Balloon

*B*alloon *A*pplication for *L*ow effort *L*exicon creation

Uwe Reichel

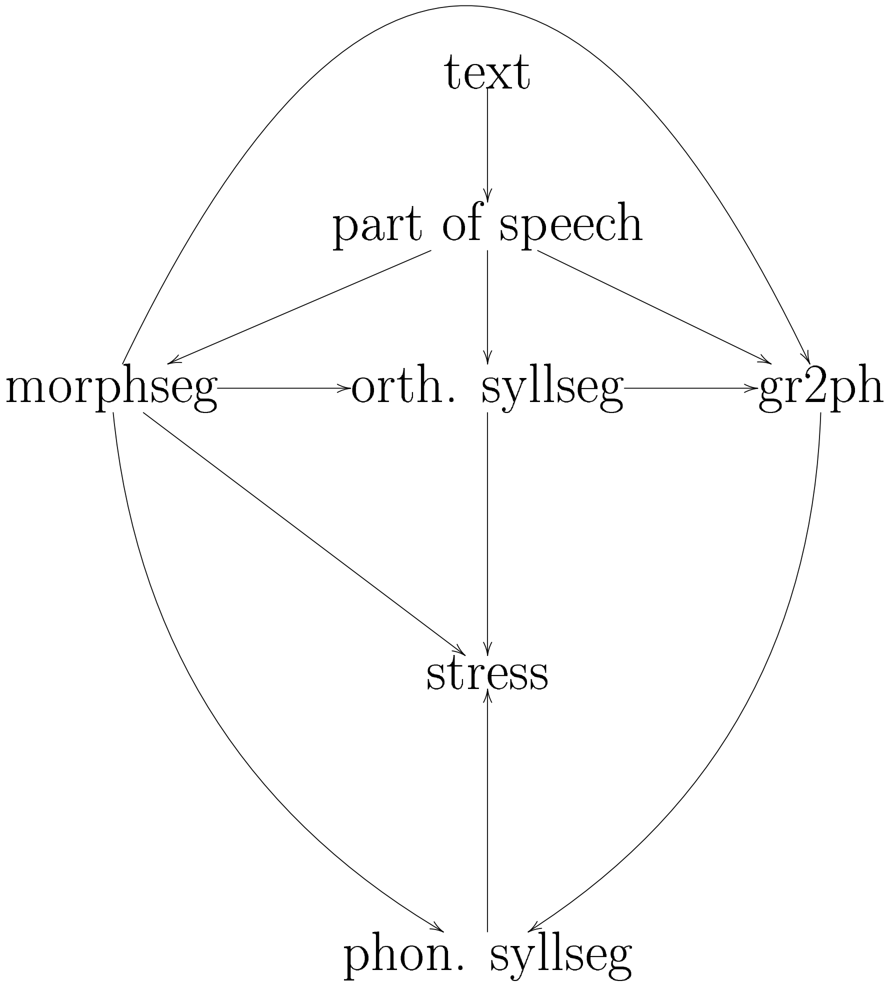
Department of Phonetics and Speech Communication

`reichelu@phonetik.uni-muenchen.de`

Input and Output

- **Input** : German text
- **Output**
  - Part-of-Speech Tags
  - Morphological Segmentation
  - Orthographical Syllable Segmentation
  - Grapheme to Phoneme Conversion
  - Phonologic Syllable Segmentation
  - Word Stress Assignment

Information Flow



text

part of speech

morphseg

orth. syllseg

gr2ph

stress

phon. syllseg

Text Preprocessing

- Tokenizer based on regular expressions (detection of ordinal numbers, abbreviations, etc.)

- Transducer converts digit numbers to letters

- Local Grammar for appropriate inflectional ending of ordinal numbers

Part-of-Speech Tagging

- **Generalization of a Markov model part-of-speech (POS) tagger:** replacing the $P(w|t)$ emission probabilities of word w given tag t by a linear interpolation of tag emission probabilities given a list of representations of w
- **Word Representation:** string suffix of word cut off at a local maximum of backward successor variety
- **What for?** retrieval of linguistically meaningful string suffixes, that may relate to certain POS labels, without the need of linguistic knowledge (language independence, addressing out of vocabulary problem)

Basic Form of a Markov POS Tagger (Jelinek, 1985)

- **Estimate for most probable tag sequence $\hat{T}$ given word sequence W**

$$\begin{aligned}\hat{T} &= \max_{T}\Big[P(T|W)\Big] \\ &= \max_{T}\Big[P(T)P(W|T)\Big] \qquad \text{(Bayes, } P(W) \text{ constant)}\end{aligned}$$

- **Simplifying Assumptions**
  - Probability of word w_i depends only on its tag t_i
  - Probability of tag t_i depends only on a limited tag history

$$\hat{T} = \max_{t_1 \ldots t_n}\left[\prod_{i=1}^{n} P(t_i|\text{t-history})P(w_i|t_i)\right]$$

- **Retrieval of $\hat{T}$ using the Viterbi algorithm**

Generalisations of the Basic Model

- **by linear interpolation**
- **replacing** $P(t_i|\mathbf{t\text{-}history})$ **by** $\sum_j u_j P(t_i|\mathbf{t\text{-}history}_j)$
- **replacing** $P(w_i|t_i)$ **by** $\frac{P(w_i)}{P(t_i)} \sum_k v_k P(t_i|\mathbf{w\text{-}representation}_k)$
  (incl. reapplication of Bayes formula)

$$\hat{T} = \max_{t_1 \ldots t_n} \Big[\prod_{i=1}^{n} \frac{1}{P(t_i)} \sum_j u_j P(t_i|\text{t-history}_j) \sum_k v_k P(t_i|\text{w-representation}_k) \Big]$$

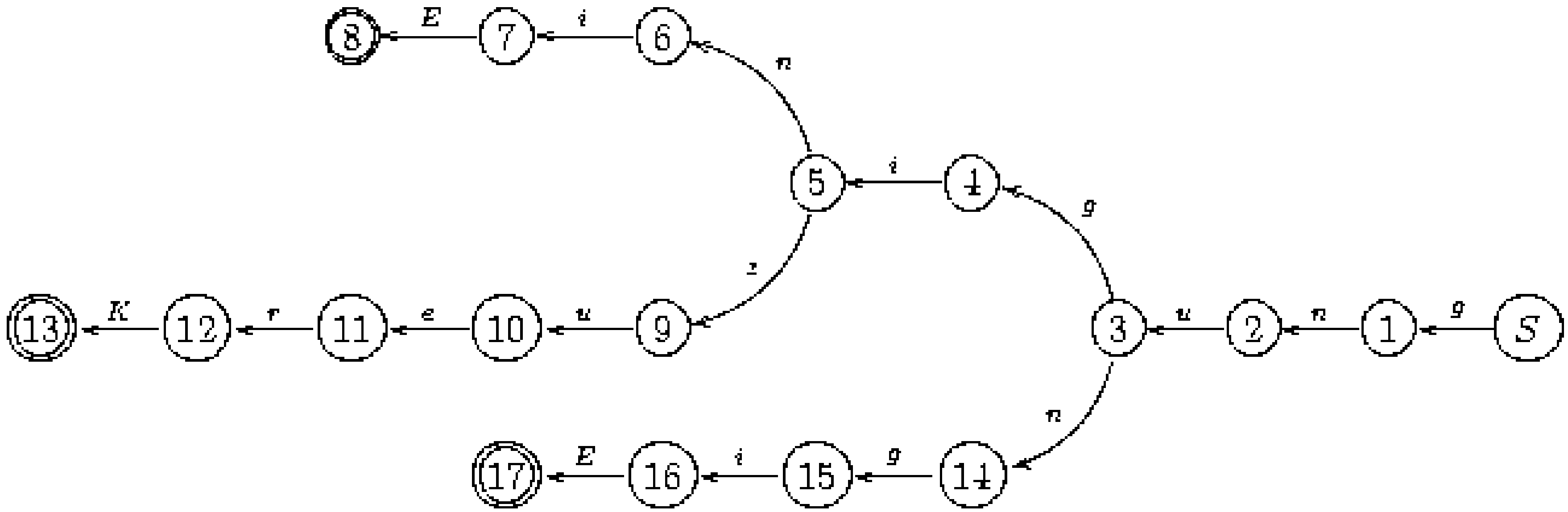
- **calculation of interpolation weights** u_j **and** v_k **via the EM algorithm**

Word Representation (I)

- suffixes are determined by **Weighted Backward Successor Variety (SV)**
- **SV** of a string: number of different characters that follow it in given lexicon
- **Backward SV:** SV‘s are calculated from reversed strings in order to separate linguistically meaningful suffixes
- **Weighting:** SV‘s are weighted w.r.t. mean SV at the corresponding string position to eliminate positional effects
- lexicon of reversed words represented in form of a trie (see next sheet)
- SV at given state: number of transitions to other states
- **Usage:** treat SV peaks as morpheme boundaries (cf. Peak and Plateau algorithm (Nascimento and da Cunha, 1998))

Word Representation (II)

- Lexicon Trie (reversely) storing the entries *Einigung*, *Kreuzigung* and *Eignung*



- The SV maxima at nodes 3 and 5 correspond to the boundaries of the morphemes *ung* and *ig* respectively

Data and Results

- **Data:** 382402 tokens tagged by the IMS Tree Tagger (Schmidt, 1995) and partially hand corrected; 85 % used for training, 15 % for testing
- **Classes:** 54 different POS tags (Tree Tagger inventory)
- **Results:**

	accuracy	κ
Baseline Taggers:		
Unigram	89.61 %	0.89
lin. interpolated Trigram	93.22 %	0.93
New Tagger:		
Trigram, word repr.	95.36 %	0.95

- This study's tagger significantly outperforms the baseline taggers (two tailed McNemar test, $p = 0.001$)
- erroneous data probably affects accuracy (e.g. finite vs. infinite verbs)

Morphological Segmentation

Input: POS labeled text

Lexicon construction

- lexicon initially comprises grammatical morphemes
- lexicon expansion by input data, applying
  - **stemming** by pattern matching and distributional analysis
  - **allomorph generation:** e.g. by applying ablaut paradigms

Segmentation Algorithm

- divide each type w recursively into string prefixes and suffixes from left to right until a permitted segmentation is achieved or until the end of w is reached.

- in the course of the recursion, a boundary dividing the current string in prefix and suffix is accepted if (i) the prefix is found in the lexicon, (ii) there exists a permitted segmentation for the suffix or (if not) the suffix is found in the lexicon, (iii) the sequence ‘prefix class $+$ class of first suffix segment’ is not in conflict with German morphotactics and (iv) the class of the last suffix is in correspondence with w’s POS.

Morphological Segmentation: Evaluation

Evaluation

- random sample: 2000 word types
- average number of morphemes: 2.63
- counting omissions and false insertions; displacement punished by one omission and one insertion
- **Recall:** 95.05 %
- **Precision:** 97.75 %
- **Word accuracy:** 91.60 %

Orthographic Syllable Segmentation

- done by C4.5 decision tree (Quinlan, 1993)
- 3 predicted classes: boundary following (y)/ not following (n)/ ambisyllabicity (a)
- **Features** (within 7-grapheme window): grapheme, morph. boundary relevant for syllabification, etc.
- **Evaluation** (12073 word types; 65 % train, 22 % develop, 13 % test):

	classified as			accuracies	precision	recall
class	y	a	n	98.76/91.16		
y	6729	–	130	98.10	98.3	98.1
a	1	443	19	95.68	97.1	95.7
n	117	13	15118	99.15		

Grapheme to Phoneme Conversion

- done by C4.5 decision tree
- **Data:** 18430 word types from Phonolex; 65 % training, 22 % developement, 13 % test
- **Features** (within 7-grapheme window): as in syllable module + position within syllable, within lexical/ functional morpheme etc.
- **Evaluation:**
  - **Word accuracy:** 84.88 %
  - **Normalized Mean Levenshtein distance:** 0.026
- significantly better than rule based P-TRA (76.36 %, 0.038) and data driven model of Daelemans and van den Bosch (79.28 %, 0.033)

Phonologic Syllable Segmentation

- **Algorithm:**
  1. split phone string at local sonority minima
  2. fine adjustment of boundaries on the basis of syllable phonotactics (Kohler, 1995) and morpheme boundaries relevant for syllabification

- **Example:** fE6hEltnIs $\xrightarrow{1.}$ fE6.hEl.tnIs $\xrightarrow{2.}$ fE6.hElt.nIs

- **Evaluation:**
  - random sample: 2000 phoneme string types
  - **Precision:** 97.3 %; **Recall:** 97.4 %; **String accuracy:** 94.5 %
  - errors partly result from mistakes of other modules

Word Stress Assignment

- done by C4.5 decision tree for simplex forms

- **Features:** syllable weight, position wrt landmark syllables, length of head and coda, nucleus characteristics, within lexical/ functional morpheme etc.

- **Evaluation** (for 13341 simplex word types; 65 % train, 22 % develop, 13 % test):

- **accuracies:** 94.85 % (syllables) 89.50 % (words)
  stress recall: 95.86 %
  stress precision: 96.32 %

- distribution of primary and secondary stress within compounds: 2 part compounds and 3 part compounds with lexicalized pair (retrieved via cooccurence counts) get primary stress on first part.