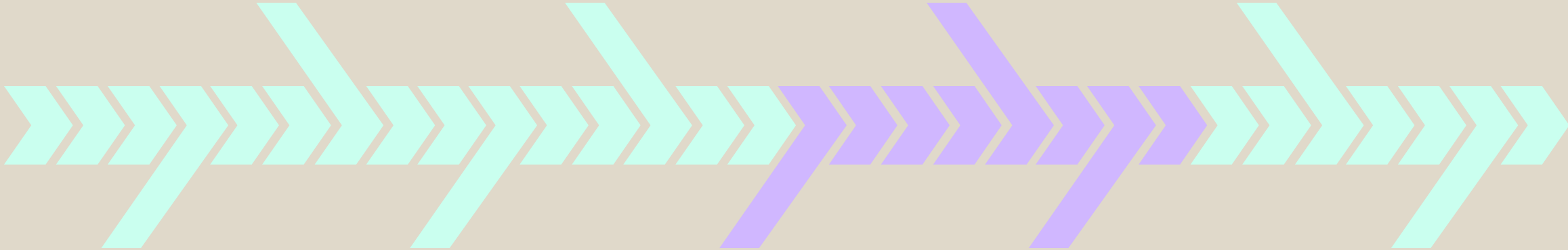
Integrate NLP solution

- NLP solution is embedded in the architectural setup.
- Interfaces are implemented for data ingestion and output.

Move NLP solution in production

- Once in production, additional user feedback can be gathered to improve the NLP solution on a regular basis.

Setting up a NLP pipeline is not trivial



Tokenization
Break sentences into words, sylables or letters.

Stemming/Normalization
Merge Tokens of the same stem.

Numericalization
Assign each token a specific numeric value

Train a language model
Teach the AI your language by providing a corpus

Generate word embeddings
Teach the AI your language by providing a corpus

Train the model for your NLP Task
Classification, Entity Extraction, Machine Translation,....

Predicting the rating of movie review

Positive Review

This is a extremely **well-made** film. The acting, script and camera-work are all first-rate. The music is **good**, too, though it is mostly early in the film, when things are still relatively cheery. There are **no** really superstars in the cast, though several faces will be familiar..

Negative Review

Every once in a long while a movie will come along that will be so **awful** that I feel **compelled** to warn people. If I labor all my days and I can save but one soul from watching this movie, how **great** will be my **joy**..

Text Classification

- Performing Sentiment Analysis
- How can we make a prediction?
  - Based on the words used
  - Based on the interaction between words
- How can we convert text into a format a machine will understand?
- Most ML models have a fix input size. How do we convert the

Tokenization

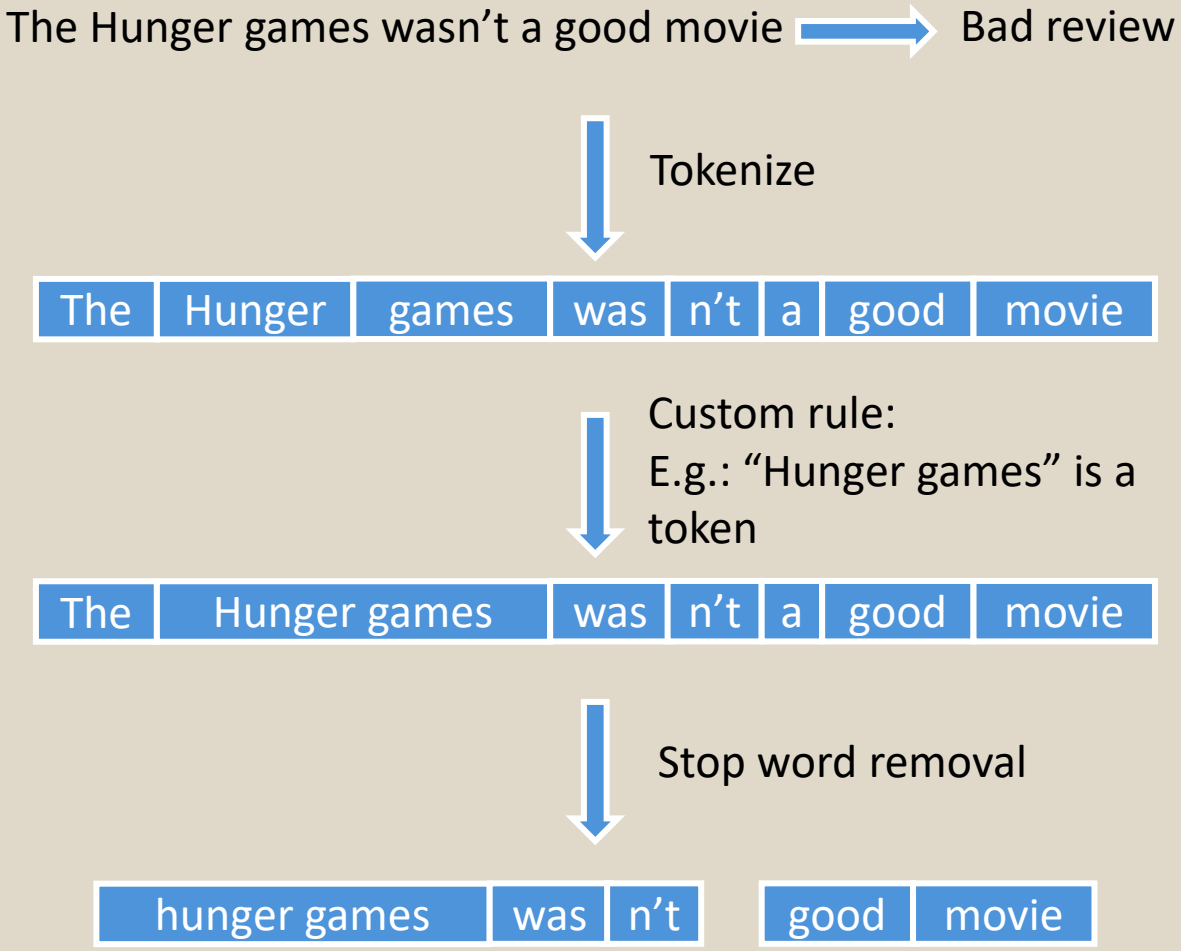
Tokenization

- Take a text and split it into tokens
- The tokenizer has certain rules on how to split the text
- Additional steps might be to convert all tokens to lower case
- This reduces the number of different tokens we can have

Also often used is stop word removal

- Remove punctuation
- Remove fill words like the, a, very, be (remove words that occur to often)
- Remove rare words

After this process, we need additional steps to further reduce the size of our vocabulary



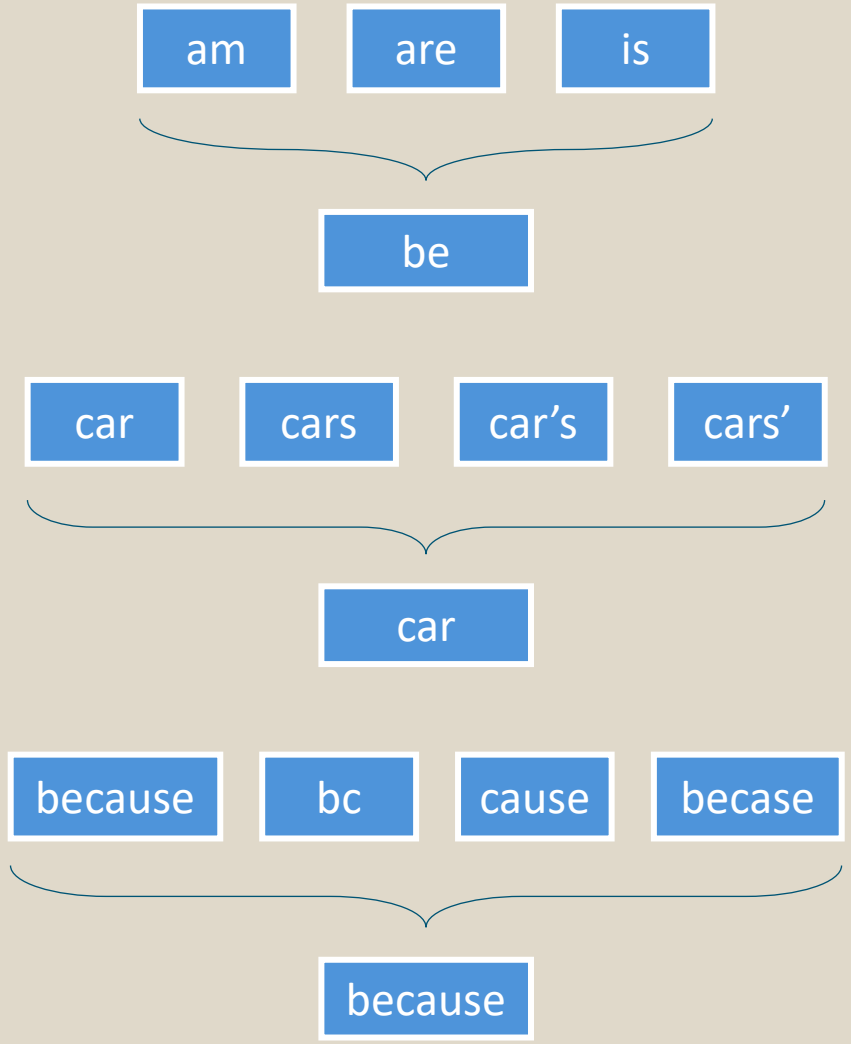
Stemming and lemmatization

Reduce the number of tokens by reducing words to base forms

Additionally

- Fix spelling mistake
- Detect abbreviations

Creating these rules is a manual process



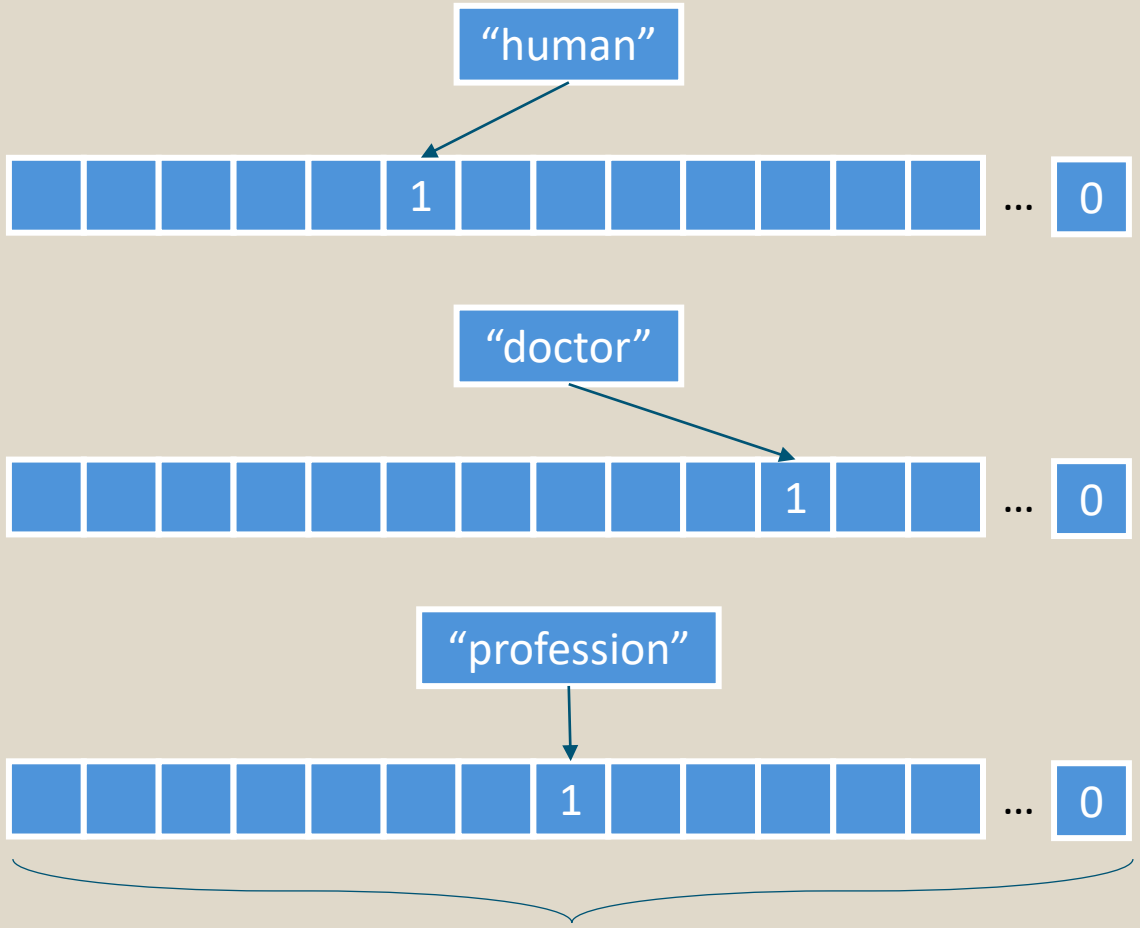
Numericalization

The easiest way to convert tokens into numeric vectors is one-hot encoding

- Each unique token gets a position in the vector
- If the word is present the position has the value one, all others the value 0

What is the problems with one-hot encoding?

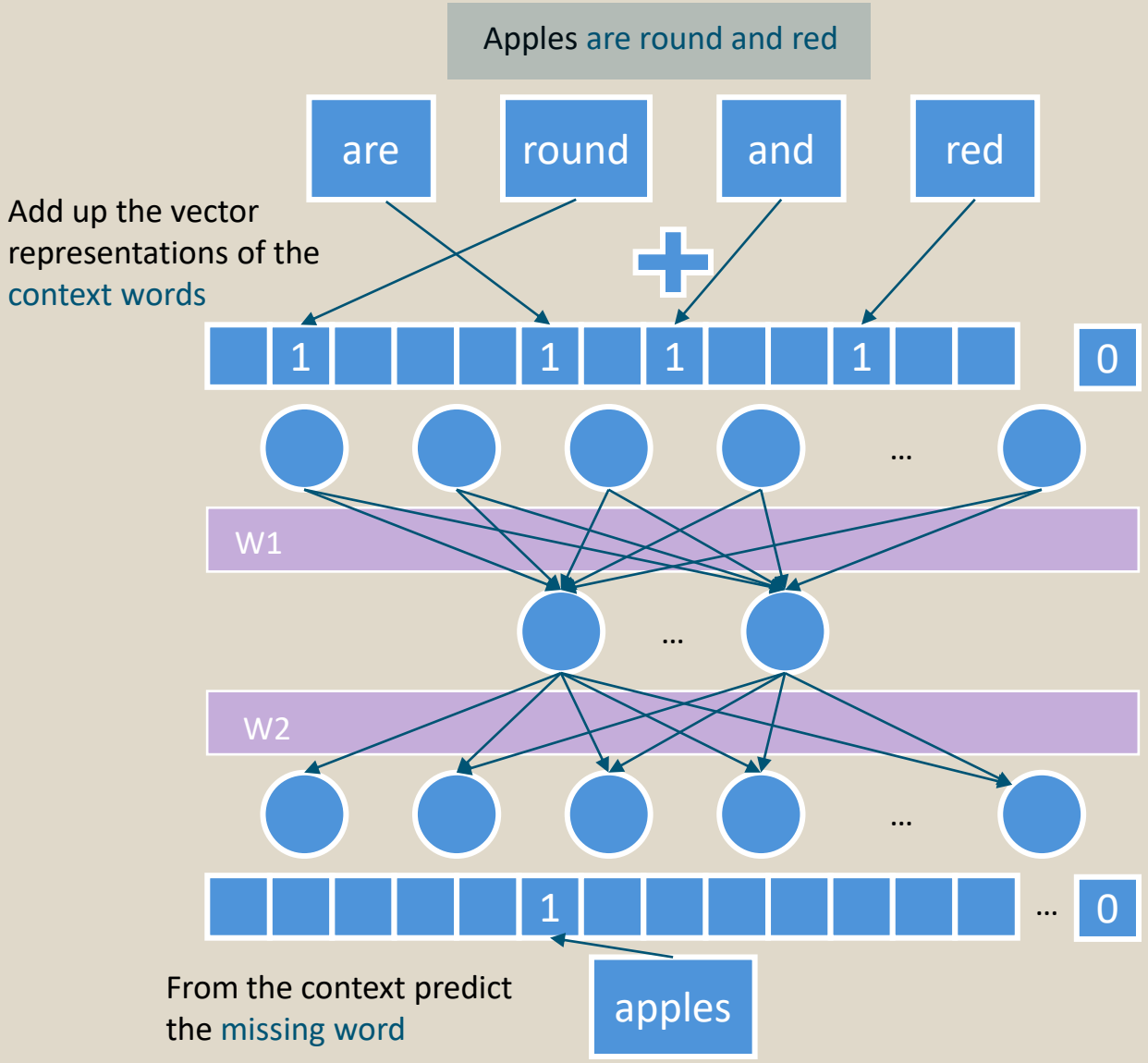
- To many different tokens
- Super sparse feature vector
- No relationship between words



Size English vocabulary: 207,016 [Wikipedia]

Word2Vec: Word embeddings explained

- Build a special Neural Network that reduces the dimensionality of the one-hot encoded words
- We try to predict the context of the word from the word or predict the word with the context of the word
- The weights W_i learn to encode a word into a low dimensional embedding
- The embeddings can be pre-trained and reused for different learning tasks



Word Embeddings

Transforming a word into low dimensional space creates a new space

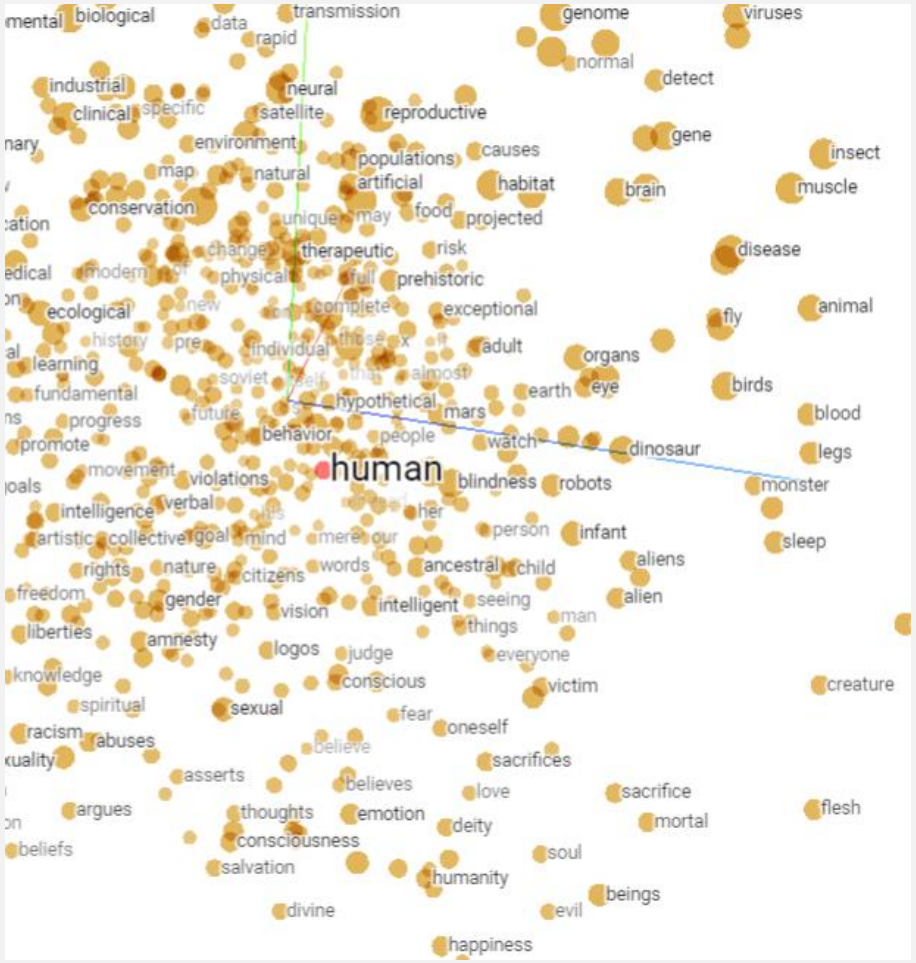
The space encodes the meaning of the word

The distance between embeddings represents real relationships

We solved our problems from above

- Reduced dimensionality to a reasonable size
- Removed the sparsity of the input vector
- Represent meaning of words rather than categories

See: http://projector.tensorflow.org/



Nearest points in the original space:

beings	0.492
humans	0.534
animal	0.575
social	0.596
mind	0.608
our	0.610
humanity	0.614
physical	0.619
individual	0.624
man	0.625
evolution	0.632
brain	0.644

We can even do meaningful calculations on this new embedding space

Two-dimensional representation of the feature embedding

Recap: Text Classification

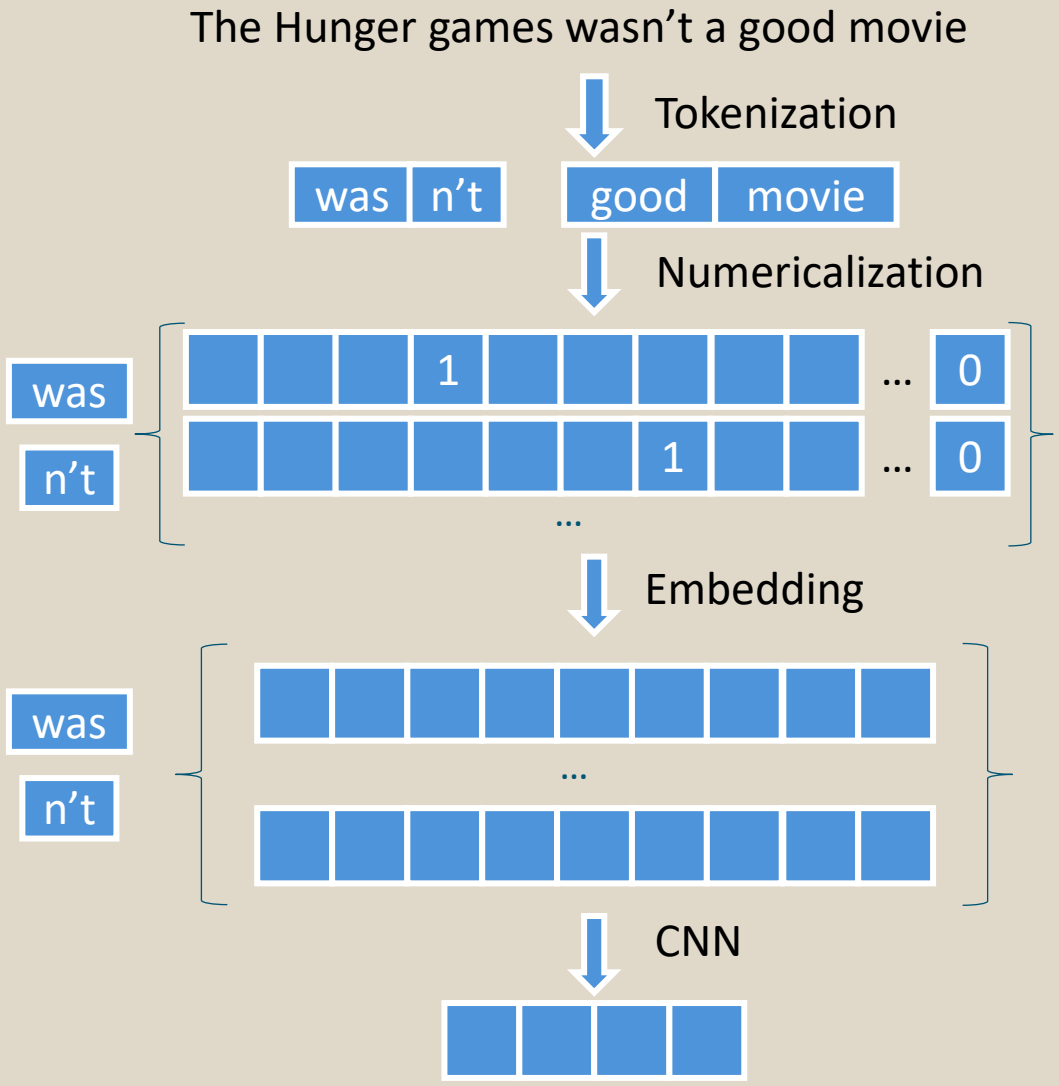
We can now take our review

- Convert each word into its vectorized form
- Calculate the embedding representation
- Pad all sentences, so they have the same size

We effectively converted our review dataset into a set of numeric vectors

We can use a 1D convolutional neural network to classify our text

- Our input will be a two-dimensional matrix where each words embedding is stacked together
- Because the fully connected part of the model has a fixed size, all inputs have to be padded at the end



Try it yourself!

In the following exercises

Recurrent Neural Networks (RNNs)

Modeling Sequential Problems with recurrent layers

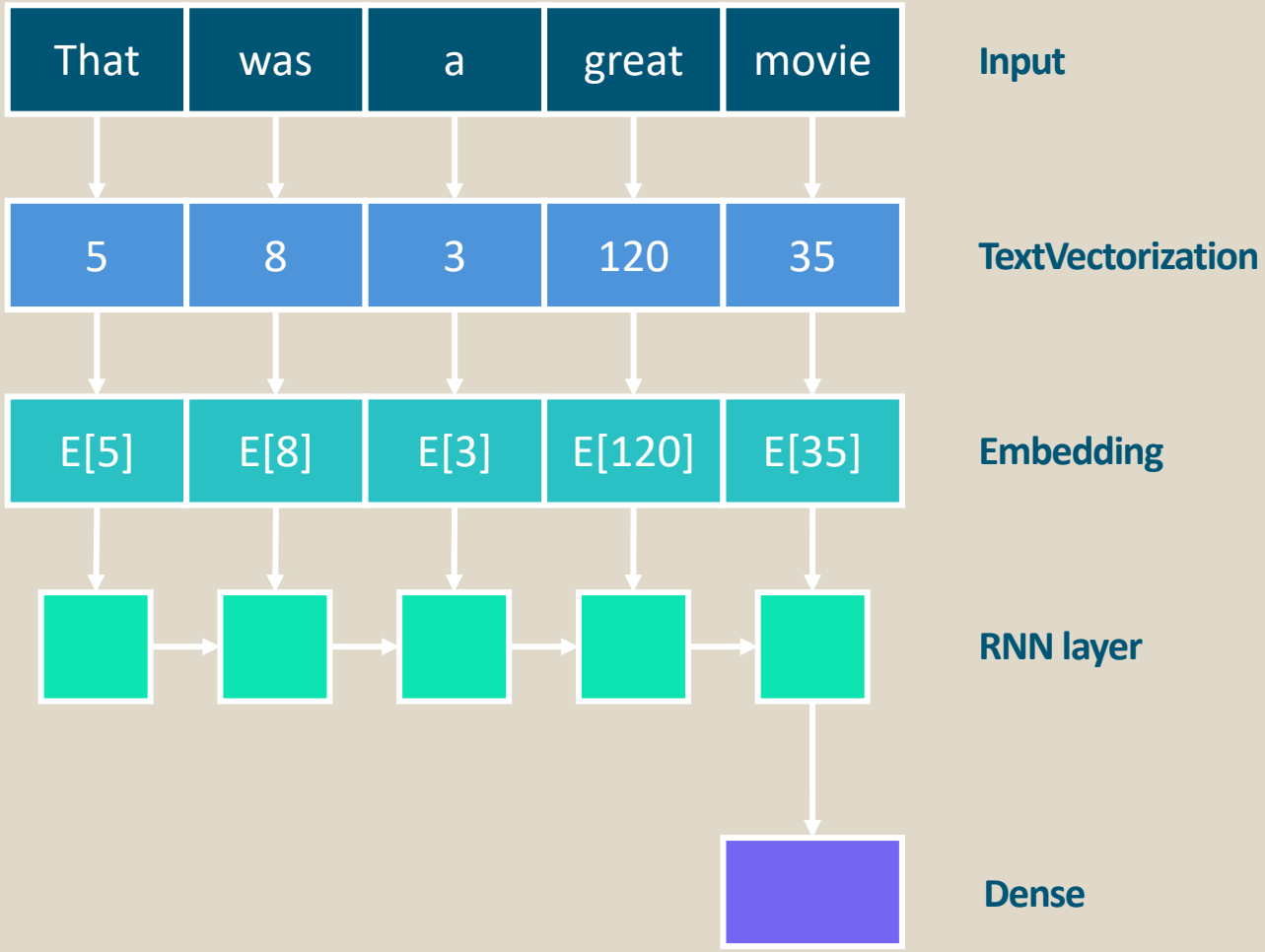
- Layers have a memory of the past, by passing a hidden state to the next neuron
- More advanced versions of this layer, like Long-Short-Term-Memory (LSTM) layers have their own memory of the past
- After traversing the whole sequence, we get a single output representing the sequence
- We can use a Dense layer to make predictions based on these features

Advantages

- Has memory
- Has no fixed input window size

Disadvantages

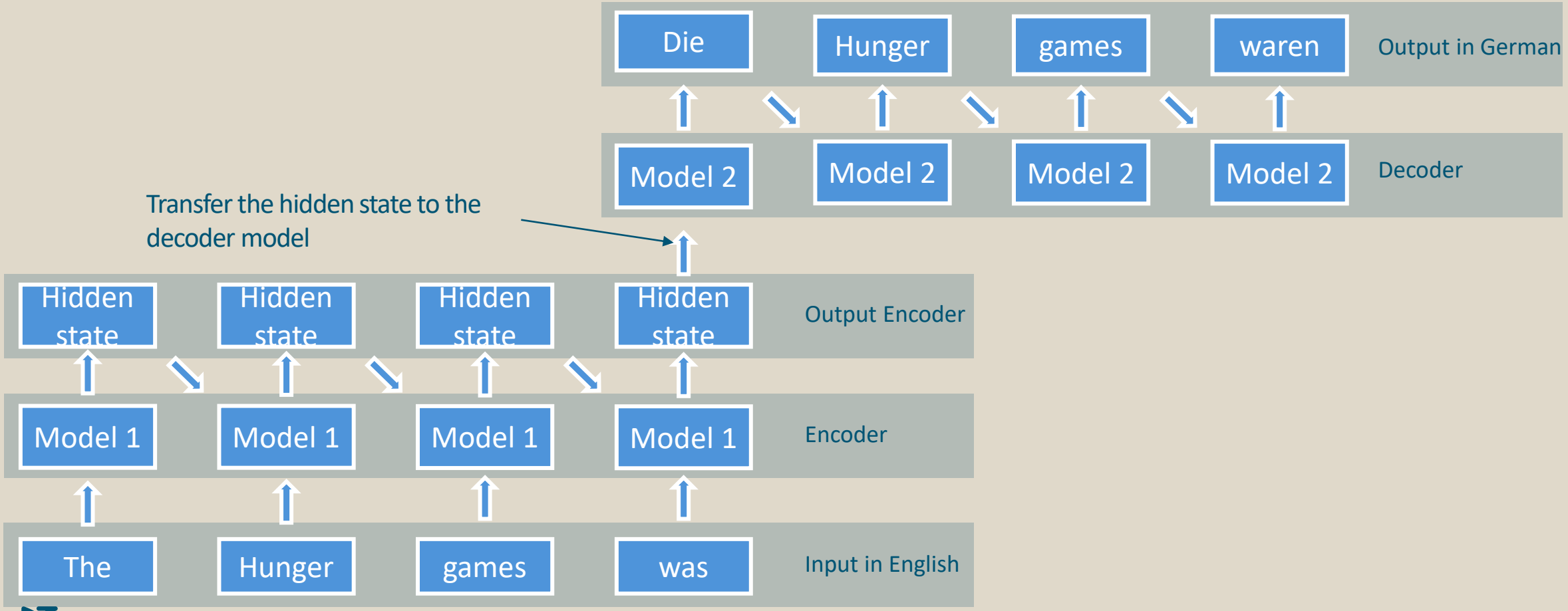
- Learns slowly because of sequential sequences
- Memory difficult to tune (often forgets context to fast -> leads to unrealistic behavior)
- Training erratic



Building our own Translator

This is a “sequence to sequence” model. We can use to translate between languages.

We use the hidden state of the encoder to initialize the decoder network for a translation.



Attention is all you need

In RNN the state is calculated sequentially

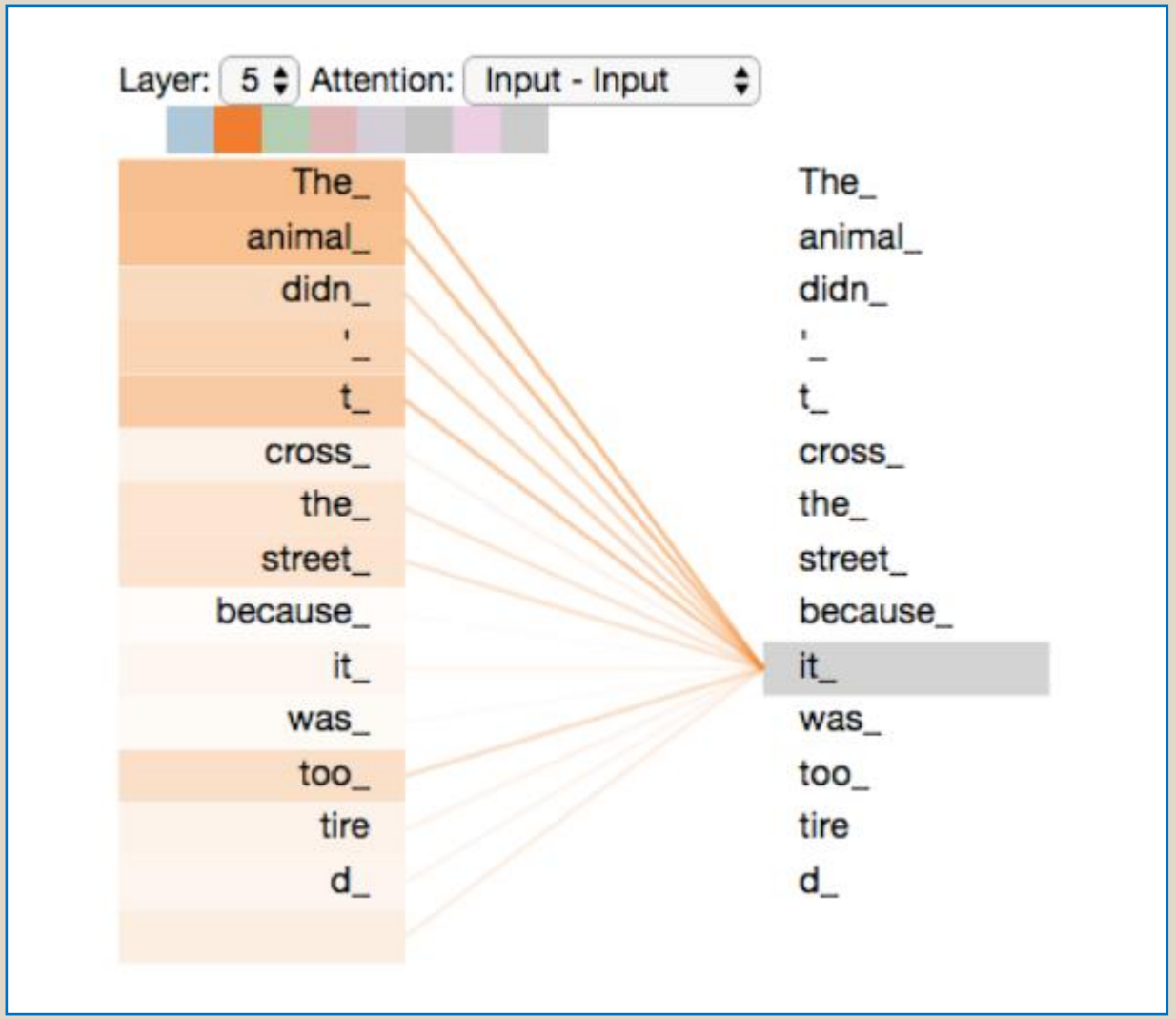
- Attentions works without dependencies on previous steps

The update of the weights can only be performed in sequence, creating problems with vanishing gradients

- Attention evaluates each sample independently

Also, the LSTM has to focus on different aspects depending on the word it is looking at

- Attention chooses a new context for each word



Entity extraction

The act of locating and classifying mentions of an entity in a piece of text

- Distinguish between different types of names (e.g., person, place, organization, product, etc.)

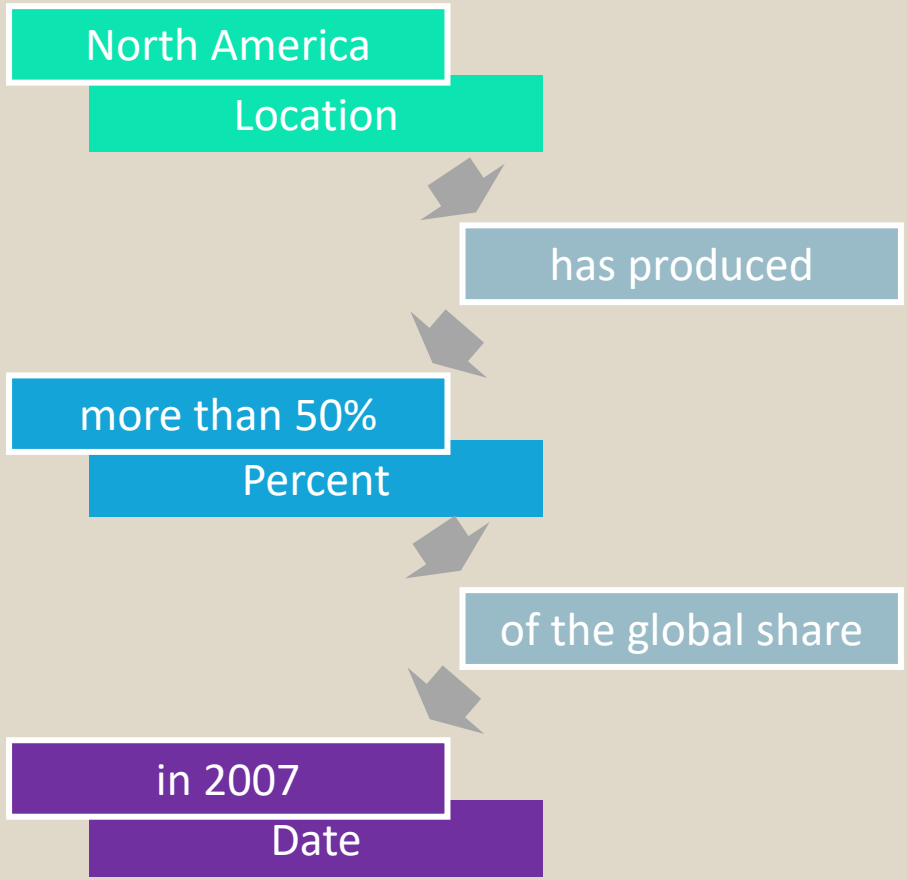
Advanced techniques that are made possible are:

- Entity relationship extraction: Reveals connections between entities
- Linking: Link different information sources for cross-referencing
- Fact extraction: Extract data associated with an entity to deliver a specific response

Use cases in:

- Customer support
- Optimizing search engines
- Preprocessing for NLP tasks
- Chatbots

Identify and classify the entities in the sentence



Using Entity extraction in chatbot systems

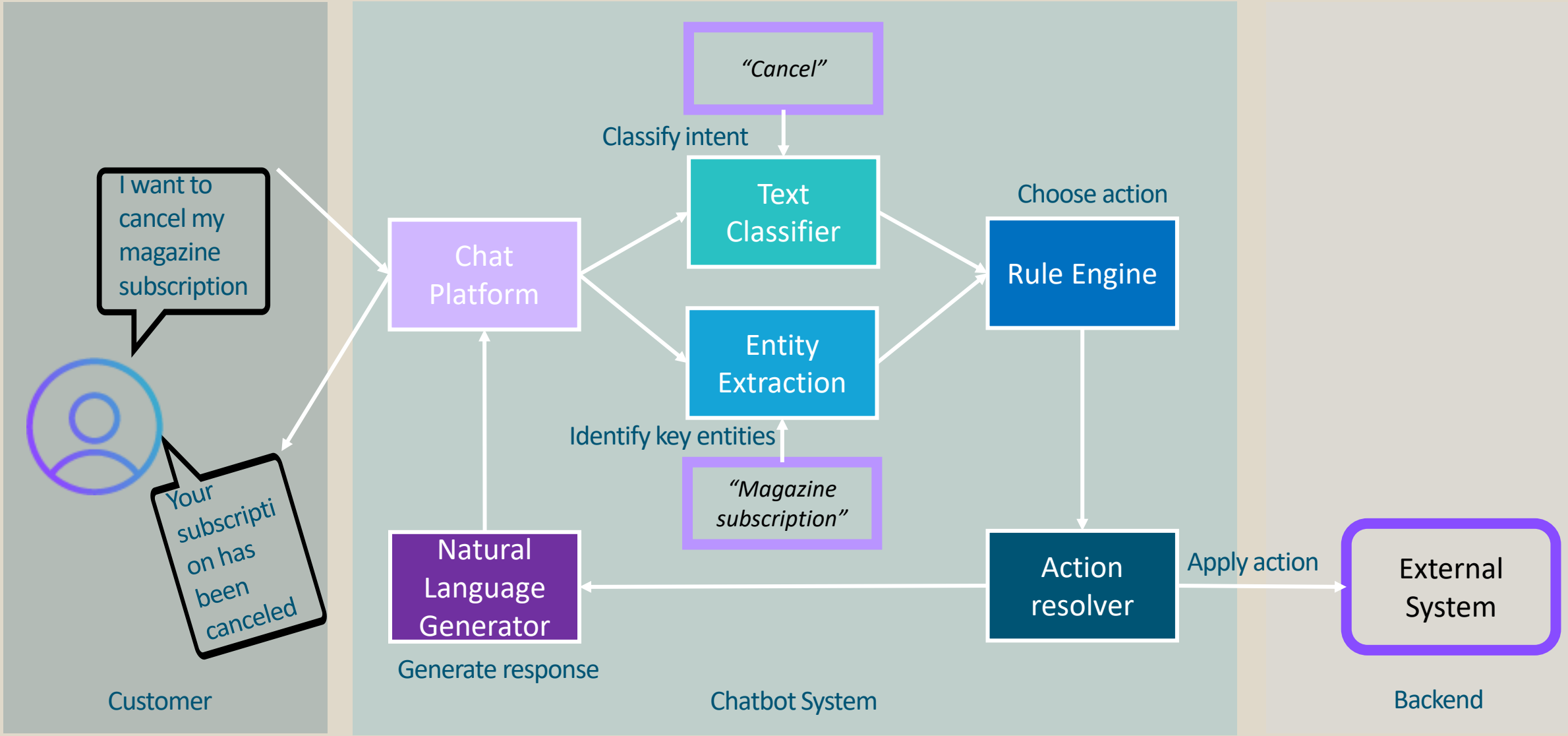
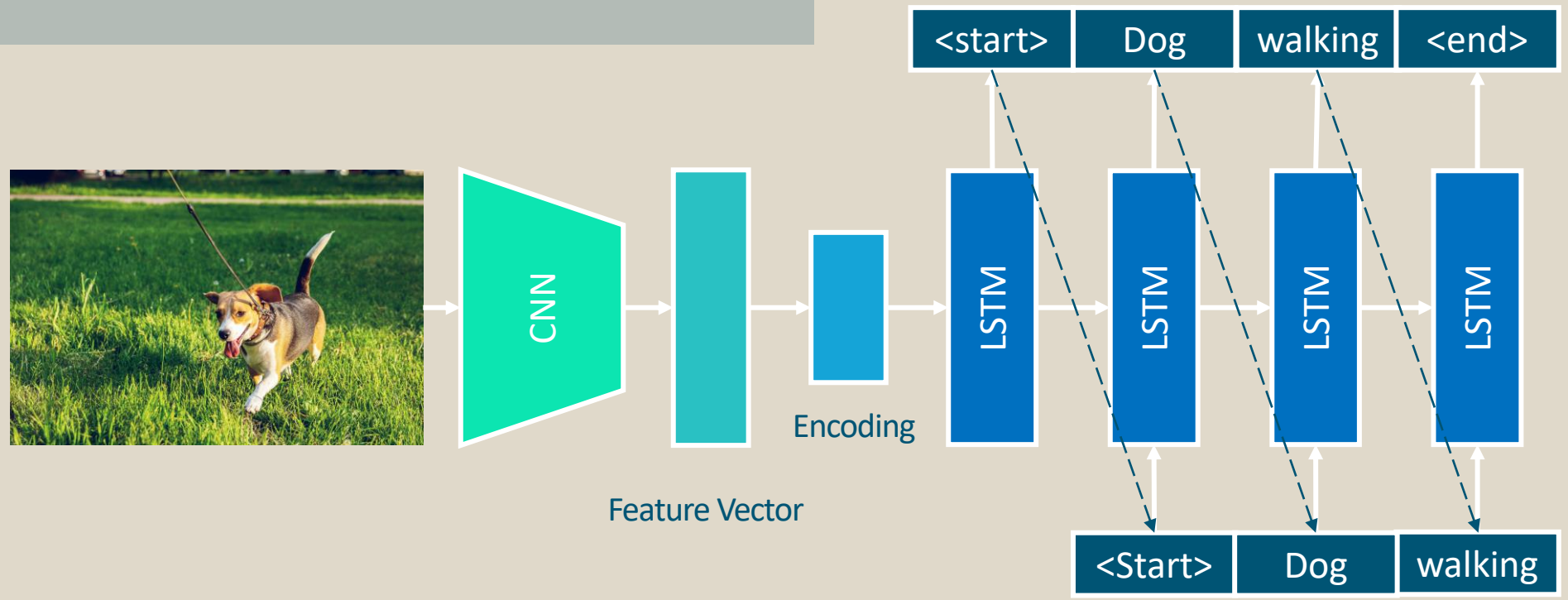


Image Captioning using Deep Learning

Given an image like the example below, your goal is to generate a caption such as “Dog walking on a leash”

Use a pretrained Image recognition model and use the extracted features as the input to the natural language processing model



Visual Question Answering

CNN

Pre-trained image classification model

Feature Vector

Concat

Network

2

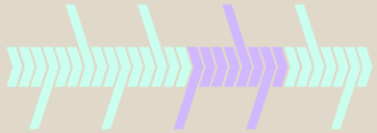
Answer

RNN Model

How many glasses are on the table?

Feature Vector

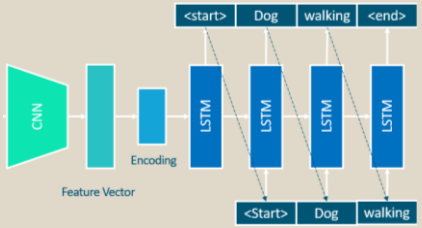
Important takeaways



For NLP we have to build a specialized preprocessing pipeline with:

- Tokenization
- Stemming
- Numericalization

The next step is to encode the numerical values in a meaningful context. This is accomplished by training low-dimensional **embeddings**.



Advanced use cases of NLP include

- Text classification
- Entity extraction
- Image captioning
- Visual question answering

Using multi-modal models allows us to combine different modalities like image and text

Try it yourself!

In the following exercises

Agenda

	Introduction
1	Recap Basic Machine Learning and Python
2	Complex Models
3	Model Evaluation
4	Hyperparameters
5	Unsupervised Learning
6	Gradient Descent
7	Deep Learning and Image Recognition
8	**Deep Learning and Natural Language Processing**
9	Repetition
10	Bias and Ethics in Machine Learning
11	Introduction to Data Science with AWS

Feedback and Q&A

Thank you

If you would like any further information please contact
Werner,Dr.,Fabian_Georg (BI X) BIX-DE-I
<fabian_georg.werner@boehringer-ingelheim.com>