Speech Recognition
Lecture 5: N-gram Language Models

Mehryar Mohri
Courant Institute and Google Research
mohri@cims.nyu.com

Language Models

- Definition: probability distribution $\Pr[w]$ over sequences of words $w = w_1 \ldots w_k$.
  - Critical component of a speech recognition system.
- Problems:
  - Learning: use large text corpus (e.g., several million words) to estimate $\Pr[w]$. Models in this course: *n*-gram models, maximum entropy models.
  - Efficiency: computational representation and use.

This Lecture

- *n*-gram models definition and problems
- Good-Turing estimate
- Smoothing techniques
- Evaluation
- Representation of *n*-gram models
- Shrinking
- LMs based on probabilistic automata

N-Gram Models

- Definition: an *n*-gram model is a probability distribution based on the *n*th order Markov assumption

$$\forall i, \Pr[w_i \mid w_1 \ldots w_{i-1}] = \Pr[w_i \mid h_i], \ |h_i| \leq n - 1.$$

  - Most widely used language models.

- Consequence: by the chain rule,

$$\Pr[w] = \prod_{i=1}^{k} \Pr[w_i \mid w_1 \ldots w_{i-1}] = \prod_{i=1}^{k} \Pr[w_i \mid h_i].$$

Maximum Likelihood

- Likelihood: probability of observing sample under distribution $p \in \mathcal{P}$, which, given the independence assumption is

$$\Pr[x_1, \ldots, x_m] = \prod_{i=1}^{m} p(x_i).$$

- Principle: select distribution maximizing sample probability

$$p_\star = \operatorname*{argmax}_{p \in \mathcal{P}} \prod_{i=1}^{m} p(x_i),$$

or $$p_\star = \operatorname*{argmax}_{p \in \mathcal{P}} \sum_{i=1}^{m} \log p(x_i).$$

Example: Bernoulli Trials

- **Problem**: find most likely Bernoulli distribution, given sequence of coin flips

$$H, T, T, H, T, H, T, H, H, H, T, T, \ldots, H.$$

- **Bernoulli distribution**: $p(H) = \theta, p(T) = 1 - \theta.$

- **Likelihood**: $$\begin{aligned} l(p) &= \log \theta^{N(H)} (1-\theta)^{N(T)} \\ &= N(H) \log \theta + N(T) \log(1-\theta). \end{aligned}$$

- **Solution**: l is differentiable and concave;

$$\frac{dl(p)}{d\theta} = \frac{N(H)}{\theta} - \frac{N(T)}{1-\theta} = 0 \Leftrightarrow \theta = \frac{N(H)}{N(H)+N(T)}.$$

Maximum Likelihood Estimation

- Definitions:
  - *n*-gram: sequence of *n* consecutive words.
  - S: sample or corpus of size m.
  - $c(w_1 \ldots w_k)$: count of sequence $w_1 \ldots w_k$.
- ML estimates: for $c(w_1 \ldots w_{n-1}) \neq 0$,

$$\Pr[w_n | w_1 \ldots w_{n-1}] = \frac{c(w_1 \ldots w_n)}{c(w_1 \ldots w_{n-1})}.$$

  - But, $c(w_1 \ldots w_n) = 0 \implies \Pr[w_n | w_1 \ldots w_{n-1}] = 0!$

N-Gram Model Problems

- Sparsity: assigning probability zero to sequences not found in the sample $\Longrightarrow$ speech recognition errors.
  - Smoothing: adjusting ML estimates to reserve probability mass for unseen events. Central techniques in language modeling.
  - Class-based models: create models based on classes (e.g., DAY) or phrases.
- Representation: for $|\Sigma| = 100{,}000$, the number of bigrams is 10^{10}, the number of trigrams 10^{15}!
  - Weighted automata: exploiting sparsity.

Smoothing Techniques

- Typical form: interpolation of n-gram models, e.g., trigram, bigram, unigram frequencies.

$$\Pr[w_3|w_1w_2] = \alpha_1 c(w_3|w_1w_2) + \alpha_2 c(w_3|w_2) + \alpha_3 c(w_3).$$

- Some widely used techniques:
  - Katz Back-off models (Katz, 1987).
  - Interpolated models (Jelinek and Mercer, 1980).
  - Kneser-Ney models (Kneser and Ney, 1995).

Good-Turing Estimate

(Good, 1953)

- Definitions:
  - Sample S of m words drawn from vocabulary Σ.
  - $c(x)$: count of word x in S.
  - S_k : set of words appearing k times.
  - M_k : probability of drawing a point in S_k.
- Good-Turing estimate of M_k :

$$G_k = \frac{k+1}{m}|S_{k+1}| = \frac{(k+1)|S_{k+1}|}{m}.$$

Properties

- **Theorem**: the Good-Turing estimate is an estimate of M_k with small bias, for small values of k/m:

$$\mathop{\mathrm{E}}_{S}[G_k] = \mathop{\mathrm{E}}_{S}[M_k] + O\left(\frac{k+1}{m}\right).$$

- **Proof**:

$$\begin{aligned}
\mathop{\mathrm{E}}_{S}[M_k] &= \sum_{x \in \Sigma} \Pr[x] \Pr[x \in S_k] \\
&= \sum_{x \in \Sigma} \Pr[x] \binom{m}{k} \Pr[x]^k (1 - \Pr[x])^{m-k} \\
&= \boxed{\sum_{x \in \Sigma} \binom{m}{k+1} \Pr[x]^{k+1} (1 - \Pr[x])^{m-(k+1)}} \frac{\binom{m}{k}}{\binom{m}{k+1}} (1 - \Pr[x]) \\
&= \frac{k+1}{m-k} \Big[\boxed{\mathrm{E}\big[|S_{k+1}|\big]} - \mathrm{E}\big[M_{k+1}\big] \Big] = \frac{m}{m-k} \mathrm{E}[G_k] - \frac{k+1}{m-k} \mathrm{E}[M_{k+1}].
\end{aligned}$$

Properties

- **Proof** (cont.): thus,

$$\left|\operatorname*{E}_S[M_k] - \operatorname*{E}_S[G_k]\right| = \left|\frac{k}{m-k}\operatorname*{E}_S[G_k] - \frac{k+1}{m-k}\operatorname{E}[M_{k+1}]\right|$$
$$\leq \frac{k+1}{m-k}. \qquad (0 \leq \operatorname*{E}_S[G_k], \operatorname{E}[M_{k+1}] \leq 1)$$

- In particular, for $k=0$,

$$\left|\operatorname*{E}_S[M_k] - \operatorname*{E}_S[G_k]\right| \leq \frac{1}{m}.$$

- It can be proved using McDiarmid's inequality that with probability at least $1-\delta$, (McAllester and Schapire, 2000),

$$M_0 \leq G_0 + O\left(\sqrt{\frac{\log(\frac{1}{\delta})}{m}}\right).$$

Good-Turing Count Estimate

- Definition: let $r = c(w_1 \ldots w_k)$ and $n_r = |S_r|$, then

$$c^*(w_1 \ldots w_k) = \frac{G_r \times m}{|S_r|} = (r+1)\frac{n_{r+1}}{n_r}.$$

Simple Method

(Jeffreys, 1948)

- Additive smoothing: add $\delta \in [0, 1]$ to the count of each *n*-gram.

$$\Pr[w_n | w_1 \ldots w_{n-1}] = \frac{c(w_1 \ldots w_n) + \delta}{c(w_1 \ldots w_{n-1}) + \delta|\Sigma|}.$$

  - Poor performance (Gale and Church, 1994).
  - Not a principled attempt to estimate or make use of an estimate of the missing mass.

Katz Back-off Model

(Katz, 1987)

- **Idea**: back-off to lower order model for zero counts.
  - if $c(w_1^{n-1}) = 0$, then $\Pr[w_n | w_1^{n-1}] = \Pr[w_n | w_2^{n-1}]$.
  - otherwise,

$$\Pr[w_n | w_1^{n-1}] = \begin{cases} d_{c(w_1^n)} \dfrac{c(w_1^n)}{c(w_1^{n-1})} & \text{if } c(w_1^n) > 0 \\ \beta \Pr[w_n | w_2^{n-1}] & \text{otherwise.} \end{cases}$$

where d_k is a discount coefficient such that

$$d_k = \begin{cases} 1 & \text{if } k > K; \\ \approx \dfrac{(k+1) n_{k+1}}{k n_k} & \text{otherwise.} \end{cases}$$

Katz suggests $K = 5$.

Discount Coefficient

- With the Good-Turing estimate, the total probability mass for unseen *n*-grams is $G_0 = n_1/m$.

$$\sum_{c(w_1^n)>0} d_{c(w_1^n)} c(w_1^n)/m = 1 - n_1/m$$

$$\Leftrightarrow \sum_{k>0} d_k\, k\, n_k = m - n_1$$

$$\Leftrightarrow \sum_{k>0} d_k\, k\, n_k = \sum_{k>0} k\, n_k - n_1$$

$$\Leftrightarrow \sum_{k>0} (1 - d_k)\, k\, n_k = n_1.$$

$$\Leftrightarrow \sum_{k=1}^{K} (1 - d_k)\, k\, n_k = n_1.$$

Discount Coefficient

- **Solution**: a search with $1 - d_k = \mu(1 - \frac{k^*}{k})$ leads to

$$
\begin{aligned}
& \mu \sum_{k=1}^{K} (1 - k^*/k)\, k\, n_k = n_1. \\
\Leftrightarrow\ & \mu \sum_{k=1}^{K} (1 - \frac{(k+1)n_{k+1}}{k n_k})\, k n_k = n_1. \\
\Leftrightarrow\ & \mu \sum_{k=1}^{K} [k n_k - (k+1) n_{k+1}] = n_1. \\
\Leftrightarrow\ & \mu [n_1 - (K+1) n_{K+1}] = n_1. \\
\Leftrightarrow\ & \mu = \frac{1}{1 - \frac{(K+1) n_{K+1}}{n_1}} \\
\Leftrightarrow\ & d_k = \frac{\frac{k^*}{k} - \frac{(K+1) n_{K+1}}{n_1}}{1 - \frac{(K+1) n_{K+1}}{n_1}}.
\end{aligned}
$$

Interpolated Models

(Jelinek and Mercer, 1980)

- Idea: interpolation of different order models.

$$\Pr[w_3|w_1w_2] = \alpha c(w_3|w_1w_2) + \beta c(w_3|w_2) + (1-\alpha-\beta)c(w_3),$$

with $0 \leq \alpha, \beta \leq 1$.

  - α and β are estimated by using held-out samples.
  - sample split into two parts for training higher-order and lower order models.
  - optimization using expectation-maximization (EM) algorithm.
  - deleted interpolation: *k*-fold cross-validation.

Kneser-Ney Model

- Idea: combination of back-off and interpolation, but backing-off to lower order model based on counts of contexts. Extension of absolute discounting.

$$\Pr[w_3|w_1w_2] = \frac{\max\{0, c(w_1w_2w_3) - D\}}{c(w_1w_2)} + \alpha\frac{c(\cdot w_3)}{\sum c(\cdot w_3)},$$

  where D is a constant.

- Modified version (Chen and Goodman, 1998): D function of $c(w_1w_2w_3)$.

Evaluation

- Average log-likelihood of test sample of size *N*:

$$\widehat{L}(p) = \frac{1}{N} \sum_{k=1}^{N} \log_2 p[w_k \mid h_k], \quad |h_k| \leq n-1.$$

- Perplexity: the perplexity $PP(q)$ of the model is defined as ('average branching factor')

$$PP(q) = 2^{-\hat{L}(q)}.$$

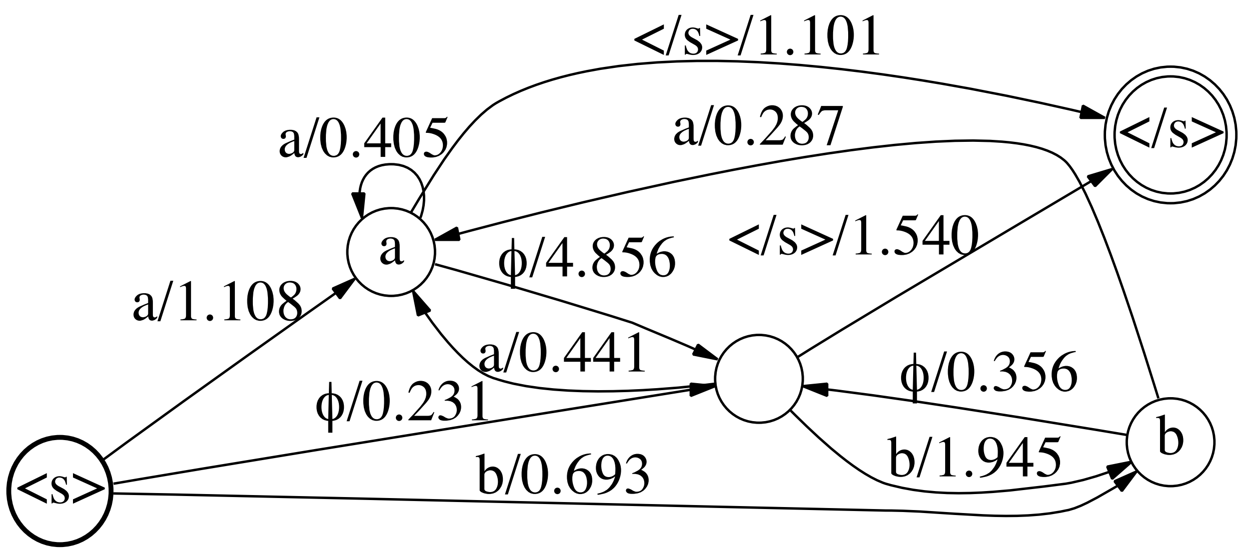
  - For English texts, typically $PP(q) \in [50, 1000]$ and

$$6 \leq \widehat{L}(q) \leq 10 \text{ bits}.$$

In Practice

- Evaluation: an empirical observation based on the word error rate of a speech recognizer is often a better evaluation than perplexity.
- *n*-gram order: typically $n = 3, 4$, or 5. Higher order *n*-grams typically do not yield any benefit.
- Smoothing: small differences between Back-off, interpolated, and other models (Chen and Goodman, 1998).
- Special symbols: beginning and end of sentences, start and stop symbols.

Example: Bigram Model



$\langle$s$\rangle$ b a a a a $\langle$/s$\rangle$

$\langle$s$\rangle$ b a a a a $\langle$/s$\rangle$

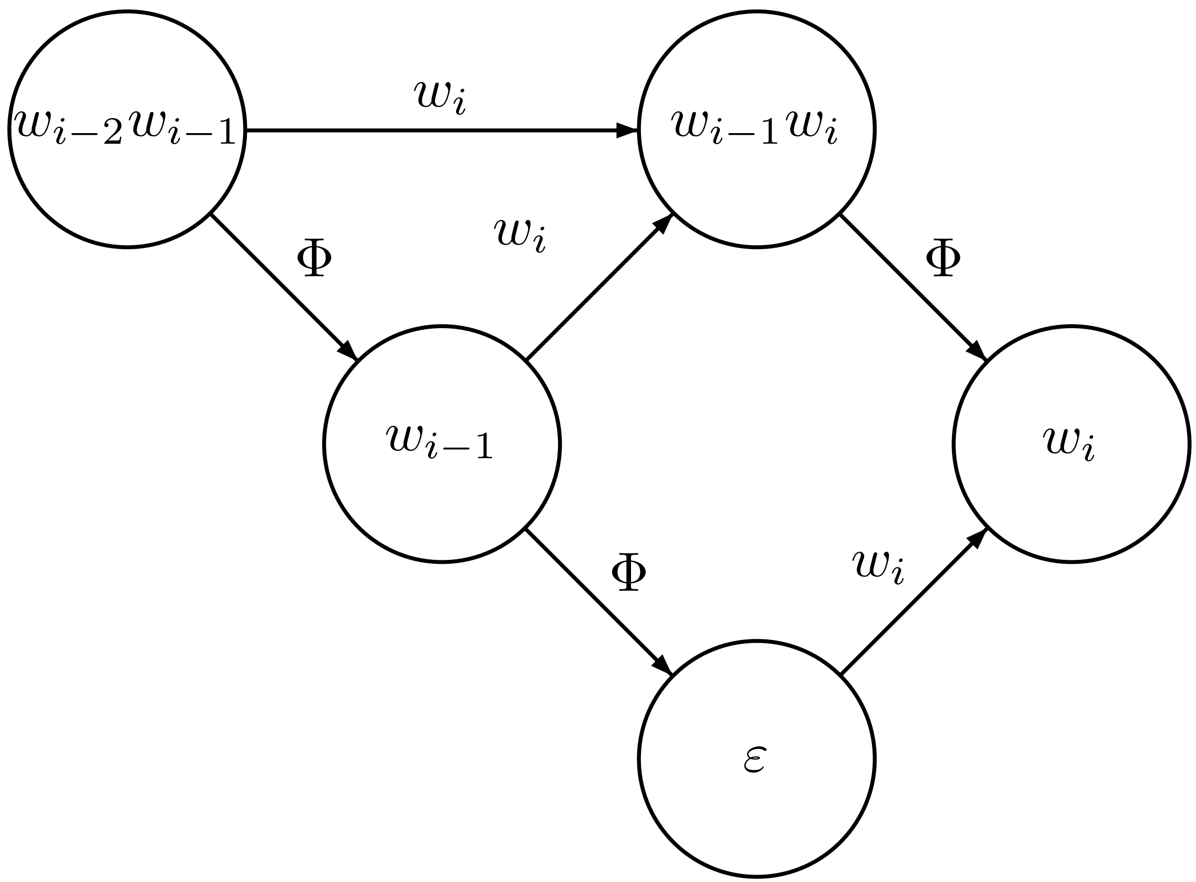
$\langle$s$\rangle$ a $\langle$/s$\rangle$

Failure Transitions

(Aho and Corasick 1975; MM, 1997)

- Definition: a failure transition Φ at state q of an automaton is a transition taken at that state without consuming any symbol, when no regular transition at q has the desired input label.
  - Thus, a failure transition corresponds to the semantics of otherwise.
- Advantages: originally used in string-matching.
  - More compact representation.
  - Dealing with unknown alphabet symbols.

Weighted Automata Representation



Representation of a trigram model using failure transitions (de Bruijn graphs).

Approximate Representation

- The cost of a representation without failure transitions and ϵ-transitions is prohibitive.
  - For a trigram model, $|\Sigma|^{n-1}$ states and $|\Sigma|^n$ transitions are needed.
  - Exact on-the-fly representation but drawback: no offline optimization.
- Approximation: empirically, limited accuracy loss.
  - Φ-transitions replaced by ϵ-transitions.
  - Log semiring replaced by tropical semiring.
- Alternative: exact representation with ϵ-transitions (Allauzen, Roark, and MM, 2003).

Shrinking

- Idea: remove some *n*-grams from the model while minimally affecting its quality.
- Main motivation: real-time speech recognition (speed and memory).
- Method of (Seymore and Rosenfeld, 1996): rank *n*-grams (w, h) according to difference of log probabilities before and after shrinking:

$$c^*(wh)\Big[\log p[w|h] - \log p'[w|h]\Big].$$

Shrinking

- Method of (Stolcke, 1998): greedy removal of *n*-grams based on relative entropy $D(p\|p')$ of the models before and after removal, independently for each *n*-gram.
  - slightly lower perplexity.
  - but, ranking close to that of (Seymore and Rosenfeld, 1996) both in the definition and empirical results.

LMs Based on Probabilistic Automata

(Allauzen, MM, and Roark, 1997)

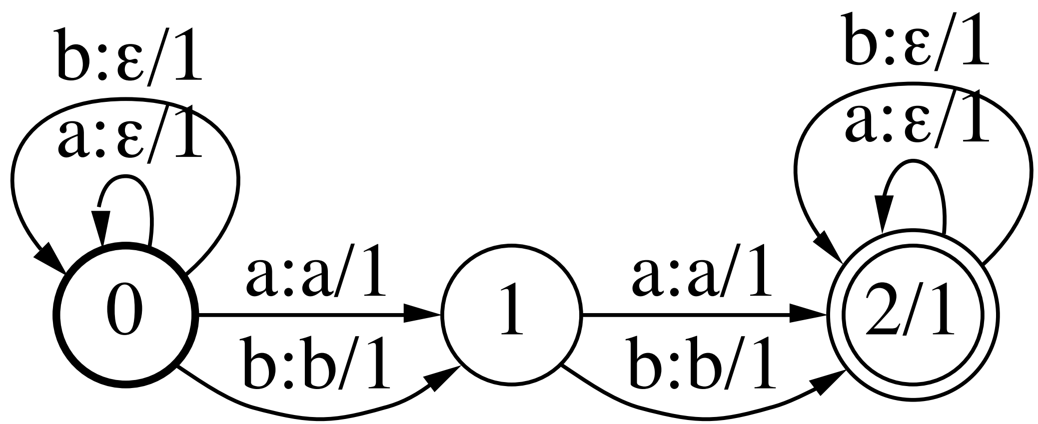
- Definition: expected count of sequence x in probabilistic automaton:

$$c(x) = \sum_{u \in \Sigma^*} |u|_x A(u),$$

where $|u|_x$ is the number of occurrences of x in u.

- Computation:
  - use counting weighted transducers.
  - can be generalized to other moments of the counts.

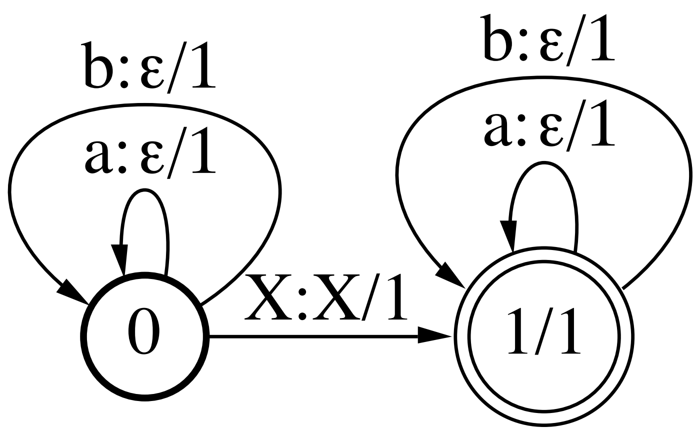
Example: Bigram Transducer

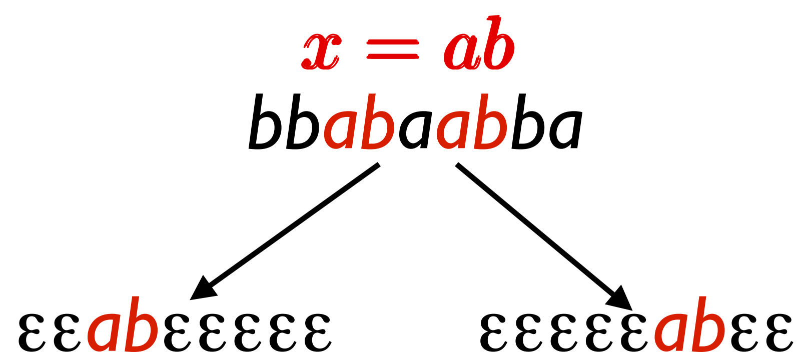


Weighted transducer T.

$X \circ T$ computes the (expected) count of each bigram $\{aa, ab, ba, bb\}$ in X.

Counting Transducers





- X is an automaton representing a string or any other regular expression.
- Alphabet $\Sigma = \{a, b\}$.

References

- Cyril Allauzen, Mehryar Mohri, and Brian Roark. Generalized Algorithms for Constructing Statistical Language Models. In 41st *Meeting of the Association for Computational Linguistics (ACL 2003)*, Proceedings of the Conference, Sapporo, Japan. July 2003.

- Peter F. Brown, Vincent J. Della Pietra, Peter V. deSouza, Jennifer C. Lai, and Robert L. Mercer. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18 (4):467-479.

- Stanley Chen and Joshua Goodman. An empirical study of smoothing techniques for language modeling. Technical Report, TR-10-98, Harvard University. 1998.

- William Gale and Kenneth W. Church. What's wrong with adding one? In N. Oostdijk and P. de Hann, editors, *Corpus-Based Research into Language*. Rodolpi, Amsterdam.

- Good, I. The population frequencies of species and the estimation of population parameters, *Biometrika*, 40, 237-264, 1953.

- Frederick Jelinek and Robert L. Mercer. 1980. Interpolated estimation of Markov source parameters from sparse data. In *Proceedings of the Workshop on Pattern Recognition in Practice*, pages 381-397.

References

- Slava Katz . Estimation of probabilities from sparse data for the language model component of a speech recognizer, *IEEE Transactions on Acoustics, Speech and Signal Processing*, 35, 400-401, 1987.

- Reinhard Kneser and Hermann Ney. Improved backing-off for m-gram language modeling. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 1, pages 181-184, 1995.

- David A. McAllester, Robert E. Schapire: On the Convergence Rate of Good-Turing Estimators. *Proceedings of Conference on Learning Theory (COLT)* 2000: 1-6.

- Hermann Ney, Ute Essen, and Reinhard Kneser. 1994. On structuring probabilistic dependences in stochastic language modeling. *Computer Speech and Language*, 8:1-38.

- Kristie Seymore and Ronald Rosenfeld. Scalable backoff language models. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, 1996.

- Andreas Stolcke. 1998. Entropy-based pruning of back-off language models. In *Proceedings of DARPA Broadcast News Transcription and Understanding Workshop*, pages 270-274.

References

- Ian H. Witten and Timothy C. Bell. The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression, *IEEE Transactions on Information Theory*, 37(4):1085-1094, 1991.