ICFHR 2014, Crete Island, Greece, September 1-4, 2014

Open-lexicon Language Modeling Combining Word and Character Levels

M. Kozielski, M. Matysiak, P. Doetsch, R. Schlueter, H. Ney

Human Language Technology and Pattern Recognition
Computer Science Department
RWTH Aachen University
D-52056 Aachen, Germany

Motivation: Recognition in Context

- **use language model (LM) in Bayes decision rule (for ASR and OCR):**
  for observation sequence $x_1^T := x_1...x_t...x_T$, find word sequence $w_1^N := w_1...w_n...w_N$:

$$x_1^T \rightarrow \hat{w}_1^N(x_1^T) \; := \; \underset{w_1^N}{\operatorname{argmax}} \{ p(w_1^N) \cdot p(x_1^T | w_1^N) \}$$

- **perplexity PP: best measure of context constraints (theory and experience)**
  = inverse of the geometric mean of LM prior $p(w_1^N)$ = 'effective vocabulary size'

$$\log PP \; := \; -1/N \cdot \log p(w_1^N) = -1/N \cdot \sum_{n=1}^{N} \log p(w_n | h_n)$$

- **problems:**
  - **OOV words: out of vocabulary (=lexicon)**
  - **more suitable units: characters rather than words**
  - **how to build a good language model for an open lexicon?**

Related Work

approaches to OOV in recognition:

- **Decompose words into characters** **[Bazzi 1999].**
- **Decompose words into sub-word units** **[Creutz 2007, Shaik 2011].**
- **Use mixed language models** **[Vertanen 2008, Rastrow 2009].**
- **Use filler models** **[Bazzi 2000, Hazen 2001].**
- **Combine of word- and character-level language models** **[Kozielski 2013].**

this paper:

- **there are well-established methods for closed-lexicon LMs**
- **question:** ***How can be build an open-lexicon language model and preserve the closed-lexicon LM probabilities?***

From Words to Characters

- **interpret the word sequence as a character sequence:**

$$c_1^M := c_1 ... c_m ... c_M$$

  with a blank symbol to separate words

- **advantages:**
  - **no OOV problem anymore; every character sequence can be recognized!**
  - **error rate should be measured at character level, too!**
    (problem with word level: long vs. short words!)
  - **perplexity at character level is always well defined and comparable!**

- **definition of character perplexity PP_c:**

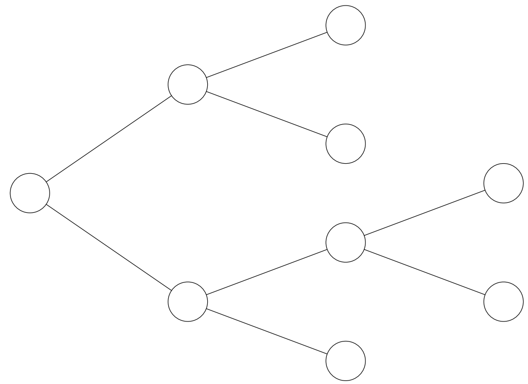
$$\log PP_c \ := \ -1/M \cdot \log p(c_1^M) = -1/M \cdot \sum_{m=1}^{M} \log p(c_m | h_m^c)$$

- **consider a closed lexicon:**
  what is the relation between word and character level?

Perplexity: From Words to Characters

- **each word has a representation as a character sequence (+ blank!):**

$$w \rightarrow \hat{c}(w) = c_1^{J_w}(w) = c_1(w), ..., c_j(w), ..., c_{J_w}(w)$$



- **organize all words as a lexical prefix tree**
- **use a closed-lexicon LM $p(w|h)$ and push the probability mass $p(w|h)$ from leaves to root and compute the character-based LM $p_c(\hat{c}(w)|h)$**

- **identity:**

$$p_c(\hat{c}(w)|h) := p(w|h) = \prod_{j=1}^{J_w} p_c(c_j(w)|c_0^{j-1}(w), h)$$

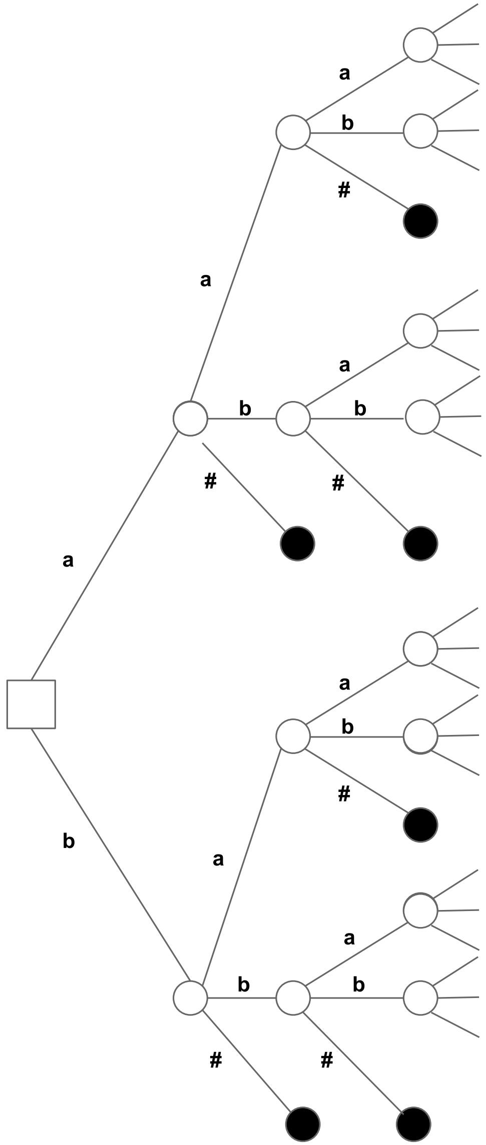
- **perplexities at word and character level: for a word sequence w_1^N and character sequence c_1^M:**

$$\log PP = M/N \cdot \log PP_c$$

- **advantage of character level: all types of LMs are now comparable!**

Open Lexicon: From Words to Characters

- **example of a simple alphabet: a, b, # (for ’blank’)**
- **organize all character sequences as a lexical prefix tree**
- **associate a conditional distribution $p_c(c_j|c_0^{j-1})$ with each interior node**



Open Lexicon: From Words to Characters

starting point:
a closed-lexicon LM $p(w|h)$ with lexicon V and unknown symbol (OOV) U:

$$p(U|h) \;:=\; \sum_{w \notin V} p(w|h) \qquad\qquad 1 - p(U|h) = \sum_{w \in V} p(w|h)$$

principles for an open-lexicon LM:

- **use an additional character-based language model (n-gram model) that allows ANY 'word' w with character sequence $\hat{c}(w) = c_1^J$:**

  $$p(\hat{c}(w)) = p(c_1^J) = \prod_{j=1}^{J} p(c_j | c_0^{j-1})$$

  note: model includes in-lexicon words and is independent of history h

- **for in-lexicon words w:**
  preserve the probabilities of closed-lexicon LM $p(w|h)$
- **for out-of-vocabulary words $w = c_1^J$:**
  re-distribute the probability mass $p(U|h)$ using $p(\hat{c}(w))$

Open Lexicon: Combination of Word and Character Levels

- **combination by backoff (V: closed lexicon):**

$$q(w|h) = \begin{cases} p(w|h) & \textbf{if } w \in V \\ p(U|h) \cdot p(\hat{c}(w)) & \textbf{if } w \notin V \end{cases}$$

normalization: model is deficient!

- **combination by sum:**

$$\begin{aligned} q(w|h) &= \begin{cases} p(w|h) + p(U|h) \cdot p(\hat{c}(w)) & \textbf{if } w \in V \\ p(U|h) \cdot p(\hat{c}(w)) & \textbf{if } w \notin V \end{cases} \\ &= p(w|h) \cdot \delta(w \in V) + p(U|h) \cdot p(\hat{c}(w)) \end{aligned}$$

normalization: model is correctly normalized,
but changes closed-lexicon LM slightly!

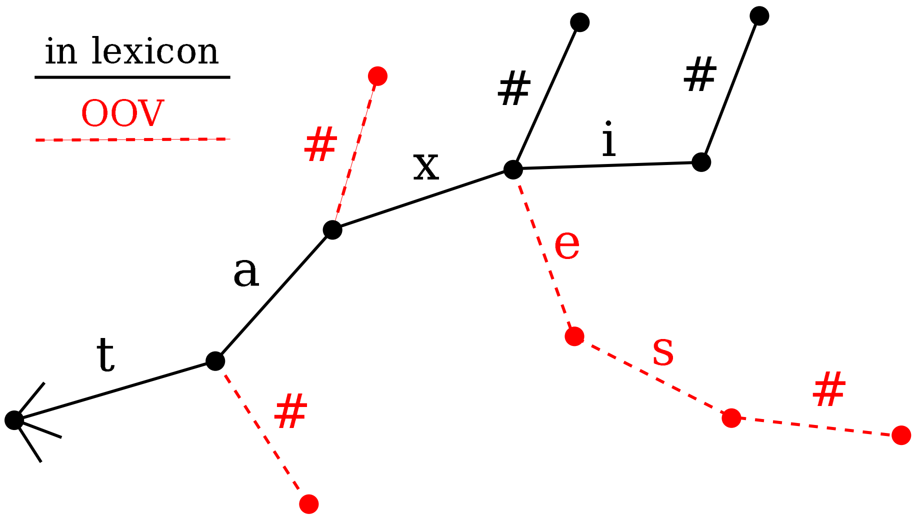
- **combination by maximum:**

$$q(w|h) = \max\{p(w|h) \cdot \delta(w \in V),\ p(U|h) \cdot p(\hat{c}(w))\}$$

normalization: model is deficient!

Combination Using Lexical Prefix Tree

ideal goals:
– preserve the closed-lexicon LM probabilities EXACTLY
– do not waste probability mass
methods so far: none of them satisfies both constraints



method that satisfies BOTH constraints:
– represent closed lexicon and open lexicon JOINTLY in a tree
– when leaving the in-lexicon tree, compute the remaining probability mass and assign it to OOV character sequence
– two variants: without and with early subtraction

Combination by Interpolation

- **starting points:**
  - **closed-lexicon LM $p(w|h)$ WITHOUT unknown symbols !**
  - **character-based LM with word probabilities $p(\hat{c}(w))$**
- **linear interpolation:**

$$q(w|h) = \lambda \cdot p(w|h) + (1 - \lambda) \cdot p(\hat{c}(w))$$

  $\lambda \in [0, 1]$: free parameter (optimized on dev data)

- **properties:**
  - **correct normalization**
  - **closed-lexicon LM probabilities are not preserved!**
- **extension:**
  go across word boundaries in the character-based LM

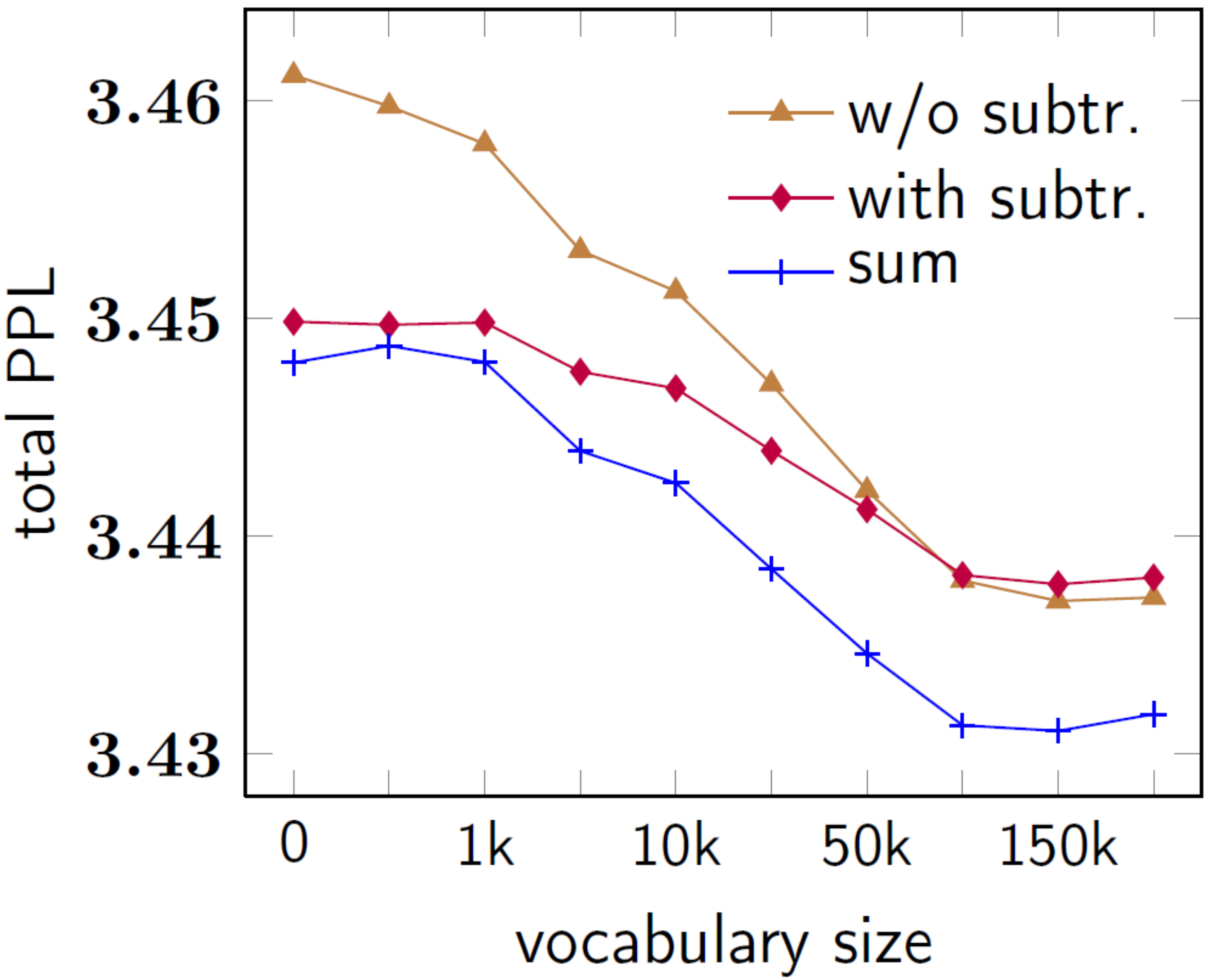
Results: Arabic

corpus:
– 20 Mio running words: GALE and
newspapers (Addustour, Alahram, Albayan, Alittihad, Alwatan, Alraya)
– OOV on test data: 1.0 % for a lexicon of ca. 200k words

type of language model	char PP			word PP
	in-lex	OOV	total	total
word-level only	3.378	–	–	–
char-level only	3.680	19.302	3.722	1438.9
combination by				
– back-off	3.394	18.860	3.438	927.5
– maximum	3.394	18.860	3.437	926.7
– sum	3.387	18.860	3.431	917.6
– prefix tree				
no early subtraction	3.394	18.569	3.437	926.4
with early subtraction	3.394	18.880	3.438	927.6
interpolation				
– not across word boundary	3.393	19.488	3.438	928.1
– across word boundary	3.349	23.846	3.404	878.1

Results on Arabic: Effect of Vocabulary Size

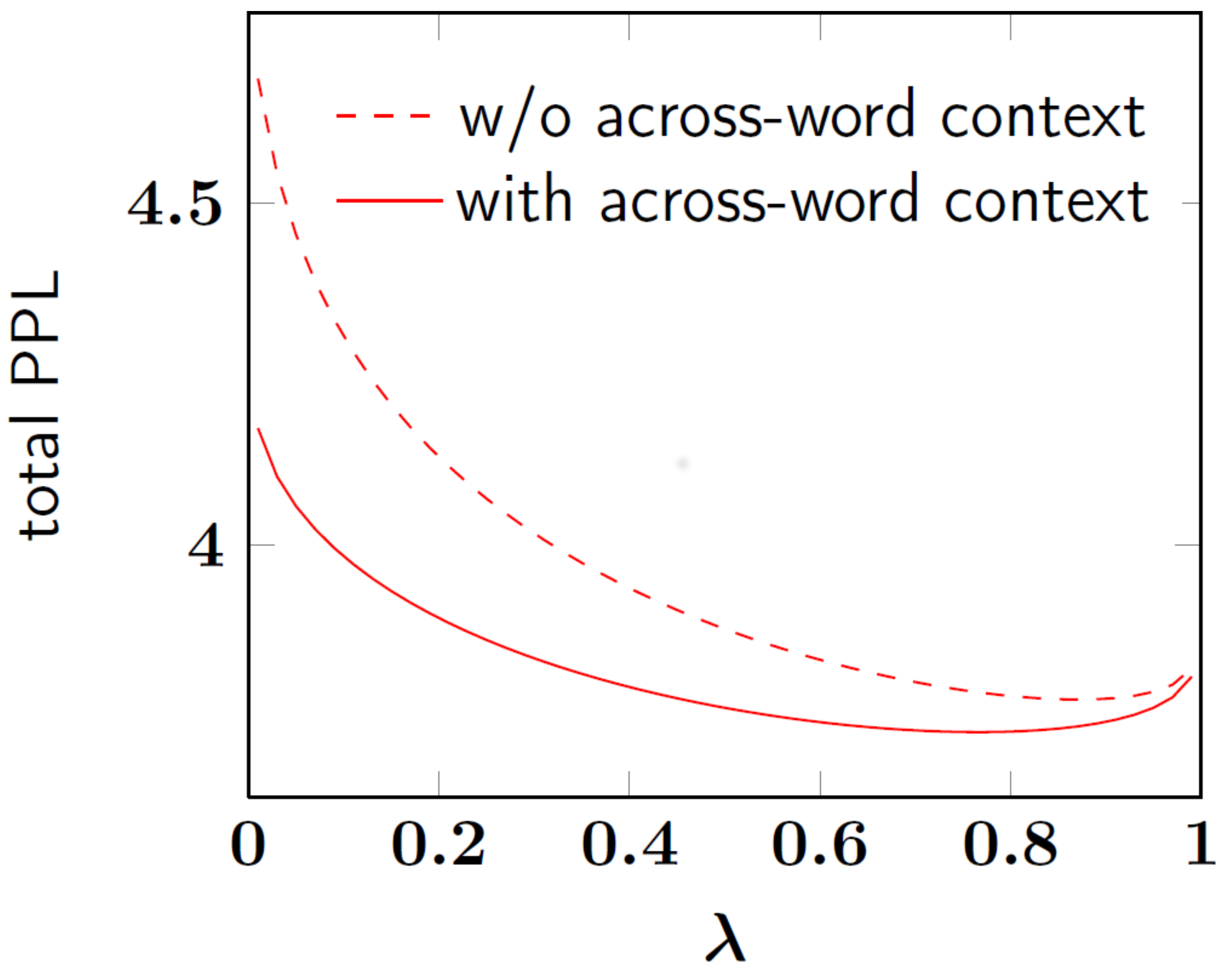
– closed lexicon: vary the vocabulary size explicitly
– measure the effect on perplexity



Results on English: Interpolation

improvements:

linear interpolation and across-word context in character-based LM



Conclusions

- **main result: yes,**
  we can build an open-lexicon language model
  and preserve the closed-lexicon LM probabilities!

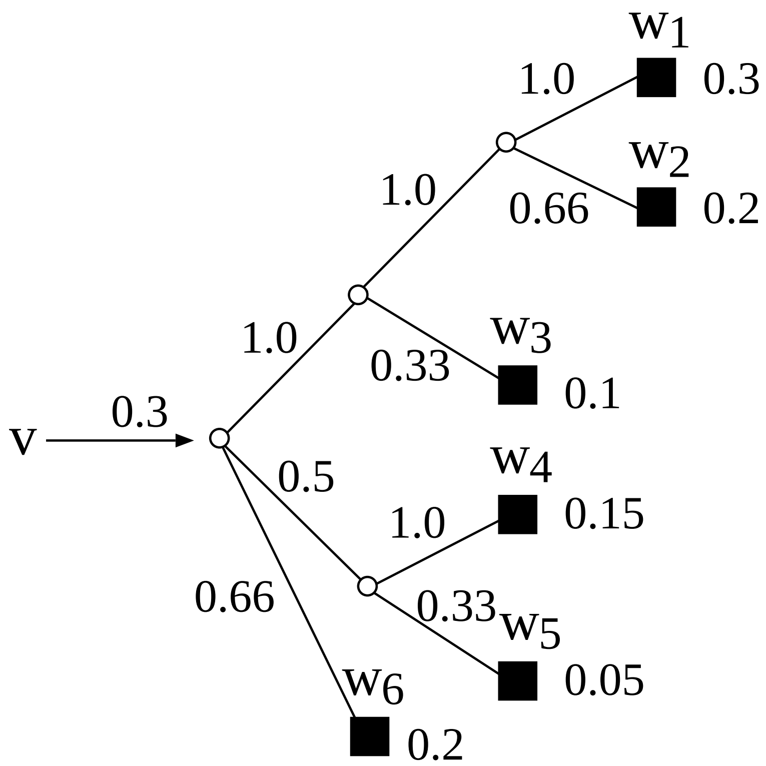
- **various methods:**
  - **exact preservation**
  - **approximate preservation: (small) improvements over closed-lexicon LM**

- **ongoing work:**
  - **experiments on more challenging tasks, e.g. OOV larger than 1%**
  - **detailed analysis of the experimental results,**
    e. g. character-based LM across word boundaries ?
  - **recognition experiments**

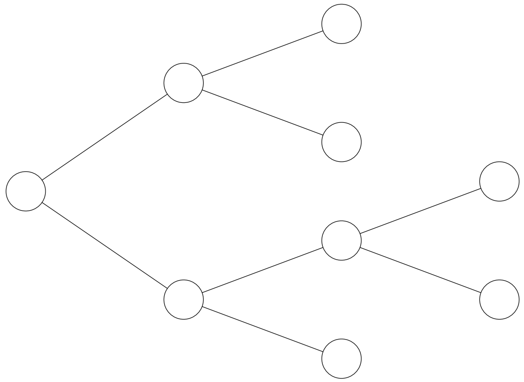
- **future: approach based on first principles:**
  - **start from characters only**
  - **learn larger units (e.g. words, syllables, ...) automatically**

END

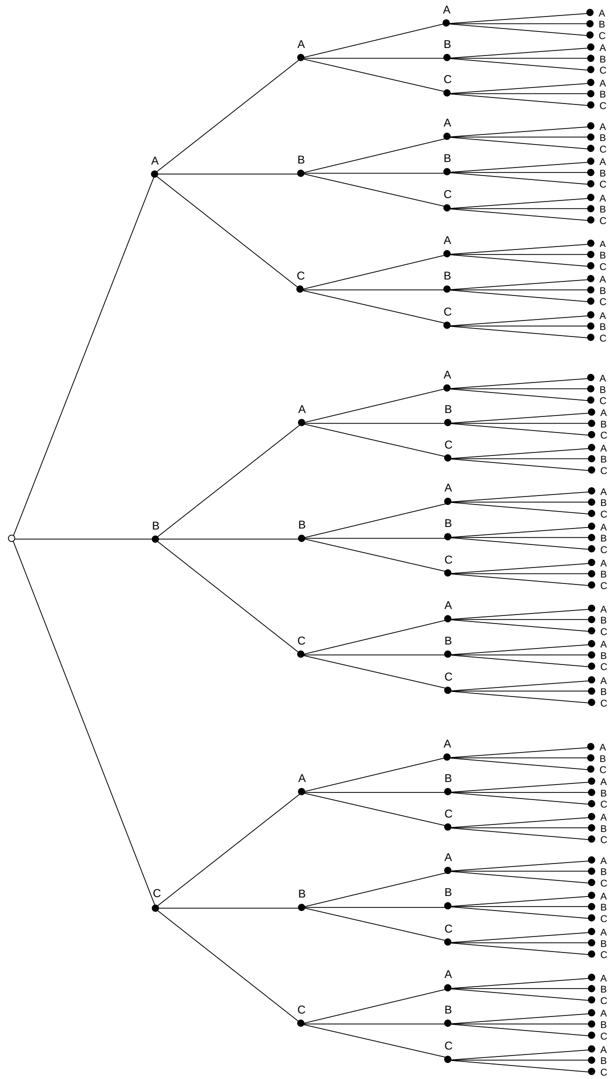
Closed Lexicon: Lexical Prefix Tree



Closed Lexicon: Lexical Prefix Tree

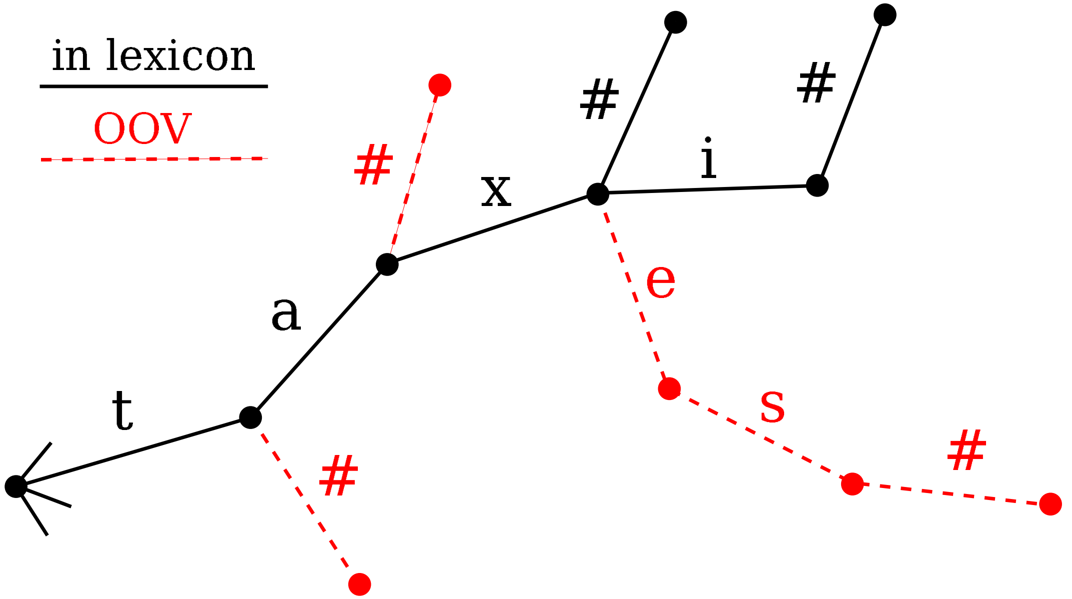


Open Lexicon: Lexical Prefix Tree



Lexical prefix tree model

Another way of achieving the normalization constraint is to represent the character-level model as an infinite lexical prefix tree and then exclude the in-lexicon words (paths).



Solid, black nodes and arcs demonstrate common prefixes for both in-lexicon and OOV words. Dashed, red nodes and arcs illustrate OOV words, outside of the common part of the tree. Once we traverse a red arc, it is impossible to arrive at a black arc again.

Lexical prefix tree model

To exclude the in-lexicon words from this tree we have to drop every in-lexicon word-boundary arc and renormalize.

In the in-lexicon part of the lexical prefix tree the probability depends on the whole word history:

$$\bar{p}(c_1^M) = \prod_{j=1}^{M} \bar{p}(c_j | c_1^{j-1}) \tag{1}$$

As soon as we go into the OOV part, the probability again depends only on the n-gram.

$$\forall c_1^{j-1} : \hat{w}(c_1^{j-1}\#) \notin V \quad \bar{p}(c_j | c_1^{j-1}) = p(c_j | c_{j-m+1}^{j-1}) \tag{2}$$