University of Zurich[UZH]

Institute of Computational Linguistics

Machine Translation

1 Introduction

Mathias Müller

Who we are

Dozent

Mathias Müller

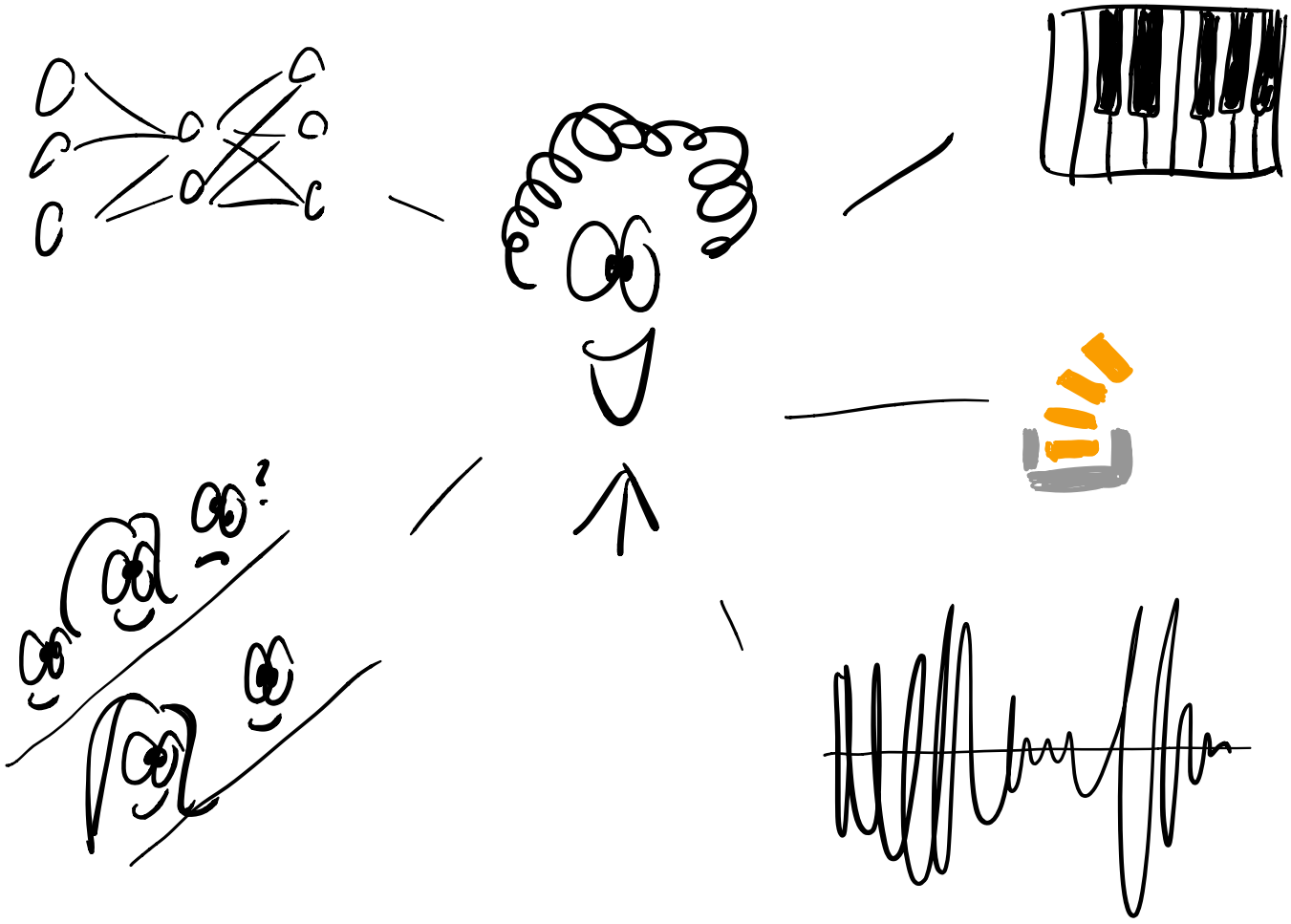
Büro: AND-2-20

Tutoren

Dominik Martinez

Nicolas Spring

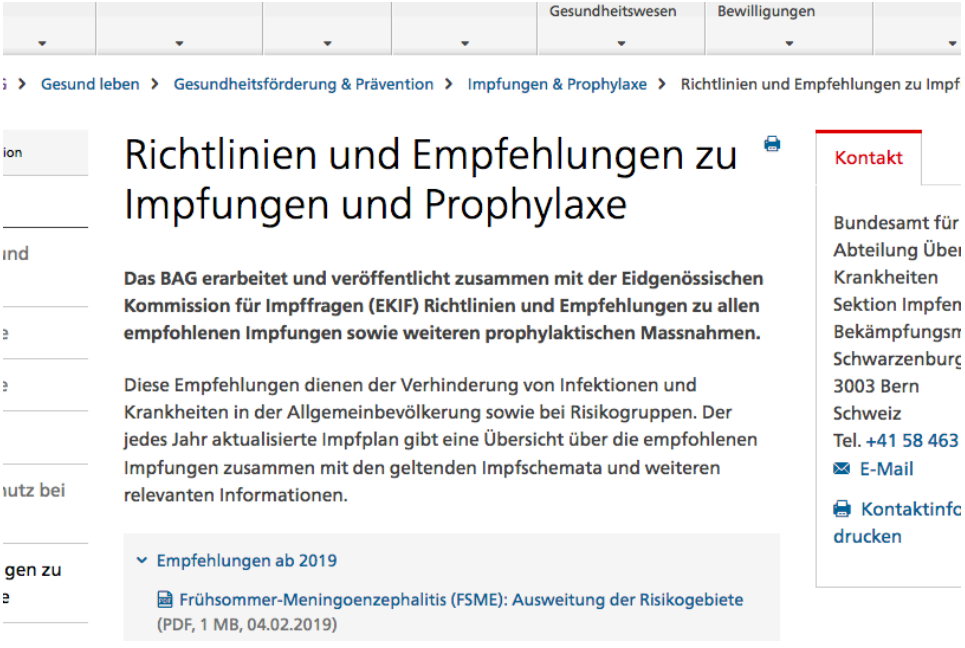
Mathias Müller



Administratives

Tutorat

- AND-3-48
- Montag 10:15 bis 12:00



Leistungsnachweise

- Übungen
- Schriftliche Prüfung

- Gewichtung: 25% Übungen, 75% Prüfung
- Rechenbeispiele:
  - Übungen 4.0, Prüfung 6.0 = Endnote 5.5
  - Übungen 5.5, Prüfung 4.0 = Endnote 4.5

Übungen

- Anzahl Übungen: 6
- Bearbeitungszeit: jeweils 2 Wochen
- Teamarbeit ist erlaubt, sofern klar deklariert
- Fragen zu Übungen an die Tutoren richten, wenn möglich im OLAT-Forum

Schriftliche Prüfung

- “Closed Book”, keine Unterlagen
- Termin: **18. Juni 2019, 16:15 bis 18:00**
- Raum: **AND-2-48**
- Achtung: 75 % der Endnote

Übungen

- Anzahl Übungen: 6
- Bearbeitungszeit: jeweils 2 Wochen
- Teamarbeit ist erlaubt, sofern klar deklariert
- Fragen zu Übungen an die Tutoren richten, wenn möglich im OLAT-Forum

Programm (siehe OLAT)

Termin	Thema	Lektüre	Übung
19.02.	Einführung; regelbasierte vs. datengetriebene Modelle	SMT Kapitel 1	
26.02.	Evaluation	SMT Kapitel 8	Übung 1
05.03.	Trainingsdaten, Vor- und Nachverarbeitung	SMT Kapitel 2	
12.03.	N-Gramm-Sprachmodelle, statistische Maschinelle Übersetzung	SMT Kapitel 3-7	Übung 2
19.03.	Grundlagen Lineare Algebra und Analysis, Numpy		
26.03.	Lineare Modelle: lineare Regression, logistische Regression		
02.04.	Neuronale Netzwerke: MLPs, Backpropagation, Gradient Descent	NMT Kapitel 1-3	Übung 3
09.04.	Word Embeddings, Recurrent neural networks	NMT Kapitel 4	
16.04.	Tensorflow und Google Cloud Platform		Übung 4
30.04.	Encoder-Decoder-Modell	NMT Kapitel 5	Übung 5
07.05.	Decoding-Strategien	NMT Kapitel 5.4	
14.05.	Attention-Mechanismus, bidirektionales Encoding, Byte Pair Encoding	NMT Kapitel 5-6	Übung 6
21.05.	Maschinelle Übersetzung in der Praxis (Anwendungen)		
28.05.	Zusammenfassung, Q&A Prüfung		
Eventuell: Gastvortrag Prof. Artem Sokolov			
04.06., Raum TBA, 16:15 bis 18:00 Uhr			
Prüfung (schriftlich)			
18.06., AND-2-48, 16.15 bis 18:00 Uhr			

University of Zurich[UZH]

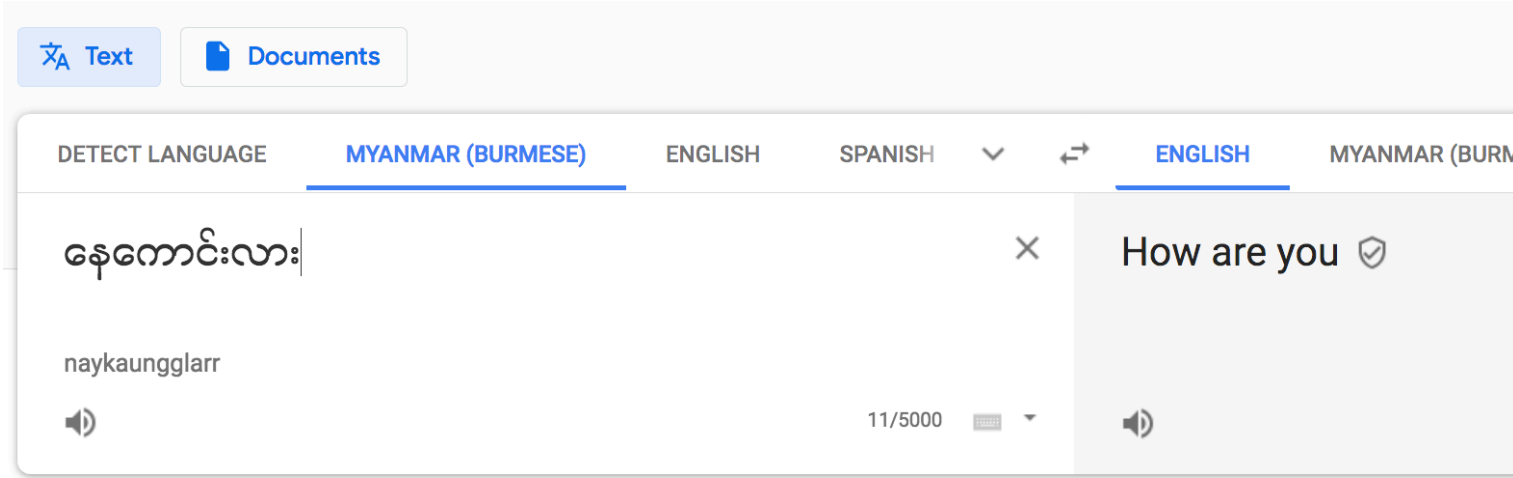
Institute of Computational Linguistics

Machine Translation

Topics of this lesson

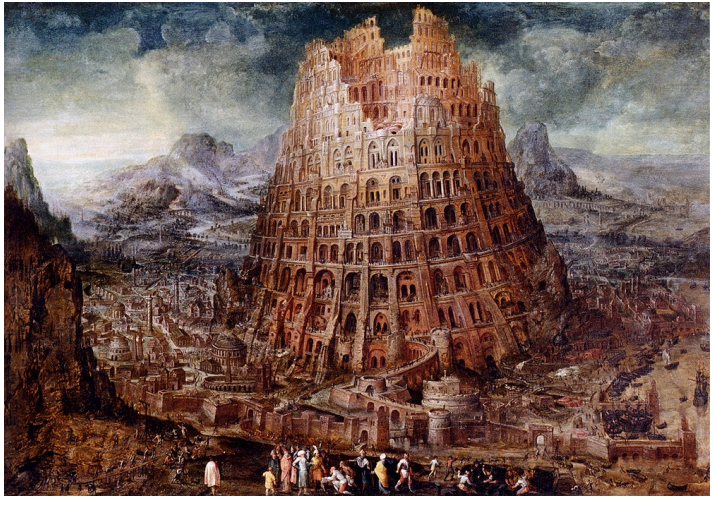
- learn what machine translation is, and appreciate the difficulty of the task
- main paradigms of machine translation
- history of machine translation (exciting!)

Well, you know what machine translation is!



Goal of machine translation

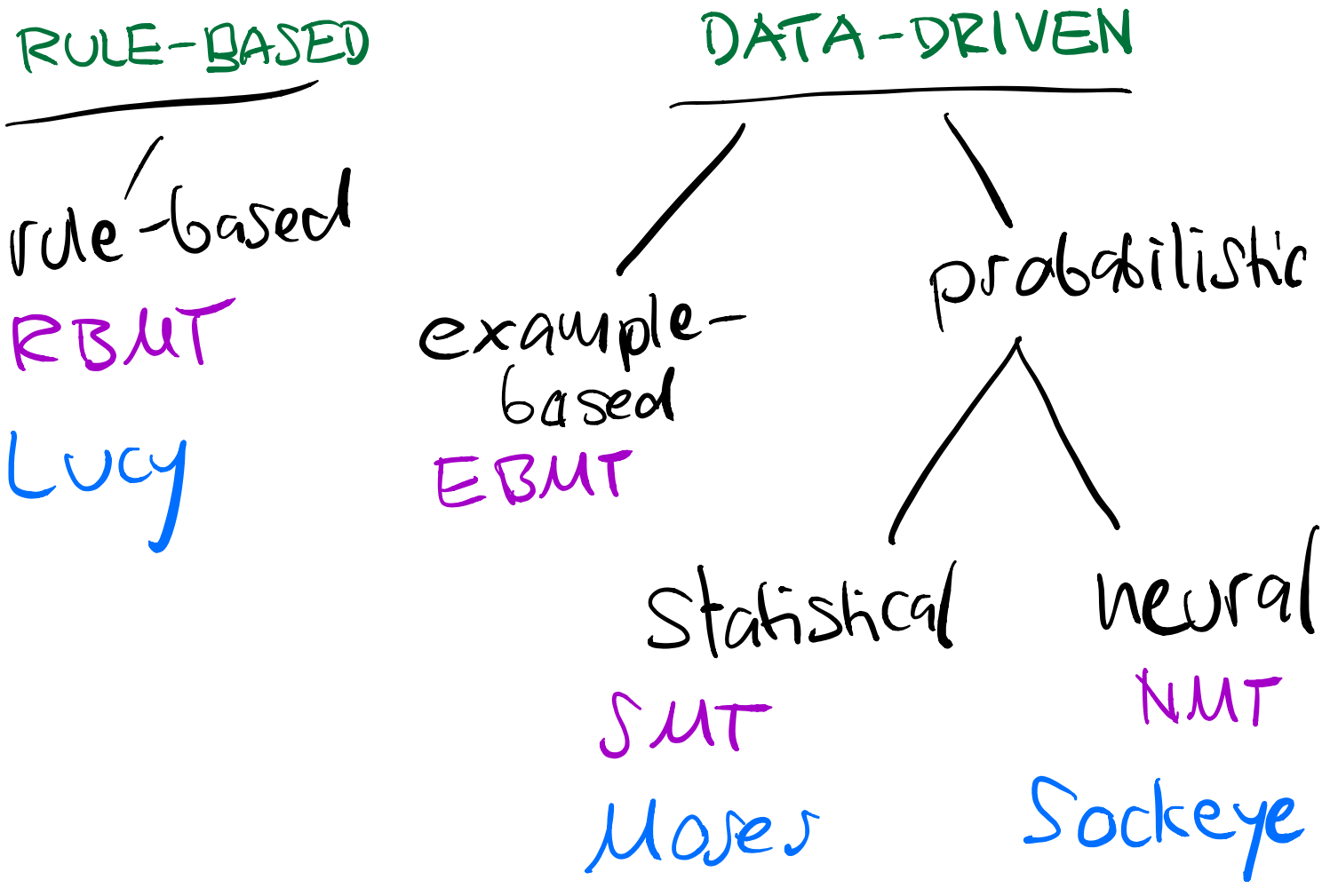
to unbabel the world



Definition for our purposes

- machine translation means translating from **1** language to **1** other language
- always about **human** languages
- always about translating **text**, no visual input, no audio signals, no handwriting

Main paradigms of machine translation



Rule-based systems

1. ADJ NN → NN ADJ
2. ADJ → "rot"
3. NN → "maison"
4. "Haus" → "maison"

Das rote Haus

↓

la maison rouge

DE FR

Data-driven systems

word-aligned corpora

Sentence-aligned corpora

Remaining system types

- example-based EBMT
- statistical SMT
- neural NMT

are data-driven.

are trained with training data

Driven by which data / what to train on?

The European Community had already agreed to phase out CFCs by 1997 and hoped that other countries would do the same.

The Protocol should be amended to reflect that situation.

But that was not enough.

The Technology and Economics Assessment Panel should be asked to assess the implications of phasing out halons, carbon tetrachloride and methyl chloroform also by 1997.

La Comunidad Europea ya había convenido en suprimir los CFC para 1997 y confiaba en que otros países hicieran lo mismo.

El Protocolo debía enmendarse para reflejar esa situación.

No obstante, eso no bastaba.

Se debería pedir al Grupo de evaluación técnica y económica que evaluara las repercusiones de la supresión gradual de los halones, el tetracloruro de carbono y el metilcloroformo también para 1997.

Blueprint for a data-driven system

```
class TranslationSystem:

  def train(self, source_sentences, target_sentences):
      # train system ...

  def translate(self, source_sentence):
      # produce translation …
      return target_sentence
```

(unit of translation typically **a sentence**)

Pseudo code for an example-based system

```
class TranslationSystem:

    def __init__(self):
        self.map = {}

    def train(self, source_sentences, target_sentences):
        for source_sentence, target_sentence in \
                zip(source_sentences, target_sentences):
            self.map[source_sentence] = target_sentence

    def translate(self, source_sentence):
        target_sentence = self.map[source_sentence]
        return target_sentence
```

Does this work in practice?

```
class TranslationSystem:

    def __init__(self):
        self.map = {}

    def train(self, source_sentences, target_sentences):
        for source_sentence, target_sentence in \
                zip(source_sentences, target_sentences):
            self.map[source_sentence] = target_sentence

    def translate(self, source_sentence):
        target_sentence = self.map[source_sentence]
        return target_sentence
```

https://padlet.com/mmueller26/xr5ryyosugpf

probabilistic
~~Statistical~~ systems (SMT + NMT)

- frame translation as a **sequence classification problem**

input

output

dog 2%
cat 20%
monster 78%

pix2pix

input "das rote Haus"

la maison verte 2.3%
la maison rouge 17.8%
la maison ... 1.7%

probabilistic

~~Statistical~~ systems

(SMT + NMT)

roughly,

1) learn probabilities from training data
2) pick the most probable translation

```
class TranslationSystem:

    def train(self, source_sentences, target_sentences):
        # estimate probabilities from training data

    def translate(self, source_sentence):
        # pick most probable translation
        return target_sentence
```

Word-based statistical machine translation

roughly,

- estimate a word dictionary from training data
- all words translated individually

TM

IBM model

```
class WordTranslationSystem:

    def train(self, source_sentences, target_sentences):

        self.estimate_translation_model(source_sentences, target_sentences)

    def translate(self, source_sentence):
        target_sentence = []
        for source_token in source_sentence:
            target_token = self.get_most_probable_translation(source_token)
            target_sentence.append(target_token)

        return target_sentence
```

Phrase-based statistical machine translation (PBSMT)

la maison → 0.37

roughly,

- estimate a phrase (= ngram) translation model and an ngram language model, from training data
- candidate translations are ranked by scores

```
class PhraseTranslationSystem:

    def train(self, source_sentences, target_sentences):

        self.estimate_translation_model(source_sentences, target_sentences)
        self.estimate_language_model(target_sentences)

    def translate(self, source_sentence):

        candidate_translations = self.generate_candidate_translations(source_sentence)
        candidate_scores = self.language_model_score(candidate_translations)

        best_translation = candidate_translations[numpy.argsort(candidate_scores)][0]

        return best_translation
```

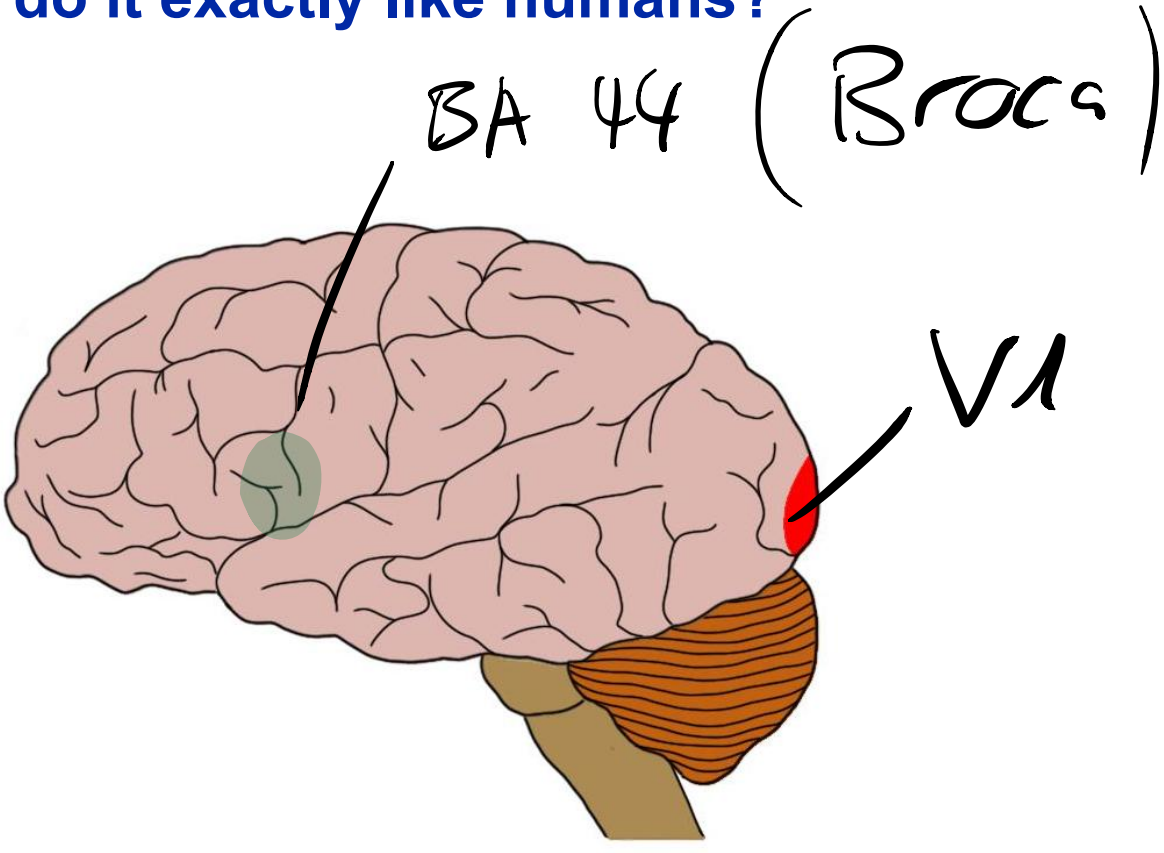
Neural machine translation (NMT)

roughly,

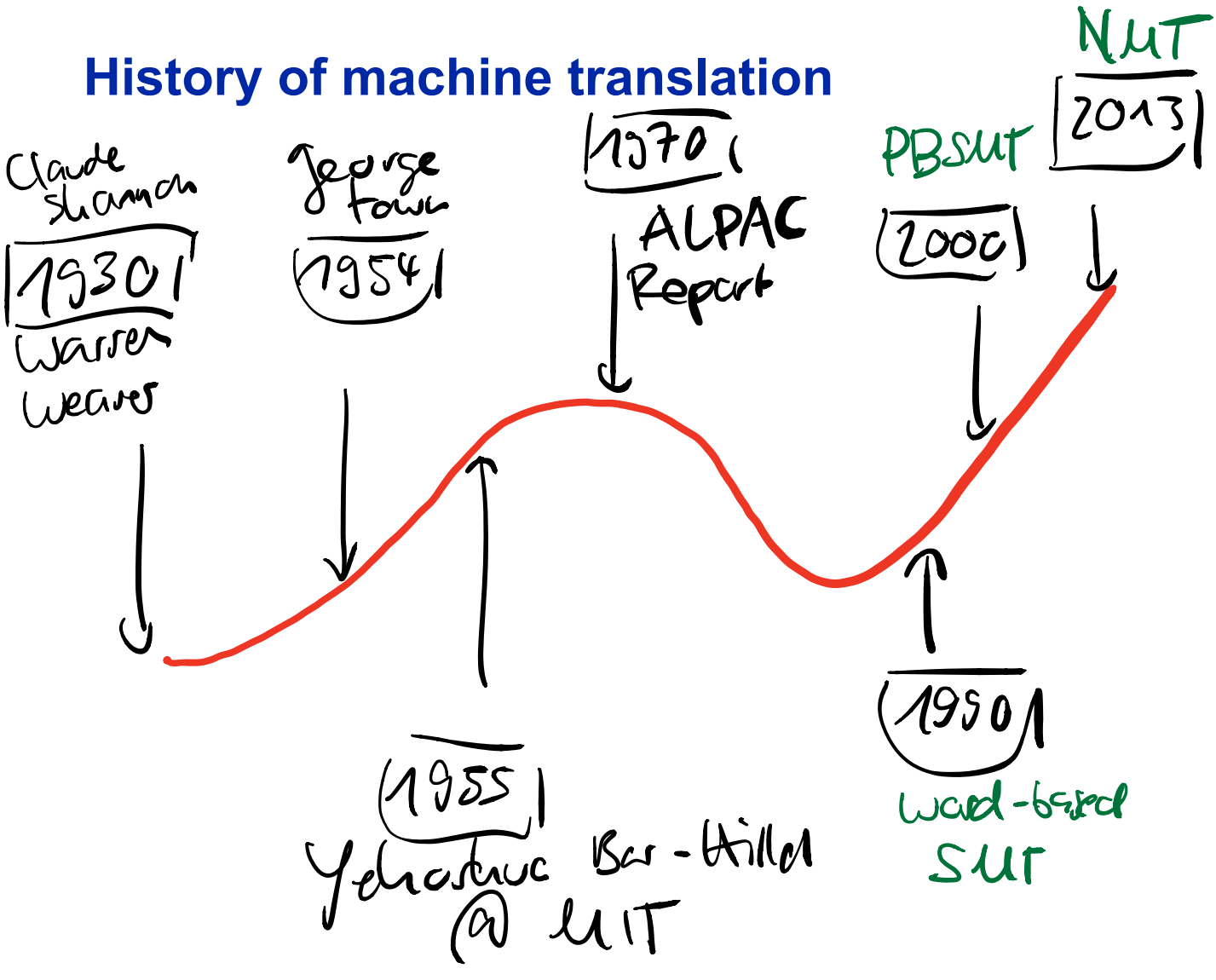
- model is a neural network with two parts: encoder and decoder
- encoder reads an entire source sentence
- decoder returns most likely next word for a translation prefix

```
class NeuralTranslationSystem:

    def train(self, source_sentences, target_sentences):

        self.estimate_encoder_and_decoder(source_sentences, target_sentences)

    def translate(self, source_sentence):
        target_sentence = []

        target_token = "<BOS>"

        encoded_source_sentence = self.encode(source_sentence)

        while True:
            target_token = self.decode(encoded_source_sentence, target_token)
            if target_token == "<EOS>":
                break
            target_sentence.append(target_token)

        return target_sentence
```

Why not do it exactly like humans?



History of machine translation



Summary

- goal of machine translation is to unbabel the world!
- main paradigms are
  rule-based (RBMT) example-based (EBMT)
  statistical (SMT) neural translation (NMT)
- EBMT, SMT and NMT rely on **training data**
- SMT and NMT estimate **probabilities** from training data

Outlook

Termin	Thema
19.02.	Einführung; regelbasierte vs. datengetriebene Modelle
26.02.	Evaluation
05.03.	Trainingsdaten, Vor- und Nachverarbeitung
12.03.	N-Gramm-Sprachmodelle, statistische Maschinelle Übersetzung
19.03.	Grundlagen Lineare Algebra und Analysis, Numpy
26.03.	Lineare Modelle: lineare Regression, logistische Regression
02.04.	Neuronale Netzwerke: MLPs, Backpropagation, Gradient Descent
09.04.	Word Embeddings, Recurrent neural networks
16.04.	Tensorflow und Google Cloud Platform
30.04.	Encoder-Decoder-Modell
07.05.	Decoding-Strategien
14.05.	Attention-Mechanismus, bidirektionales Encoding, Byte Pair Encoding
21.05.	Maschinelle Übersetzung in der Praxis (Anwendungen)
28.05.	Zusammenfassung, Q&A Prüfung
Eventuell: Gastvortrag Prof. Artem Sokolov	
04.06., Raum TBA, 16:15 bis 18:00 Uhr	
Prüfung (schriftlich)	
18.06., AND-2-48, 16.15 bis 18:00 Uhr