RUHR-UNIVERSITÄT BOCHUM

RUB

Fakultät für Philologie
Sprachwissenschaftliches Institut

Spelling normalization: How far can you get without context?

Marcel Bollmann, Stefanie Dipper, Florian Petran

Steps common to both models

- **Context-free normalization**
  - Input is always a single wordform, *without* surrounding words as context
  - Cannot resolve certain ambiguities or perform merging of input words
- **Capitalization**
  - Not explicitly modelled; input is lower-cased for both models
  - Simple heuristic: Capitalize sentence beginnings and single-letter abbreviations
  - We also tried *truecasing* using a statistical model learned from parts of Wikipedia, but found it more problematic (due to noisy output) than useful
- **Punctuation**
  - Not explicitly modelled (requires context information)
  - Removed before normalization, then re-inserted from original text afterwards
- **Training data**
  - We train only on normalizing from the 1637 to the 1888 bible translation
  - Sentence pairs are aligned using MGIZA to generate word pairs for training
  - Resolve 1:n alignments using the underscore notation:

    hyse — *hij* *ze* ⟶ `hyse` `hij_ze`

  - Merge unaligned words with their neighbours and try to find best split by using Levenshtein alignment on characters:

    so *lange* — *zolang* ⟶ `so` `zo` / `lange` `lang`

- **Lexical filtering**
  - Restrict model output by words in a lexicon
  - Lexicon: tokens from 1888 bible + Dutch part of CELEX (Baayen et al., 1995)

Bochum-1: The Norma tool (Bollmann, 2012)

1 Lexical mapping

- Look up words in a translation lexicon
- If found, replace them with the learned mapping

huylet	huilt	12
huylt	huil	3
huylt	huilt	2
huys	huis	1441
huys	huis_van	130
huys	huizes	15
huys-besorger	huisbezorger	1
...	...	...

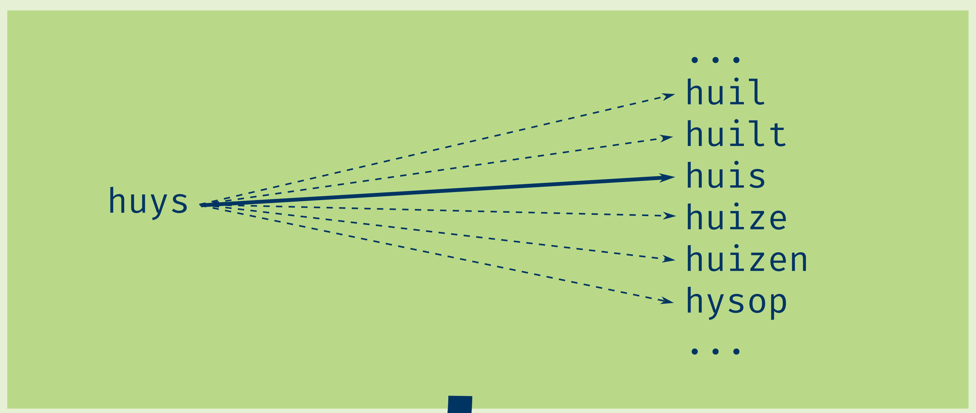
2 Rule-based algorithm

- Learn replacement rules from the training data
- Apply the most probable rules from left to right

y	⟶	ij / h _ #	10005
hy	⟶	ij / g _ #	8382
h	⟶	h / # _ u	2763
y	⟶	i / u _ s	2549
y	⟶	y / g _ p	729
y	⟶	ε / u _ r	86
uy	⟶	ε / # _ t	41
...			...

3 Weighted Levenshtein distance

- Learn Levenshtein weights from the training data
- Find lexicon word with the lowest distance

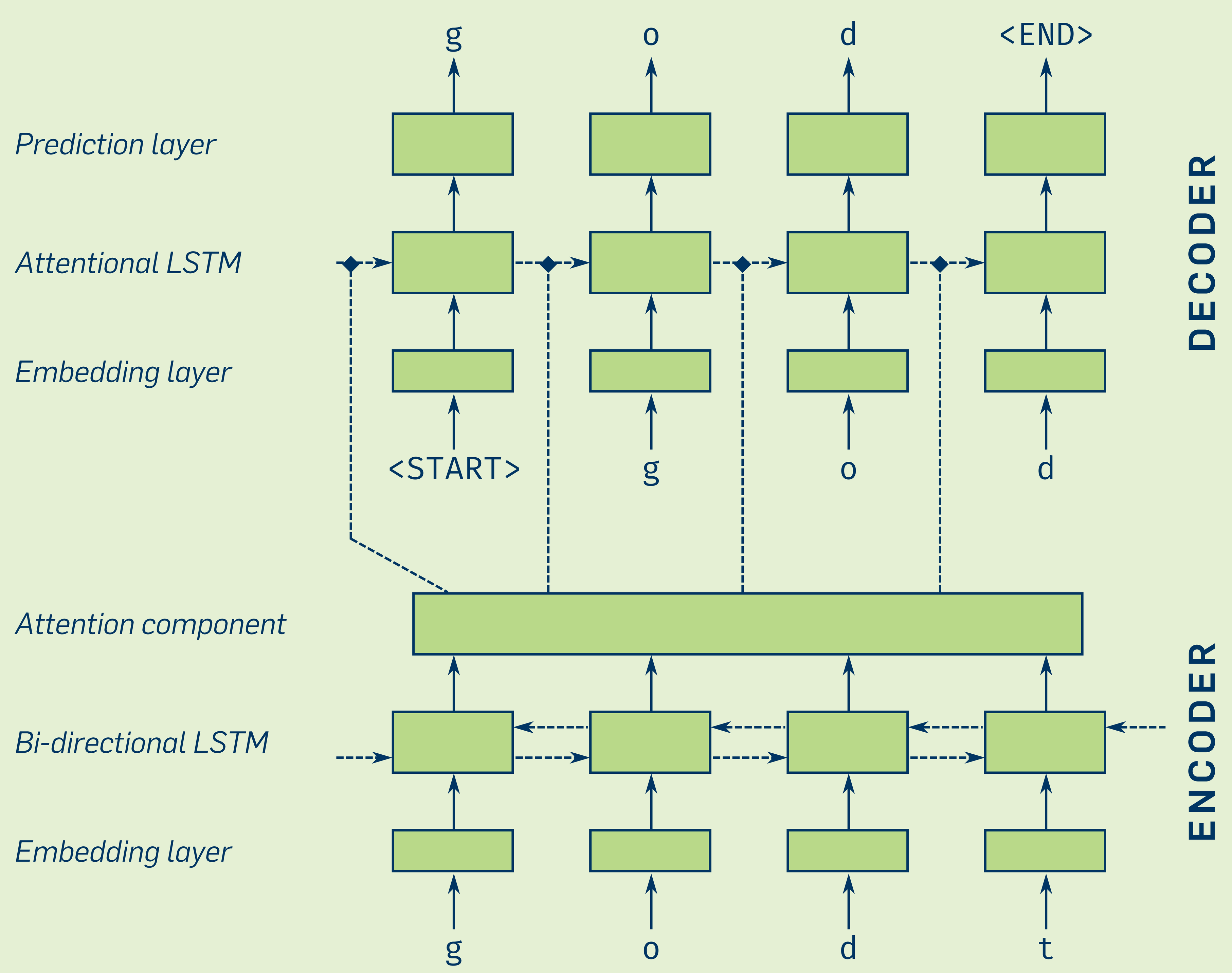


Majority voting

- Ties are resolved in order: Mapper > Rule-based > Weighted Levenshtein

Bochum-2: Encoder–decoder neural network architecture (Similar to Sutskever et al., 2014)

- Implemented using Keras (Chollet, 2015) and lots of custom code
- **Encoder**
  - Embedding layer maps characters to vectors
  - Bi-directional LSTM encodes the input sequence
  - Encoder output is fed into the decoder's hidden state using an attention mechanism (closely following Xu et al., 2015)
- **Decoder**
  - Reads partially predicted sequence (or *<START>* at the beginning), predicts next output character
  - Embedding layer maps characters to vectors
  - Attentional LSTM reads input characters, calculates new hidden state by combining old hidden state and the encoded input sequence
  - Prediction layer generates prediction, used as input for next timestep
- **Beam-search decoding**
  - Keep 5 best predictions per timestep
  - Filter possible beams using the lexicon
- **Hyperparameters**
  - Number of neurons in each layer is 256
  - Dropout = 0.2 for the LSTM inputs
  - Trained in mini-batches of 1000 tokens for a total of 10 epochs
  - Used Adam algorithm (Kingma & Ba, 2015) with learning rate = 0.003



- R. Harald Baayen, Richard Piepenbrock, & Léon Gulikers (1995). *The CELEX Lexical Database (Release 2) (CD-ROM)*. Linguistic Data Consortium, University of Pennsylvania, Philadelphia, PA.
- Marcel Bollmann (2012). *(Semi-)automatic normalization of historical texts using distances measures and the Norma tool.* Proceedings of the Second Workshop on Annotation of Corpora for Research in the Humanities (ACRH-2). Lisbon, Portugal.
- François Chollet (2015). *Keras*. `https://github.com/fchollet/keras`
- Diederik P. Kingma & Jimmy Lei Ba (2015). *Adam: A Method for Stochastic Optimization.* The International Conference on Learning Representations (ICLR). San Diego, CA.
- MGIZA. `https://github.com/moses-smt/mgiza`
- Ilya Sutskever, Oriol Vinyals, & Quoc V. Le (2014). *Sequence to Sequence Learning with Neural Networks.* Advances in Neural Information Processing Systems (NIPS), pp. 3104–3112. Montréal, Canada.
- Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, & Yoshua Bengio. *Show, Attend and Tell: Neural Image Caption Generation with Visual Attention.* JMLR Workshop and Conference Proceedings: Proceedings of the 32nd International Conference on Machine Learning, pp. 2048–2057. Lille, France.

The research reported here was financed by Deutsche Forschungsgemeinschaft (DFG), Grant No. DI 1558/4.