Automatic Annotation for Historical German

Ercong Nie

Center for Information and Language Processing (CIS),
Ludwig-Maximilians-Universität München (LMU)
nie@cis.lmu.de

June 20, 2024 Berlin

Workshop on Methods in Historical Corpus Building

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Ercong Nie

- PhD candidate at Center for Information and Language Processing (CIS) of LMU Munich.
- Supervised by PD. Dr. Helmut Schmid.
- **Master**: Computational Linguistics + Informatics at CIS, LMU.
- **Bachelor**: German + Finance at Shanghai Jiao Tong University, China.
- **Research interest**: multilingual Natural Language Processing (NLP), low-resource NLP, NLP for historical languages etc.

Outline

Lemmatization, POS Tagging, Morphosyntactic annotation, ...

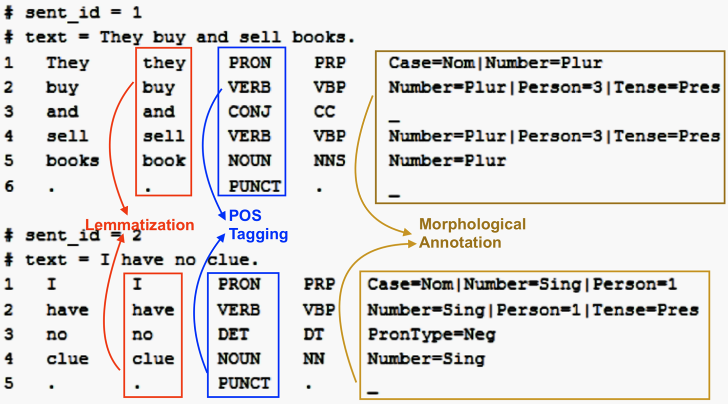


Figure: An example of linguistic annotation in the **CoNLL** format (Ishola, 2019).

Constituency Parsing

Sentence: `That cold, empty sky was full of fire and light.`

```
((S
   (NP-SBJ (DT That)
     (JJ cold) (, ,)
     (JJ empty) (NN sky) )
   (VP (VBD was)
     (ADJP-PRD (JJ full)
       (PP (IN of)
         (NP (NN fire)
           (CC and)
           (NN light) ))))
   (. .) ))
```

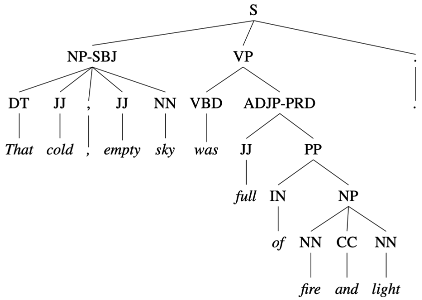


Figure: An example of constituency parse (Jurafsky and Martin).

Why we need linguistically annotated corpora of historical languages?

- form the foundation for **linguistic analysis** (language change, contact and variation, linguistic evolution of morphology, syntax, etc.).
- serve as a building block for **NLP applications**.
- enrich **interdisciplinary** cultural, literature and historical studies.

- **Corpora annotated on the token level**
  - German Reference Corpus (Referenzkorpus)[1]

- **Syntactically annotated corpora**

Id.	Name	Languages	Style	Size
DDB[2]	German Diachronic Treebank	OHG, MHG, ENHG	Tiger	8,580
ReF[3]	Reference Corpus of Early New High German: Treebank	FNHD	Tiger	∼500,000
IPCHG[4]	Indiana Parsed Corpus of Historical (High) German	OHG, MHG, ENHG	PTB	∼10,000
CHLG[5]	Corpus of Historical Low German	MLG, OLG	PTB	∼200,000

Why we want automatic annotation for historical languages?

- **Difficulties** in constructing parsed corpora for historical languages:
  - Scarcity of digital text resources,
  - High demand of linguistic expertise,
  - Large manual effort.

Why we want automatic annotation for historical languages?

- **Difficulties** in constructing parsed corpora for historical languages:
  - Scarcity of digital text resources,
  - High demand of linguistic expertise,
  - Large manual effort.

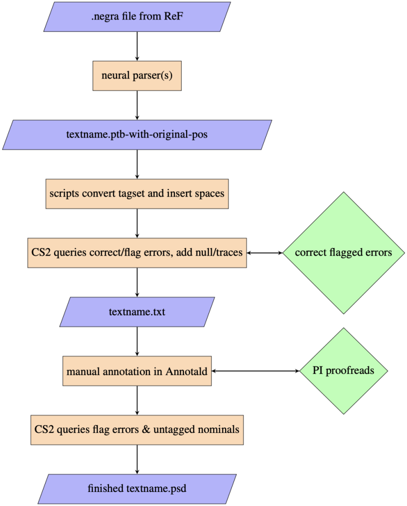
→ **Solution**: Train automatic linguistic structure annotation and analysis systems.

Automatic annotation in corpus construction

Example of automatic annotation applied to corpus construction

- **Construction of PCENHG Corpus**:

  an early new high German (ENHG) corpus released by the IPCHG (Indiana Parsed Corpus of Historical High German) team (Sapp et al., 2023).



Two Research Cases

Automatic Linguistic Annotation for Historical German Languages:

- **Case 1**: Automatic annotation of medieval lyrics with POS tags and lemmas.
- **Case 2**: Automatic constituency parsing for historical German.

Outline

1. Introduction
2. Automatic Annotation with POS Tag and Lemma
3. Automatic Constituency Parsing
4. Summary

Credits to PD. Helmut Schmid.
Errors and defects are solely my responsibility.

Task: Annotation with POS Tag and Lemma

Token	POS Tag	Lemma
do	AVD	dô
begagenda	VVFIN.Ind.Past.Sg.3	be-gègenen
imo	PPER.Masc.Dat.Sg.3	ër
min	DPOSA.Masc.Nom.Sg.*	mîn
trohtin	NA.Masc.Nom.Sg.st	truhtîn
mit	APPR	mit
inero	DPOSA.Fem.Dat.Sg.st	sîn
arngrihte	NA.Fem.Dat.Sg.st	êre-grëhte
.	$_	.

Example sentence from the Referenzkorpus Mittelhochdeutsch[6].

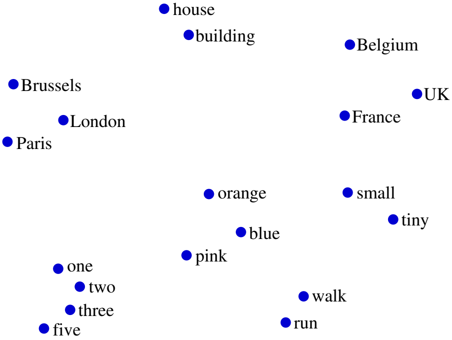
Overview

- Word Representations
- Construction of the POS Tagger
- Construction of the Lemmatizer
- Application to the Corpus *Medieval Lyrics*

Word Representations

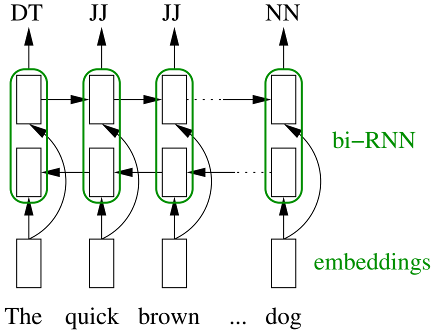
- The POS tagger is based on a neural network.
- Neural networks can only process numbers.
- So, each word should be represented as a number vector:
  (-0.7, 1.5, 12.8, -5.5, 0.2, ..., 3.5)
- These representations are part of the neural network and are trained with it.
- Similar words have similar representations.

Word Embeddings



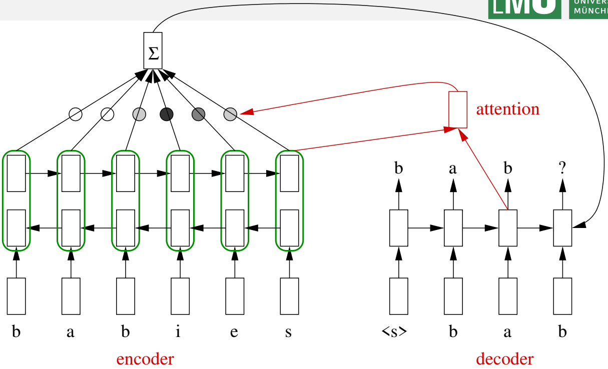
- The number vectors correspond to points in the n-dimensional space (where n is the length of the number vector).
- The representations of semantically similar words are close to each other.

Construction of the POS Tagger



The POS tagger is trained on manually annotated data.

Lemmatizer

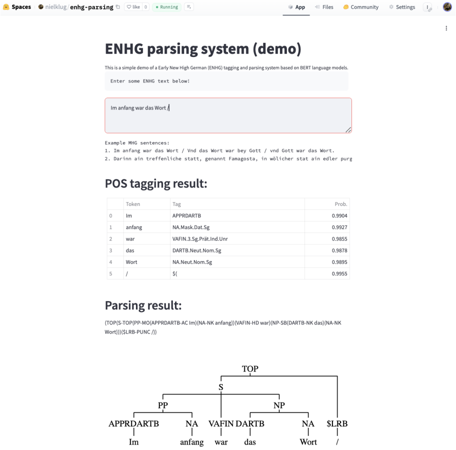


- Input: Character sequence of a word + POS tag
- Output: Character sequence of the lemma
- The encoder provides a representation in the context for each character.
- The decoder generates the lemma character one by one..
- The attention module provides a summary of the encoder representations, which depends on the current status of the decoder.

- Tagger and lemmatizer were trained on the ReM-Korpus.
- ReM was annotated with the POS tag set HiTS, which is based on the STTS (a tag set for modern German).
- The corpus Medieval Lyrics was annotated by the trained system.
- Annotations were manually checked by experts on a short text of 138 words.
- 114 of the 138 words were annotated with the correct POS tag and the correct morphosyntactic features (number, gender, case)
  $\Rightarrow$ 83% accuracy

mod. German	MHG	POS Tag	Lemma	Correction
Nachtigall	Nahtegal	ADJA.Pos.Neut.Nom.Sg.*	nahtegalw	Nomen
gutes	gůt	ADJA.Pos.Neut.Nom.Sg.*	guot	
Vögelein	vogellin	NA.Neut.Nom.Sg.st	vogellîn	
meiner	miner	DPOSA.Fem.Dat.Sg.st	mîn	
Frau	frǒwen	NA.Fem.Gen.Sg.st	vrouwe	Dativ
sollst	solt	VMFIN.Ind.Pres.Sg.2	soln	
Du	du	PPER.*.Nom.Sg.2	dû	
singen	singen	VVINF	singen	
in	in	APPR	in	
ihr	ir	DPOSA	ir	
Ohr	ore	NA.Neut.Akk.Sg.wk	ôre	
dorthin	dar	AVD	dar	
weil	sit	KOUS	sîd	
sie	si	PPER.Fem.Nom.Sg.3	ër	
hat	hat	VAFIN.Ind.Pres.Sg.3	haben	Vollverb
das	daz	DDART.Neut.Akk.Sg.*	dër	
Herz	herze	NA.Neut.Akk.Sg.wk	hërze	
mein	min	DPOSN.Neut.Akk.Sg.wk	mîn	

A demo system of annotation and parsing for Early New High German (ENHG):


Spaces nielklug/enhg-parsing like 0 Running App Files Community Settings

ENHG parsing system (demo)

This is a simple demo of a Early New High German (ENHG) tagging and parsing system based on BERT language models.

```
Enter some ENHG text below!
```

Im anfang war das Wort /

```
Example MHG sentences:
1. Im anfang war das Wort / Vnd das Wort war bey Gott / vnd Gott war das Wort.
2. Darinn ain treffenliche statt, genannt Famagosta, in wölicher stat ain edler purg
```

POS tagging result:

	Token	Tag	Prob.
0	Im	APPRDARTB	0.9904
1	anfang	NA.Mask.Dat.Sg	0.9927
2	war	VAFIN.3.Sg.Prät.Ind.Unr	0.9855
3	das	DARTB.Neut.Nom.Sg	0.9878
4	Wort	NA.Neut.Nom.Sg	0.9895
5	/	$(	0.9955

Parsing result:

(TOP(S-TOP(PP-MO(APPRDARTB-AC Im)(NA-NK anfang))(VAFIN-HD war)(NP-SB(DARTB-NK das)(NA-NK Wort)))($LRB-PUNC /))

Figure: https://huggingface.co/spaces/nielklug/enhg-parsing

Outline

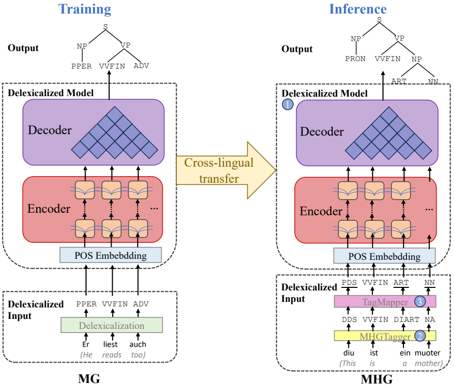
LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Our work focused on the constituency parsing of **Middle High German** (MHG):

- a historical stage of the German language that was spoken between 1050 and 1350.
- the linguistic predecessor of Modern German (MG).

Motivation of the Delexicalization Method:

- The continuity in the process of language evolution gives rise to **linguistic similarities** between **MG** and **MHG**.
    - Similar sentence structure
    - Similar word order
- **Rich resources of MG** texts with syntactic annotations.
    - Tiger Corpus (Smith, 2003)



Delexicalization Parsing System for MHG

LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

The delexicalization parsing system for MHG comprises three modules:

- **POS Tagger**
  - Annotates a sequence of MHG tokens with POS and morphological tags.
  - Trained on the ReM corpus using RNNTagger (Schmid, 2019).

Delexicalization Parsing System for MHG

The delexicalization parsing system for MHG comprises three modules:

- **POS Tagger**
  - Annotates a sequence of MHG tokens with POS and morphological tags.
  - Trained on the ReM corpus using RNNTagger (Schmid, 2019).
- **Tag Mapper**

Mapping tags from the HiTS tag set (used for ReM) to STTS tag set (used for MG treebanks).

MHD-Tag	**MD-Tag**
CARDD	CARD
DDART	ART
NA	NN

The delexicalization parsing system for MHG comprises three modules:

- **POS Tagger**
  - Annotates a sequence of MHG tokens with POS and morphological tags.
  - Trained on the ReM corpus using RNNTagger (Schmid, 2019).

- **Tag Mapper**

  Mapping tags from the HiTS tag set (used for ReM) to STTS tag set (used for MG treebanks).

MHD-Tag	**MD-Tag**
CARDD	CARD
DDART	ART
NA	NN

- **Delexicalized Parser**
  - Based on the Berkeley Neural Parser (Benepar) (Kitaev and Klein, 2018)
  - Trained on the Tiger Treebank (50,474 MG parse trees)

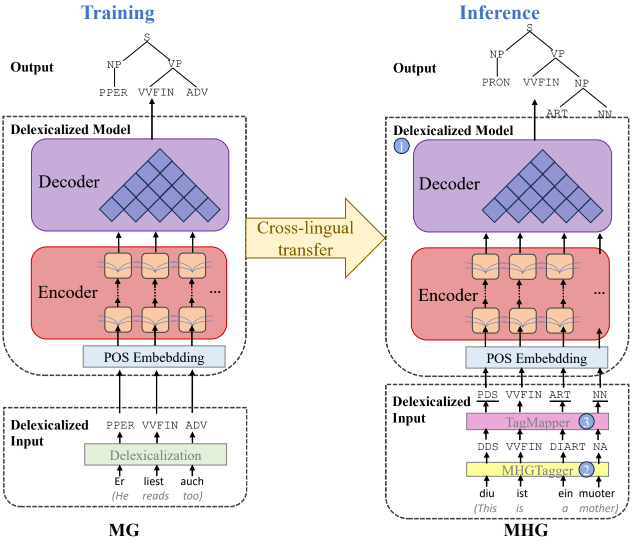


Figure: Overview of the cross-lingual delexicalized parsing system for MHG

Dataset

- Training set: Tiger Treebank (MG)
- Test set: DDB (MHG)

Baselines

- **Vanilla Benepar**: performing a vanilla zero-shot cross-lingual transfer, training a Benepar model directly on MG treebanks without the delexicalization.
- **Tetra-Tagging with PLMs**: a technique reducing constituency parsing to sequence labeling (Kitaev and Klein, 2020)
  - **gBERT**: Tetra-Tagging with the German BERT model (Chan et al., 2020)
  - **mBERT**: Tetra-Tagging with the multilingual BERT model (Devlin et al., 2019)

- Calculated by comparing the constituents of **model-generated** parse tree and the **gold standard** parse tree.
- E.g.: *The cat sat on the mat*

Gold standard parse tree:

```
(S (NP (DT The) (NN cat))
(VP (VBD sat) (PP (IN on)
(NP (DT the) (NN mat)))))
```

Extracted constituents:

- (S, 0, 6)
- (NP, 0, 1)
- (VP, 2, 6)
- (PP, 3, 6)
- (NP, 4, 5)

Predicted parse tree:

```
(S (NP (DT The) (NN cat))
(VP (VBD sat) (VP (IN on)
(NP (DT the) (NN mat)))))
```

Extracted Constituents:

- (S, 0, 6)
- (NP, 0, 1)
- (VP, 2, 6)
- (**VP**, 3, 6)
- (NP, 4, 5)

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

- Calculated by comparing the constituents of **model-generated** parse tree and the **gold standard** parse tree.
- E.g.: *The cat sat on the mat*

Gold standard parse tree:

```
(S (NP (DT The) (NN cat))
(VP (VBD sat) (PP (IN on)
(NP (DT the) (NN mat)))))
```

Predicted parse tree:

```
(S (NP (DT The) (NN cat))
(VP (VBD sat) (VP (IN on)
(NP (DT the) (NN mat)))))
```

Extracted constituents:

- (S, 0, 6)
- (NP, 0, 1)
- (VP, 2, 6)
- (PP, 3, 6)
- (NP, 4, 5)

Extracted Constituents:

- (S, 0, 6)
- (NP, 0, 1)
- (VP, 2, 6)
- (**VP**, 3, 6)
- (NP, 4, 5)

$$\text{Precision} = \frac{\#\text{Correct Constituents}}{\#\text{Total Predicted Constituents}} = \frac{4}{5} = 0.8$$

$$\text{Recall} = \frac{\#\text{Correct Constituents}}{\#\text{Total Gold Standard Constituents}} = \frac{4}{5} = 0.8$$

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

(*harmonic mean of precision and recall*)

Main results

	Recall		Precision		FScore		CM	
	MG	MHG	MG	MHG	MG	MHG	MG	MHG
Baselines								
Vanilla Benepar	84.18	34.41	87.57	44.40	85.84	38.77	45.80	0.00
Tetra-gBERT	**86.31**	23.20	**88.19**	29.53	**87.24**	25.98	**51.70**	3.12
Tetra-mBERT	60.68	19.69	65.61	23.25	63.15	21.32	21.35	0.00
Our proposed method								
Dexparser	81.39	**64.72**	84.89	**70.19**	83.10	**67.34**	39.03	**12.50**

Table: Parsing performance of different cross-lingual transfer methods. **CM** refers to "complete match" The best value of each column is indicated in **bold**.

- Dexparser demonstrates substantial advantages in parsing MHG compared to other baselines.
- Dexparser also achieves comparable results on MG.

Ablation Study

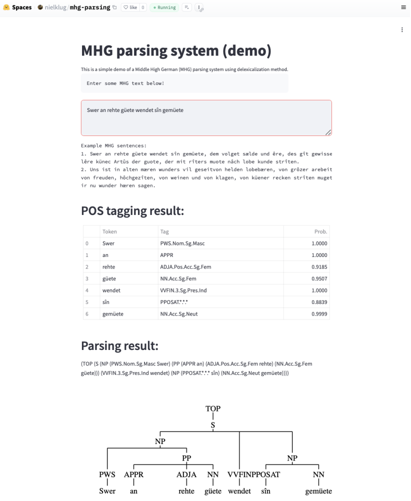
	Recall	Precision	FScore	CM
Delexicalized parser using gold tags	**66.18**	**71.17**	**68.59**	**14.58**
- *using predicted tags*	64.72	70.19	67.34	12.50
- *without mapping*	59.16	68.82	63.63	7.29
- *without morphological information*	48.66	65.38	55.8	9.28

Table: The MHG parsing results with delexicalized parser in the ablation study.

- **Quality of POS annotation**, **tag set mapping** and **annotation of morphological information** collectively contribute to the performance of the delexicalization parser on MHG.

A Demo for MHG

A demo system of annotation and parsing for Middle High German (MHG):



Spaces nielklug/mhg-parsing like 0 Running

MHG parsing system (demo)

This is a simple demo of a Middle High German (MHG) parsing system using delexicalization method.

```
Enter some MHG text below!
```

Swer an rehte güete wendet sîn gemüete

```
Example MHG sentences:
1. Swer an rehte güete wendet sîn gemüete, dem volget sælde und êre, des gît gewisse lêre künec Artûs der guote, der mit rîters muote nâch lobe kunde strîten.
2. Uns ist in alten mæren wunders vil geseitvon helden lobebæren, von grôzer arebeit von freuden, hôchgezîten, von weinen und von klagen, von küener recken strîten muget ir nu wunder hæren sagen.
```

POS tagging result:

	Token	Tag	Prob.
0	Swer	PWS.Nom.Sg.Masc	1.0000
1	an	APPR	1.0000
2	rehte	ADJA.Pos.Acc.Sg.Fem	0.9185
3	güete	NN.Acc.Sg.Fem	0.9507
4	wendet	VVFIN.3.Sg.Pres.Ind	1.0000
5	sîn	PPOSAT.*.*.*	0.8839
6	gemüete	NN.Acc.Sg.Neut	0.9999

Parsing result:

(TOP (S (NP (PWS.Nom.Sg.Masc Swer) (PP (APPR an) (ADJA.Pos.Acc.Sg.Fem rehte) (NN.Acc.Sg.Fem güete))) (VVFIN.3.Sg.Pres.Ind wendet) (NP (PPOSAT.*.*.* sîn) (NN.Acc.Sg.Neut gemüete))))

Figure: `https://huggingface.co/spaces/nielklug/mhg-parsing`

Outline

Summary

1. Automatic linguistic annotation is helpful in building corpora for historical language studies.
2. **Research Case 1**: Character-based RNNs for POS tagging and lemmatization of medieval lyrics
3. **Research Case 2:** Delexicalized parser for middle high German.

Thanks for your attention!

Branden Chan, Stefan Schweter, and Timo Möller. 2020. German's next language model. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6788–6796, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

lájídé Ishola. 2019. *Universal Dependencies for Yorùbá*. Ph.D. thesis.

Daniel Jurafsky and James H Martin. Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition.

Nikita Kitaev and Dan Klein. 2018. Constituency parsing with a self-attentive encoder. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2676–2686, Melbourne, Australia. Association for Computational Linguistics.

Nikita Kitaev and Dan Klein. 2020. Tetra-tagging: Word-synchronous parsing with linear-time inference. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6255–6261, Online. Association for Computational Linguistics.

References II

Christopher Sapp, Daniel Dakota, and Elliott Evans. 2023. Parsing early New High German: Benefits and limitations of cross-dialectal training. In *Proceedings of the 21st International Workshop on Treebanks and Linguistic Theories (TLT, GURT/SyntaxFest 2023)*, pages 54–66, Washington, D.C. Association for Computational Linguistics.

Helmut Schmid. 2019. Deep learning-based morphological taggers and lemmatizers for annotating historical texts. In *Proceedings of the 3rd international conference on digital access to textual cultural heritage*, pages 133–137.

George Smith. 2003. A brief introduction to the tiger treebank, version 1. Technical report, Universität Potsdam.

Outline

Problem of unknown words

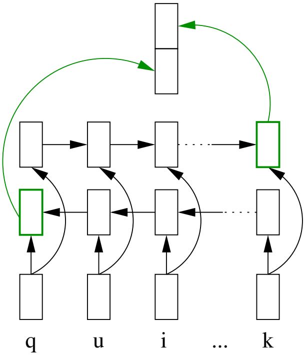
- **Challenge:** Strong graphmatic variation in historical German
  tuon, dun, doyn, thuon, tuen, tvon, tûon, tuͦn, tvͦn, toͮn
- Supposing we saw the word tuon in the training data, but did not see the word tvon.
- Which representation should we use for tvon?
- Because *u* is usually replaced by *v*, tuon and tvon should have similar representations.

⇒ Computing the word representations from the **character sequence** (instead of word sequence).

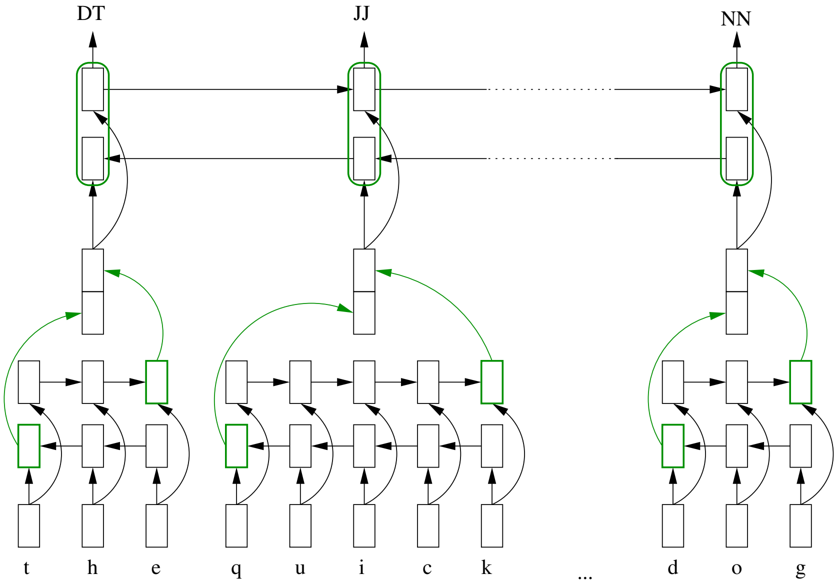
Character-based word representations

- Each character is represented by a number vector.
- The vector sequence is processed by a bidirectional RNN.
- The last representations of each direction are collected.

⇒ character-based word representations



The Whole Net



Advantages

LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

The character-based neural network

- learns regular writing variations
  e.g. u $\leftrightarrow$ v, uo $\rightarrow$ ů etc.
- generalizes from words to their possible writing variations
  tuon $\rightarrow$ thv̊n
- provides good word representations for unseen words