

Word Topical Mixture Models for Language Model Adaptation



Berlin Chen
Department of Computer Science & Information Engineering
National Taiwan Normal University



Outline

- Introduction
- Conventional Language Model Adaptation Methods
- Proposed Word Topical Mixture Model (WTMM)
- Comparison between word TMM , PLSALM and TBLM
- Experimental Results
- Conclusions and Future Work

Introduction

- n -gram modeling is not always adequate
 - Only capture local contextual information or word regularities
- Probabilistic Latent Semantic Analysis (PLSA)–based LM can be used to complement n -gram models
 - Model the co-occurrence relationship between a word and its history through a set of latent topical distributions
- Trigger-based LM can also be used
 - The long-distance relationship between the words in the search history and the currently predicted word can be captured

Probabilistic Latent Semantic Analysis (1/2)

- PLSA models the co-occurrence of word and documents and evaluates the relevance in a low dimensional semantic/topic space

- Each document is treated as a document model

$$P(w_i | M_D) = \sum_{k=1}^K P(w_i | T_k) P(T_k | M_D)$$

- Model parameters are trained beforehand using a set of text documents

- Maximize the log-likelihood of entire collection

$$\log L_D = \sum_{D \in \mathbf{D}} \log P(D | M_D) = \sum_{D \in \mathbf{D}} \sum_{w_n \in D} n(w_n, D) \log P(w_n | M_D)$$

Probabilistic Latent Semantic Analysis (2/2)

- PLSA in LM Adaptation
 - The search history can be treated as a pseudo-document which is varying during the speech recognition process

$$P(w_i | H_{w_i}) = \sum_{k=1}^K P(w_i | T_k) P(T_k | H_{w_i})$$

- The topic unigrams $P(w_i | T_k)$ are kept unchanged
- The history's probability distribution over the latent topics is gradually updated
- The topic mixture weights are estimated on the fly

Trigger-based LM (1/2)

- To capture long-distance information, we also can use trigger pairs
- Instead of using the average mutual information (MI) for the selection of trigger pairs, the TF/IDF measure which captures both local and global information can be used

$$Score_{MI}(w_j, w_i) = \log \frac{P(w_j, w_i)}{P(w_j)P(w_i)}$$

$$Score_{TF/IDF}(w_j, d_k) = \frac{tf(w_i, d_k) \log(N/df_i)}{\sqrt{\sum_{j=1}^{|d_k|} tf(w_i, d_j)^2 [\log(N/df_i)]^2}}$$

- Word pairs with MI or TF/IDF scores above a threshold are selected

Trigger-based LM (2/2)

- The associated conditional probability of the selected trigger pair can be estimated by using a context window

$$P_{Trig}(w_i | w_j) = \frac{n(w_j, w_i)}{\sum_{w_l} n(w_j, w_l)}$$

- The search history for a decoded word can be viewed as a series of words and the probability of the search history predicting word can be expressed by linearly combining the conditional probabilities of the trigger pairs

$$P_{Trig}(w_i | H_{w_i}) = \frac{1}{|H_{w_i}|} \sum_{w_j \in H_{w_i}} P_{Trig}(w_i | w_j)$$

Word Topical Mixture Model (1/4)

- In this research, each word of language are treated as a word topical mixture model (WTMM) for predicting the occurrences of other words

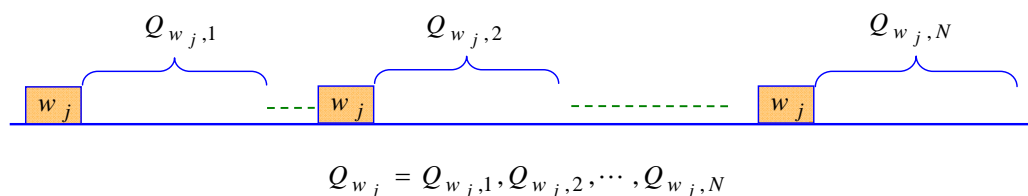
$$P(w_i | M_{w_j}) = \sum_{k=1}^K P(w_i | T_k) P(T_k | M_{w_j})$$

- WTMM in LM Adaptation
 - Each history consists of words
 - History model is treated as a composite word TMM
 - The history model of a decoded word can be dynamically constructed

$$\begin{aligned} P(w_i | H_{w_j}) &= \sum_{j=1}^{i-1} \alpha_j P(w_i | M_{w_j}) = \sum_{j=1}^{i-1} \alpha_j \sum_{k=1}^K P(w_i | T_k) P(T_k | M_{w_j}) \\ &= \sum_{k=1}^K P(w_i | T_k) \sum_{j=1}^{i-1} \alpha_j P(T_k | M_{w_j}) = \sum_{k=1}^K P(w_i | T_k) P'(T_k | M_{H_{w_j}}) \end{aligned}$$

Word Topical Mixture Model (2/4)

- Exploration of Training Exemplars
 - Collect the words within a context window around each occurrence of word in the training corpus
 - Concatenate them to form the relevant observations for training the word TMM



- Maximize the sum of log-likelihoods of WTMM models generating their corresponding training exemplars

$$\log L_{\mathbf{Q}_{TrainSet}} = \sum_{Q_{w_j} \in \mathbf{Q}_{TrainSet}} \log P(Q_{w_j} | M_{w_j}) = \sum_{Q_{w_j} \in \mathbf{Q}_{TrainSet}} \sum_{w_n \in Q_{w_j}} n(w_n, Q_{w_j}) \log P(w_n | M_{w_j})$$

Word Topical Mixture Model (3/4)

- Training of WTMM models
 - Expectation-Maximization (EM) Training formulas

$$\hat{P}(w_n | T_k) = \frac{\sum_{w_j \in \mathbf{W}} n(w_n, Q_{w_j}) P(T_k | w_n, M_{w_j})}{\sum_{w_l \in \mathbf{W}} \sum_{w_n \in Q_{w_l}} n(w_n, Q_{w_s}) P(T_k | w_n, M_{w_l})} \quad \hat{P}(T_k | M_{w_j}) = \frac{\sum_{w_s \in Q} n(w_s, Q_{w_j}) P(T_k | w_s, M_{w_j})}{\sum_{w_l \in Q_{w_j}} n(w_l, Q_{w_j})}$$

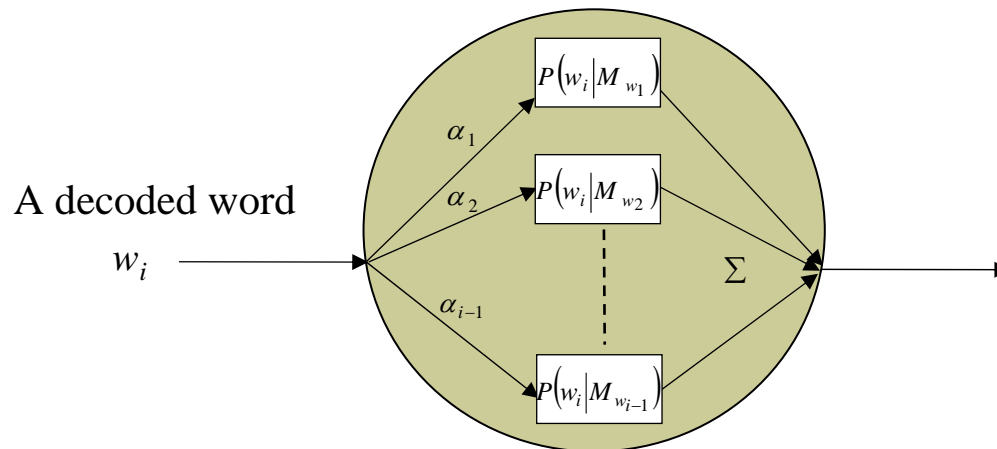
$$\text{where } P(T_k | w_n, M_{w_j}) = \frac{P(T_k | M_{w_j}) P(w_n | T_k)}{\sum_{l=1}^K P(T_l | M_{w_j}) P(w_n | T_l)}$$

- Similar to PLSA but trained in the supervised manner (for its prediction ability)

Word Topical Mixture Model (4/4)

- Recognition using WTMM models
 - A simple linear combination of WTMM models of the words occurring in the search history

A composite word TMM model for the search history $H_{w_i} = w_1, w_2, \dots, w_{i-1}$



- Weights are empirically set to be exponentially decayed as the words in the history are apart from current decoded word

Comparison of WTMM, PLSALM and TBLM

	WTMM	PLSALM	TBLM
Modeling Relationship	Words	Word and History	Words
Model Estimation	Offline	On the fly	Offline
Topic Modeling	Explicit	Explicit	Implicit
Parameters	$V \times K \times 2$	$V \times K + K \times D$	At most $V \times V$
Prediction Ability	Yes	No	Yes

V : Vocabulary size; K : Topic number; D : Number of documents used for training

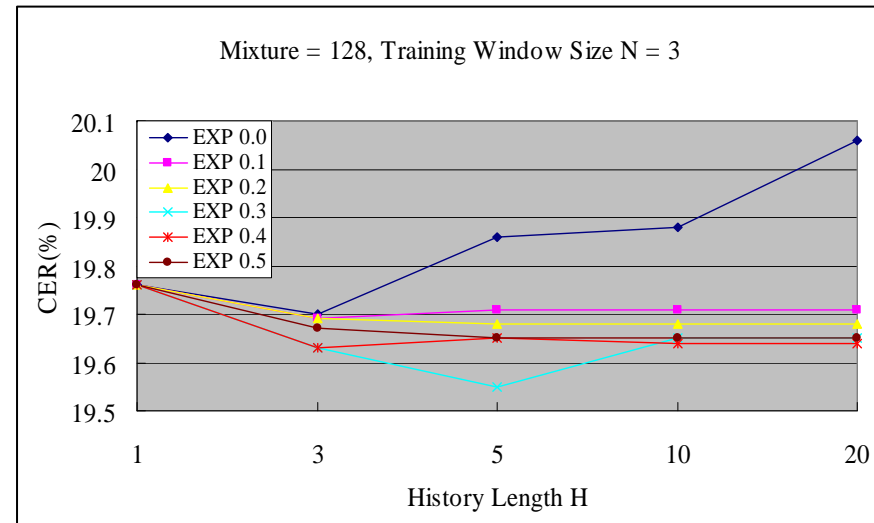
- Topic Modeling: Model topics with explicit or implicit probability distribution
- Prediction Ability: The prediction of the decoded word given the search history

Experimental Setup

- Background LM corpus
 - Central News Agency Text News 2001 ~ 2002
 - 170 million words
- LM Adaptation corpus
 - Mandarin Across Taiwan Broadcast News (MATBN) collected during 2001~2002 and consisting of 1 million words
- Speech Recognition Test Set
 - 2003 MATBM consisting of 1.5 hr speech data
- In this study, the language model adaptation experiments were performed in the lattice rescoring procedure

Experimental Results (1/4)

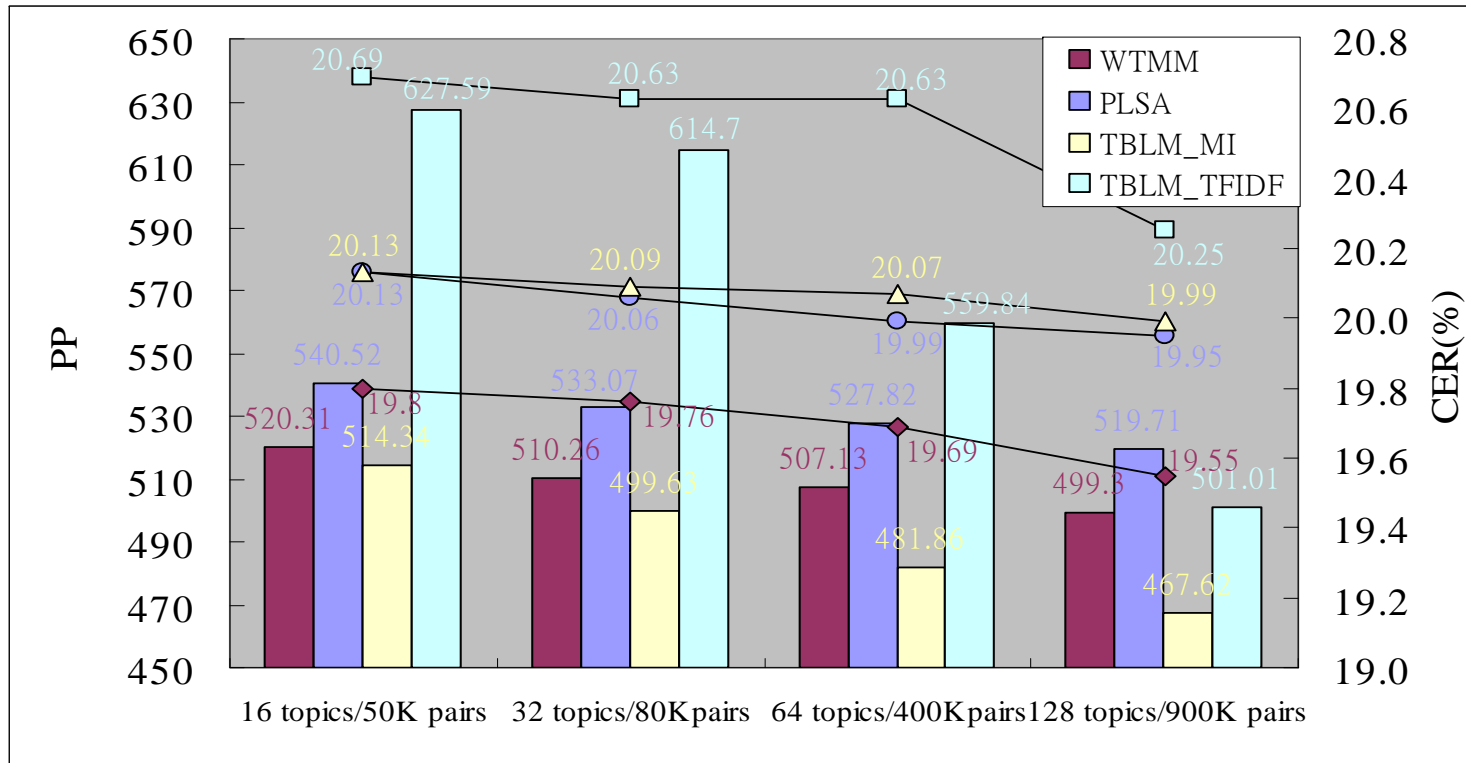
- Experiment-I: WTMM baseline settings



- The best CER (Chinese character error rate) result was achieved when training window size $M = 3$, exponential decay rate = 0.3 and history length $H = 5$ in our task

Experimental Results (2/4)

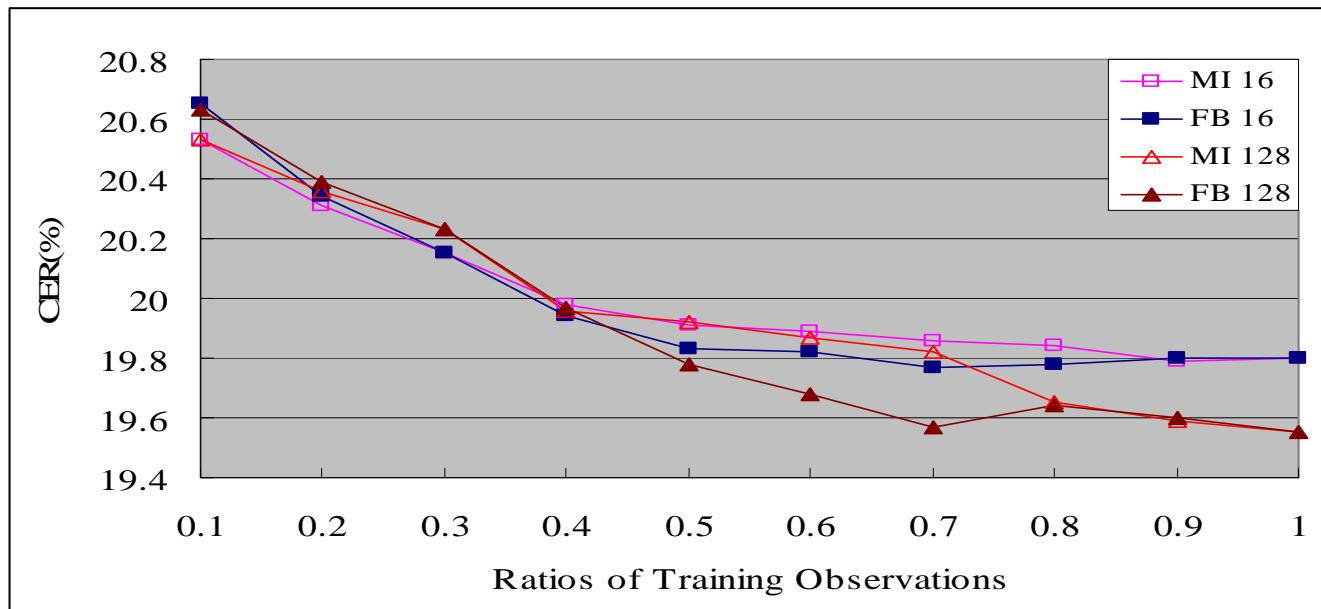
- Experiment-II: Comparison of WTMM, PLSALM, TBLM



- WTMM performs slightly better than PLSALM and TBLM in CER measure
- TBLM trained with MI score performs better in PP (perplexity) measure

Experimental Results (3/4)

- Experiment-III: MI and FB score for WTMM training observation selection



- Training observations can be further reduced (by 30% in our task) using the two statistical measures without loss of performance
- The results obtained using FB score is better than that using MI score

Experimental Results (4/4)

- Experiment-IV: Comparison of WTMM and other LMs

CER(%)	16 Topics	32 Topics	64 Topics	128 Topics
WTMM	19.80	19.76	19.69	19.55
Class-based Bigram LM	20.13	19.92	19.95	19.88
Aggregate Markov Model	19.67	19.67	19.70	19.79
Mixed-Order Markov Model	Order 2	Order 3	Order 4	Order 5
	19.75	19.86	19.74	19.73

PP	16 Topics	32 Topics	64 Topics	128 Topics
WTMM	520.31	510.26	507.13	499.30
Class-based Bigram LM	546.69	526.65	509.37	497.47
Aggregate Markov Model	515.00	504.69	501.97	498.78
Mixed-Order Markov Model	Order 2	Order 3	Order 4	Order 5
	496.28	489.68	487.33	486.09

- WTMM performs as well as the other models
- Aggregate Markov model is a specific case of WTMM (with training window size $M = 1$ and history length $H = 1$)
- Mixed-order Markov model can be considered as a combination of a set of skip- K bigram models

Conclusions

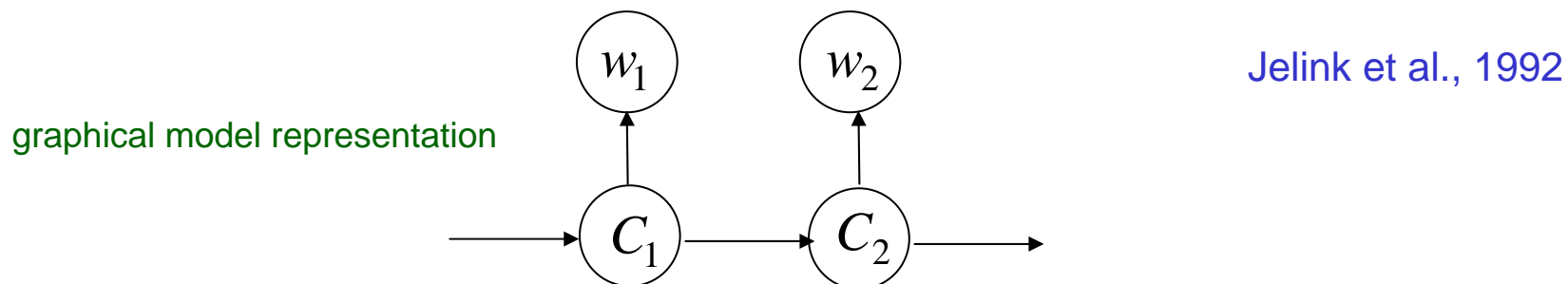
- We have proposed a word topical mixture model (WTMM) for dynamic language model adaptation
- We compared it with the PLSA- and TBLM-based approaches and very promising results in both perplexity and character error rate reductions were initially obtained
- WTMM has also been properly applied to the spoken document summarization task
- More in-deep investigation and analysis of the word TMM-based approaches are currently undertaken

References

- H.S. Chiu, B. Chen. Word topical mixture models for dynamic language model adaptation, *ICASSP2007*
- C. Troncoso, T. Kawahara. Trigger-based language model adaptation for automatic meeting transcription, *Interspeech2005*
- D. Gildea, T. Hofmann. Topic-based language models using EM, *Eurospeech1999*
- L. K. Saul and F. C. N. Pereira. Aggregate and mixed-order Markov models for statistical language processing. *EMNLP1997*

Appendix A: Class-based Bigram Model

- (Hidden Markov models for) Class-based bigram model



- Nondeterministic class assignment

$$P(w_2 | w_1) = \sum_{c_1=1}^K \sum_{c_2=1}^K P(c_1 | w_1) \cdot P(c_2 | c_1) \cdot P(w_2 | c_2)$$

- Deterministic class assignment

$$P(w_2 | w_1) = P(c_2 | c_1) \cdot P(w_2 | c_2)$$

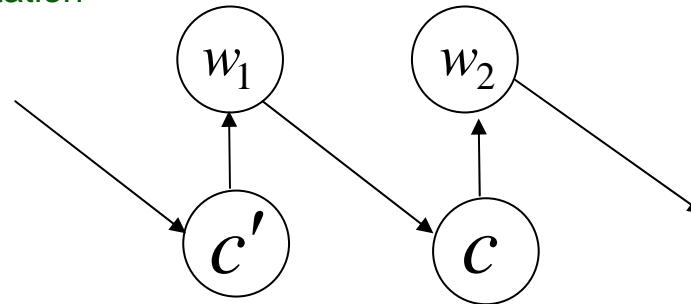
- Estimation of class bigram and word unigram probabilities

Appendix B: Aggregate Markov Model

- An alternative approach for class-based bigram LMs

graphical model representation

Saul & Pereira, 1997



$$P(w_2 | w_1) = \sum_{c=1}^K P(w_2 | c)P(c | w_1)$$

- Models trained by maximizing the log-likelihood of the training corpus

$$l = \sum_{w_1, w_2} n(w_1, w_2) \ln P(w_2 | w_1)$$

Appendix B: Aggregate Markov Model (2/2)

- Model Training Using the EM algorithm
 - Expectation

$$P(c | w_1, w_2) = \frac{P(w_2 | c)P(c | w_1)}{\sum_{c'} P(w_2 | c')P(c' | w_1)}$$

- Maximization

$$P(c | w_1) = \frac{\sum_w N(w_1, w)P(c | w_1, w)}{\sum_{w, c'} N(w_1, w)P(c' | w_1, w)}$$

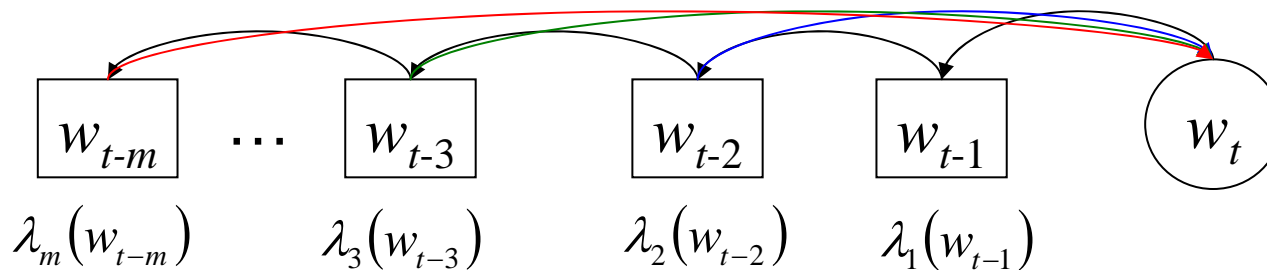
$$P(w_2 | c) = \frac{\sum_w N(w, w_2)P(c | w, w_2)}{\sum_{w, w'} N(w, w')P(c | w, w')}$$

Appendix C: Mixed-order Markov Model (1/2)

- Probability distribution
 - Combine skip- k transition matrix

$$p(w_t | w_{t-m}, \dots, w_{t-1}) = \sum_{k=1}^m \lambda_k(w_{t-k}) M_k(w_{t-k}, w_t) \prod_{j=1}^{k-1} (1 - \lambda_j(w_{t-j}))$$

- Can be viewed as a coin toss process



At position 1, the probability = $\lambda_1(w_{t-1}) M_1(w_{t-1}, w_t)$
 At position 2, the probability = $\lambda_2(w_{t-2}) M_2(w_{t-2}, w_t) (1 - \lambda_1(w_{t-1}))$
 At position 3, the probability = $\lambda_3(w_{t-3}) M_3(w_{t-3}, w_t) (1 - \lambda_1(w_{t-1})) (1 - \lambda_2(w_{t-2}))$
 At position m , the probability = $\lambda_m(w_{t-m}) M_m(w_{t-m}, w_t) \prod_{j=1}^{m-1} (1 - \lambda_j(w_{t-j}))$

Note: $\lambda_m(w) = 1$ for all w

Appendix C: Mixed-order Markov Model (2/2)

- Model Training Using the EM algorithm
 - Expectation

$$\phi_k(t) = \frac{\lambda_k(w_{t-k}) M_k(w_{t-k}, w_t) \prod_{j=1}^{k-1} (1 - \lambda_j(w_{t-j}))}{p(w_t | w_{t-m}, \dots, w_{-1})}$$

- Maximization

$$\lambda_k(w) = \frac{\sum_t \delta(w, w_{t-k}) \phi_k(t)}{\sum_t \sum_{j=k}^m \delta(w, w_{t-k}) \phi_j(t)}$$

$$M_k(w_1, w_2) = \frac{\sum_t \delta(w_1, w_{t-k}) \delta(w_2, w_t) \phi_k(t)}{\sum_t \delta(w_1, w_{t-k}) \phi_k(t)}$$

The raw counts of k -separated bigrams do give good initial estimates