Topic Language Models and their Applications

主題式語言模型及其應用

Berlin Chen (陳柏琳)
Professor, Department of Computer Science & Information Engineering
National Taiwan Normal University

2013/01/11

Introduction

- Language is unarguably the most nuanced and sophisticated medium to express or communicate our thoughts
  - A natural vehicle to convey our thoughts and the content of all wisdom and knowledge
- Language modeling (LM), aiming to capture the regularities in human natural language and quantify the acceptability of a given word sequence, has long been an interesting yet challenging research topic in the speech and language processing community
  - Recently, it also has been introduced to information retrieval (IR) problems, and provided an effective and theoretically attractive (statistical or probabilistic) framework for building IR systems

1. T. Hofmann, "ProbMap - A probabilistic approach for mapping large document collections," *IDA*, 2000
2. B. Chen, "Word topic models for spoken document retrieval and transcription," *ACM TALIP*, 2009

Introduction

- The *n*-gram language model that determines the probability of an upcoming word given the previous *n*-1 word history is the most prominently used

$$P(\mathbf{W} = w_1, w_2, \ldots, w_m)$$
$$= P(w_1)P(w_2|w_1)P(w_3|w_1, w_2)\ldots P(w_m|w_1, w_2, \ldots, w_{m-1})$$
$$= P(w_1)\prod_{i=2}^{m} P(w_i|w_1, w_2, \ldots, w_{i-1})$$

Chain Rule

  - *n*-gram assumption

$$P(w_i|w_1, w_2, \ldots, w_{i-1}) \approx P(w_i|\underbrace{w_{i-n+1}, w_{i-n+2}, \ldots, w_{i-1}}_{\text{History of length } n\text{-}1})$$

$$P(w_i|w_1, w_2, \ldots, w_{i-1}) \approx P(w_i|w_{i-2}, w_{i-1}) \quad \text{Trigram}$$

$$P(w_i|w_1, w_2, \ldots, w_{i-1}) \approx P(w_i|w_{i-1}) \quad \text{Bigram}$$

$$P(w_i|w_1, w_2, \ldots, w_{i-1}) \approx P(w_i) \quad \text{Unigram}$$

R. Rosenfeld, "Two decades of statistical language modeling: Where do we go from here?," *Proceedings of IEEE*, 2000

Introduction

- Known Weakness of *n*-gram Language Models
  - Sensitive to changes in the style or topic of the text on which they are trained
  - Assume the probability of next word in a sentence depends only on the identity of last *n*-1 words
    - Capture only local contextual information or lexical regularity of a language
- Ironically, *n*-gram language models take no advantage of the fact that what is being modeled is language
  - Frederick Jelinek said "*put language back into language modeling*" (1995)

$$P\left(w_i \middle| w_1, w_2, \ldots, w_{i-1}\right) \approx P\left(w_i \middle| w_{i-2}, w_{i-1}\right)$$

F. Jelinek, "The dawn of statistical ASR and MT," Computational Linguistics, 35(4), pp. 483-494, 2009.

Introduction

- Topic language models have been introduced and investigated to complement the *n*-gram language models
  - A commonality among them is that a set of latent topic variables is introduced to describe the "**word-document**" co-occurrence characteristics
- Models developed generally follow two lines of thought
  - Algebraic
    - Latent Semantic Analysis (LSA) (Deerwester et al., 1990), nonnegative matrix factorization (NMF) (Lee and Seung, 1999), etc.
  - Probabilistic
    - Probabilistic latent semantic analysis (PLSA) (Hofmann, 2001), latent Dirichlet allocation (LDA) (Blei et al., 2003), etc.

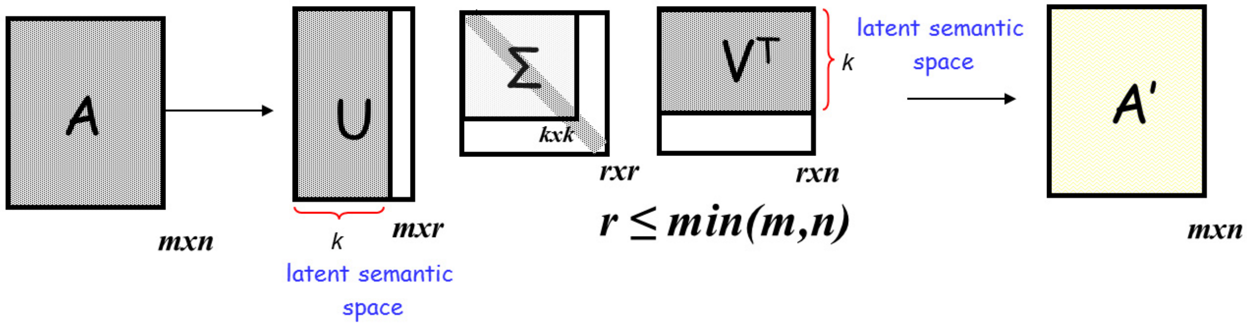
Typical Issues for LM

- Evaluation
  - How can you tell a good language model from a bad one
  - Run a speech recognizer or adopt other statistical measurements
- Smoothing
  - Deal with data sparseness of real training data
  - Various approaches have been proposed
- Caching/Adaptation
  - If you say something, you are likely to say it again later
  - Adjust word frequencies observed in the current conversation
- Clustering
  - Group words with similar properties (similar semantic or grammatical) into the same class
  - Another efficient way to handle the data sparseness problem

NTNU

Latent Semantic Analysis (LSA)

- Start with a matrix describing the intra- and Inter-document statistics between all terms and all documents
- Singular value decomposition (SVD) is then performed on the matrix to project all term and document vectors onto a reduced latent topical space



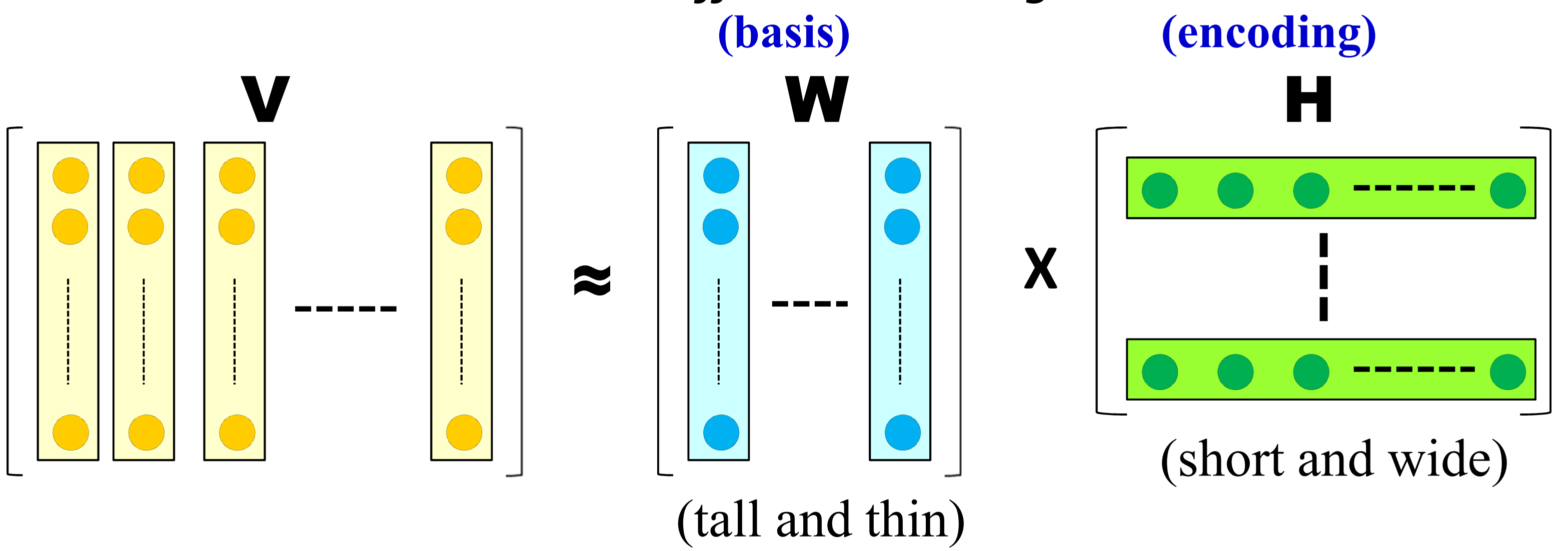
$$\|A\|_F^2 = \sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}^2 \Rightarrow \|A\|_F^2 = \sigma_1^2 + \sigma_2^2 + \ldots + \sigma_r^2 \quad ?$$

- In the context of IR, matching between queries and documents can be carried out in this topical space

S. Deerwester, "Indexing by latent semantic analysis", *JASIS*, 1990.

Nonnegative Matrix Factorization (NMF)

- NMF approximates data with an **additive and linear combination** of nonnegative components (or basis vectors)
  - Given a **nonnegative data matrix** $\mathrm{V} \in \mathrm{R}^{L \times M}$, NMF computes another two **nonnegative matrices** $\mathrm{W} \in \mathrm{R}^{L \times r}$ and $\mathrm{H} \in \mathrm{R}^{r \times M}$ such that $\mathrm{V} \approx \mathrm{WH}$
    - *r<< L and r<< M to ensure efficient encoding*

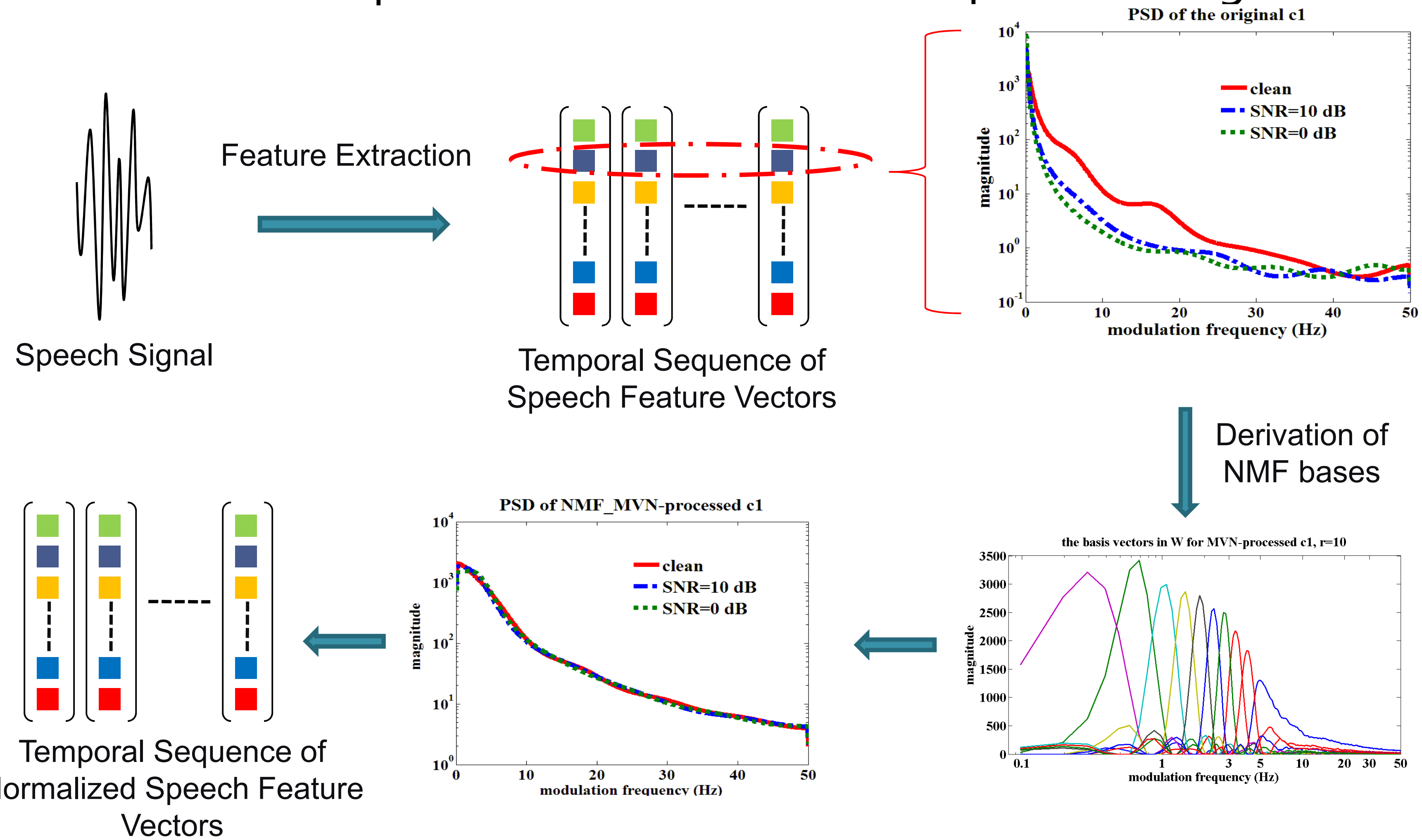


$$\mathbf{v} \approx \mathbf{Wh} = \sum_{r=1}^{R} h_r \mathbf{w}_r = h_1 \mathbf{w}_1 + \ldots + h_R \mathbf{w}_R$$

1. D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, 1999.
2. W.-Y. Chu, et al., "Modulation spectrum factorization for robust speech recognition," *APSIPA ASC*, 2011.

Nonnegative Matrix Factorization (NMF)

- Modulation Spectrum Factorization for Speech Recognition



1. D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, 1999.
2. W.-Y. Chu, et al., "Modulation spectrum factorization for robust speech recognition," *APSIPA ASC*, 2011.

Probabilistic Latent Semantic Analysis (PLSA)

- Each document as a whole consists of a set of shared latent topics with different weights -- A document topic model (DTM)
  - Each topic in turn offers a unigram (multinomial) distribution for observing a given word

$$P_{\text{PLSA}}(w \mid D) = \sum_{k=1}^{K} P(w_i \mid T_k) P(T_k \mid D)$$

- LDA (latent Dirichlet allocation) differs from PLSA mainly in the inference of model parameters:
  - PLSA assumes the model parameters are fixed and unknown
  - LDA places additional a priori constraints on the model parameters, i.e., thinking of them as random variables that follow some Dirichlet distributions

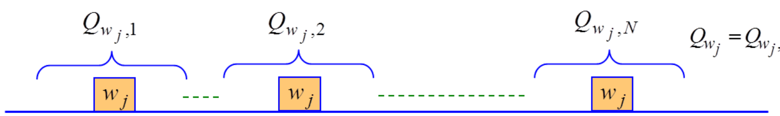
1. T. Hoffmann, "Unsupervised learning by probabilistic latent semantic analysis," Machine Learning, 2001.
2. D. M. Blei et al., "Latent Dirichlet allocation," *Journal of Machine Learning Research*, 2003.

Word Topic Model (WTM)

- Each word of language is treated as a word topic model (WTM) for predicting the occurrences of other words

$$P_{\mathrm{WTM}}\left(w_i \mid \mathrm{M}_{w_j}\right) = \sum_{k=1}^{K} P\left(w_i \mid T_k\right) P\left(T_k \mid \mathrm{M}_{w_j}\right)$$

- The WTM $P_{\mathrm{WTM}}\left(w_i \mid \mathrm{M}_{w_j}\right)$ of each word can be trained with maximum likelihood estimation (MLE)
  - By concatenating those words occurring within a context window around each occurrence of the word, which are assumed to be relevant to the word, to form the training observation



$Q_{w_j} = Q_{w_j,1}, Q_{w_j,2}, \cdots, Q_{w_j,N}$

$$\log L_{\mathbf{W}} = \sum_{w_j \in \mathbf{W}} \log P_{\mathrm{WTM}}\left(Q_{w_j} \middle| \mathrm{M}_{w_j}\right) = \sum_{w_j \in \mathbf{W}} \sum_{w_i \in Q_{w_j}} c\left(w_i, Q_{w_j}\right) \log P_{\mathrm{WTM}}\left(w_i \middle| \mathrm{M}_{w_j}\right)$$

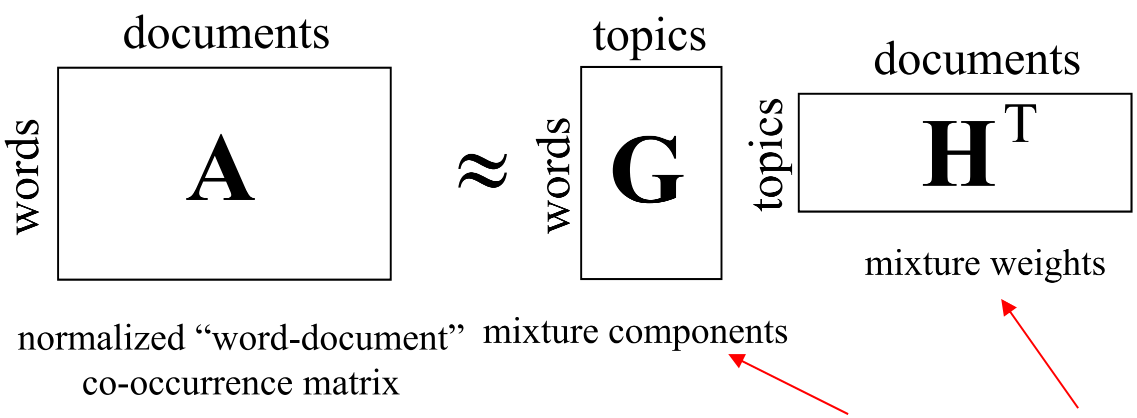
- $\mathbf{W}$: the set of words in the language

Can we model topic al information using other units beyond "documents" ?

Comparison Between WTM and DTM

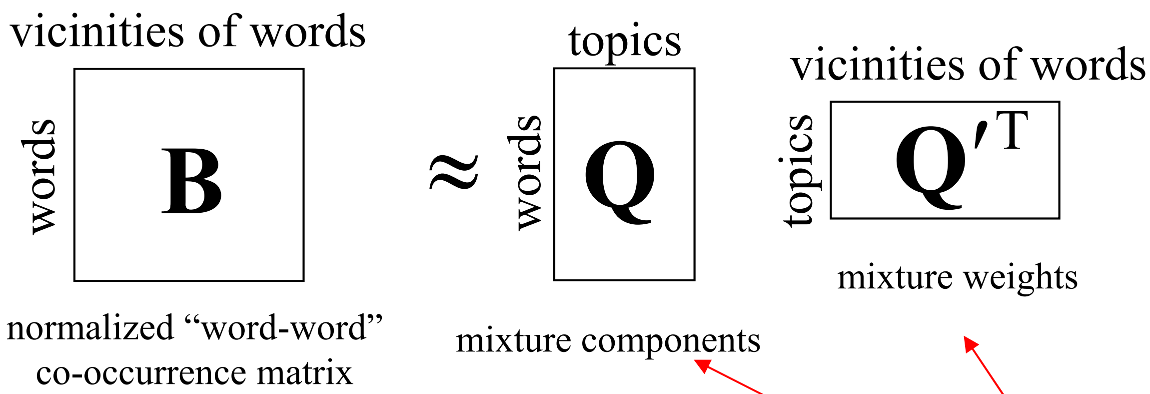
- Probabilistic Matrix Decompositions

PLSA/LDA



$$P_{\text{PLSA/LDA}}\left(w_i \mid D\right) = \sum_{k=1}^{K} P\left(w_i \mid T_k\right) P\left(T_k \mid D\right)$$

WTM



$$P_{\text{WTM}}\left(w_i \mid \mathrm{M}_{w_j}\right) = \sum_{k=1}^{K} P\left(w_i \mid T_k\right) P\left(T_k \mid \mathrm{M}_{w_j}\right)$$

B. Chen, "Word topic models for spoken document retrieval and transcription," *ACM Transactions on Asian Language Information Processing, 8(1), 2009.*

Example Topic Distributions of WTM

Topic 13	
word	**weight**
Vena (靜脈)	1.202
Resection (切除)	0.674
Myoma (肌瘤)	0.668
Cephalitis (腦炎)	0.618
Uterus (子宮)	0.501
Bronchus (支氣管)	0.500

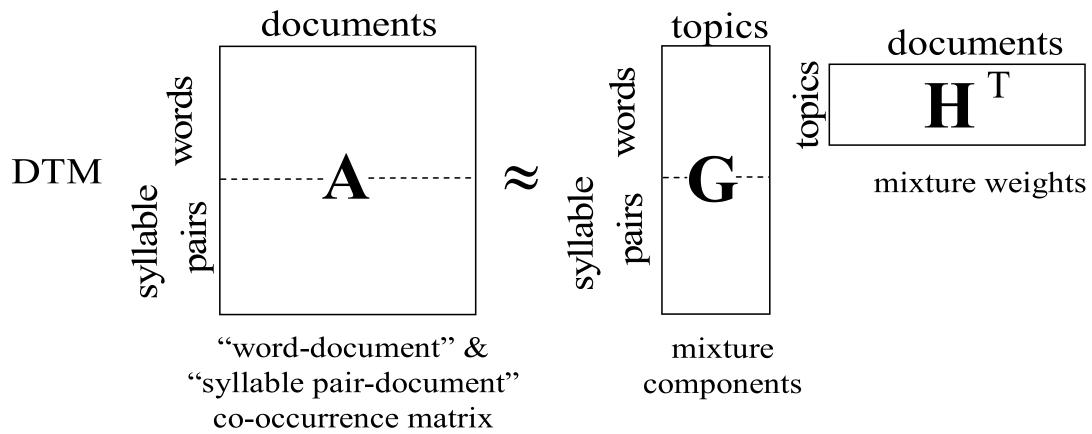
Topic 14	
word	**weight**
Land tax (土地稅)	0.704
Tobacco and alcohol tax law (菸酒稅法)	0.489
Tax (財稅)	0.457
Amend drafts (修正草案)	0.446
Acquisition (購併)	0.396
Insurance law (保險法)	0.373

Topic 23	
word	**weight**
Cholera (霍亂)	0.752
Colorectal cancer (大腸直腸癌)	0.681
Salmonella enterica (沙門氏菌)	0.471
Aphtae epizooticae (口蹄疫)	0.337
Thyroid (甲狀腺)	0.303
Gastric cancer (胃癌)	0.298

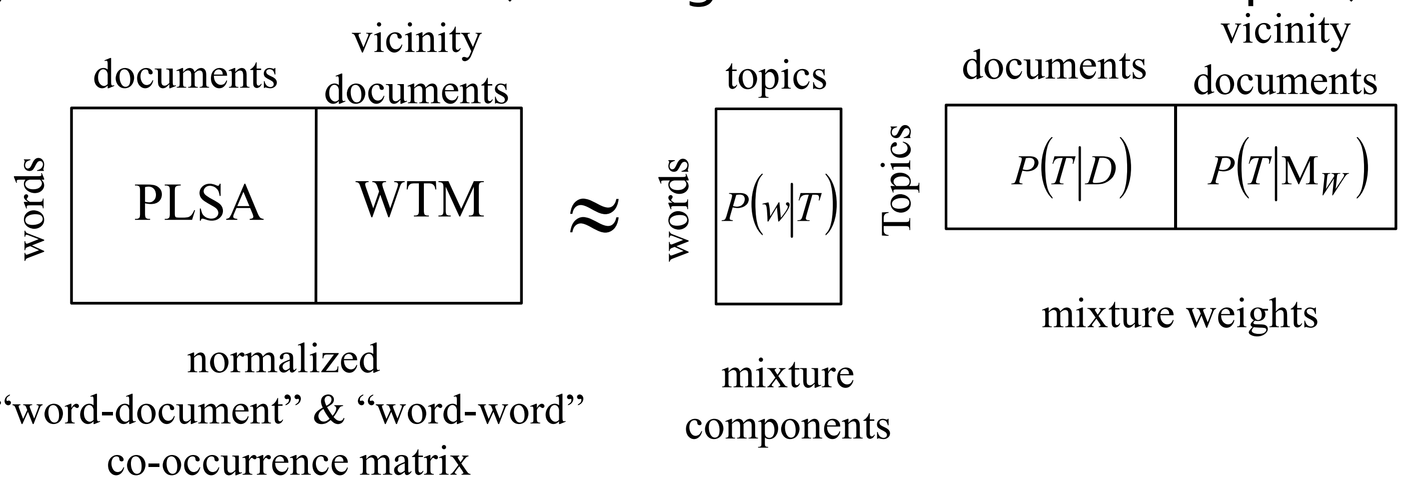
N T N U

Some Extensions of DTM and WTM

- Hybrid of Different Indexing Features for DTM/WTM



- Pairing of DTM and WTM (Sharing the Same Latent Topics)



S.-H. Lin and B. Chen, "Topic modeling for spoken document retrieval using word- and syllable-level information," *SSCS 2009*.

Visualization of Document Collections with PLSA

- The original formulation of PLSA

$$P_{\text{PLSA}}(w \mid D) = \sum_{k=1}^{K} P(w_i \mid T_k) P(T_k \mid \text{D})$$

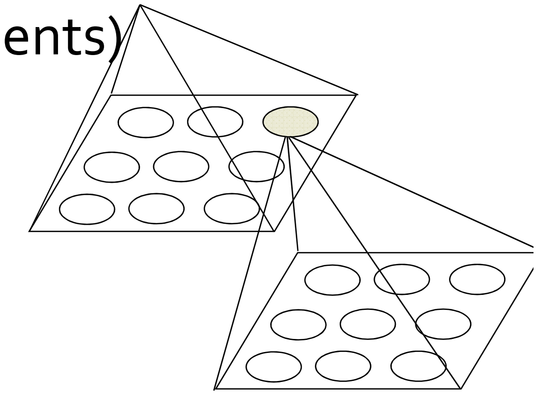
- ProbMap: PLSA additionally takes into account the relationships between topics

$$P_{\text{ProbMap}}(w \mid D) = \sum_{k=1}^{K} \left[\sum_{j=1}^{K} P(w \mid T_j) P(T_j \mid T_k) \right] P(T_k \mid \text{D})$$

  - Where $P(T_j \mid T_k)$ has to do with the topological distance between any two topics (or clusters of documents)

$$E(T_l, T_k) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{dist(T_l, T_k)^2}{2\sigma^2} \right]$$

$$P(T_j | T_k) = \frac{E(T_j, T_k)}{\sum_{j'=1}^{K} E(T_s, T_k)}$$



Two-dimensional
Tree Structure for Organized Topics

T. Hofmann, "ProbMap - A Probabilistic Approach for Mapping Large Document Collections," *IDA*, 2000.

Visualization of Document Collections with PLSA

- Estimation of the Component Distributions (with EM algorithm)

$$\hat{P}(w \mid T_k) = \frac{\sum_{i=1}^{N} c(w, D_i) P_U(T_k \mid w, D_i)}{\sum_{j=1}^{M} \sum_{h=1}^{N} c(w_j, D_h) P_U(T_k \mid w_j, D_h)}$$

$$\hat{P}(T_k \mid D_i) = \frac{\sum_{j=1}^{M} c(w_j, D_i) P_V(T_k \mid w_j, D_i)}{\sum_{j'=1}^{M} c(w_{j'}, D_i)}$$

  - Where

$$P_U(T_k \mid w, D_i) = \frac{P(w \mid T_k) \cdot P(T_k \mid D_i)}{\sum_{m=1}^{K} P(w \mid T_m) \cdot P(T_m \mid D_i)}$$

$$P_V(T_k \mid w, D_i) = \frac{P(T_k \mid D_i) \sum_{k'=1}^{K} P(T_{k'} \mid T_k) P(w \mid T_{k'})}{\sum_{s=1}^{K} P(T_s \mid D_i) \sum_{l=1}^{K} P(T_l \mid T_s) P(w \mid T_l)}$$

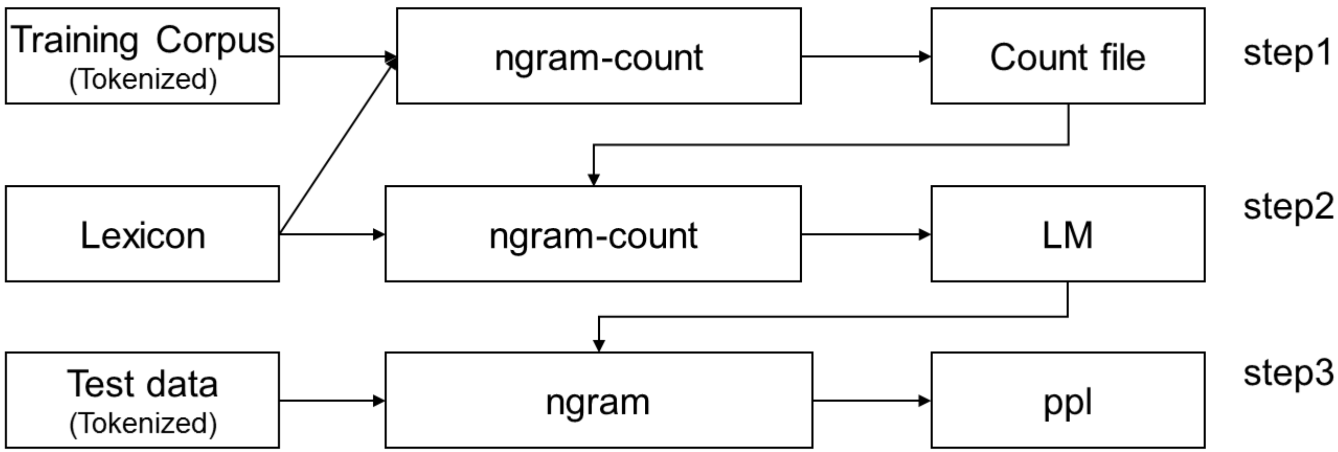
- Selection of Representative Topic Words

$$S(w, T_k) = \frac{\sum_{i=1}^{N} c(w, D_i) P(T_k \mid D_i)}{\sum_{i'=1}^{N} c(w, D_{i'}) [1 - P(T_k \mid D_{i'})]}$$

L.-S. Lee and B. Chen, "Spoken document understanding and organization," *IEEE Signal Processing Magazine*, 2005.

Commonly-used Language Modeling Toolkit

- For example, SRILM is a toolkit for building and applying various statistical language models
  - Three main functionalities
    - Generate the *n*-gram count file from the corpus
    - Train the language model from the *n*-gram count file
    - Calculate the test data perplexity using the trained language model



A. Stolcke, "SRILM - An Extensible Language Modeling Toolkit," *Interspeech*, 2002.

Other Families of Language Models

- Discriminative Language Model
- Neural Network Language Model
- Relevance Model
- Positional Language Model

Discriminative Language Model (DLM)

- DLM for Speech Recognition
  - DLM takes a testing utterance X together with a set of top-scoring recognition hypotheses $\mathbf{GEN}(X)$, produced by the baseline speech recognition system, as the input
  - DLM selects the most promising hypothesis W^* out from $\mathbf{GEN}(X)$ through the following equation:

$$W^* = \mathrm{DLM}\big(X, \mathbf{GEN}(X)\big) = \underset{W \in \mathbf{GEN}(X)}{\arg\max}\, \mathbf{\Phi}(X, W) \bullet \boldsymbol{\alpha}$$

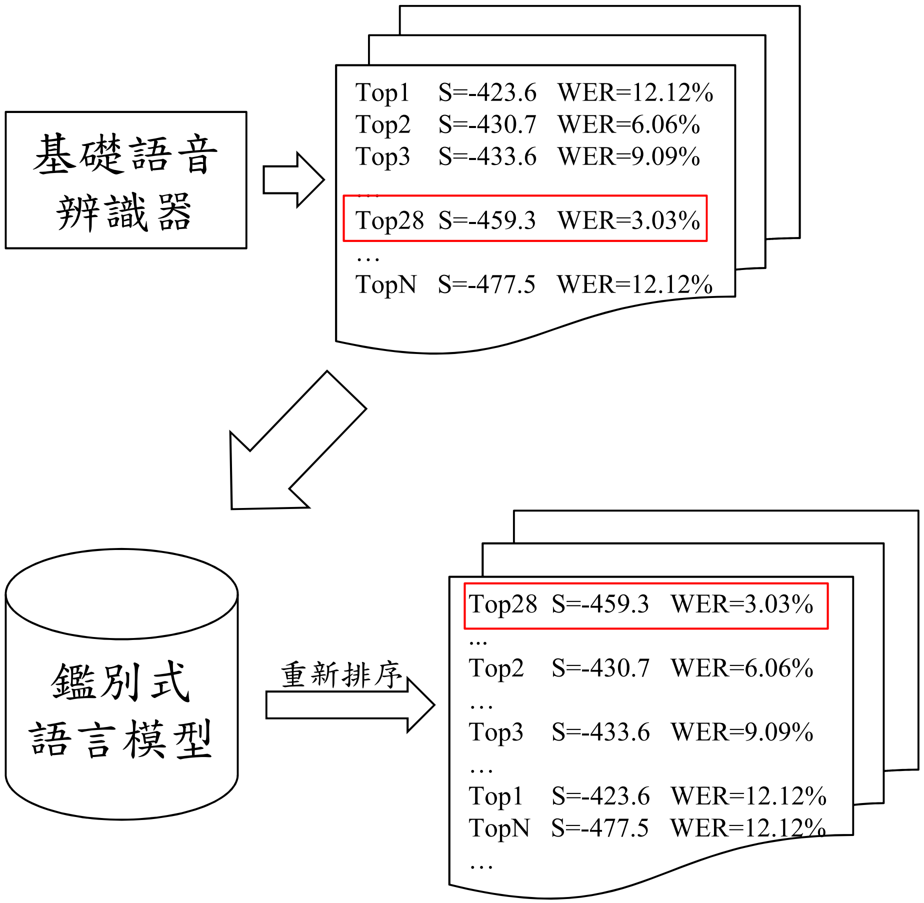
    - Where $\mathbf{\Phi}(X,W)$ is a feature vector used to characterize a recognition hypothesis W for X, and $\boldsymbol{\alpha}$ is the parameter vector of a DLM model

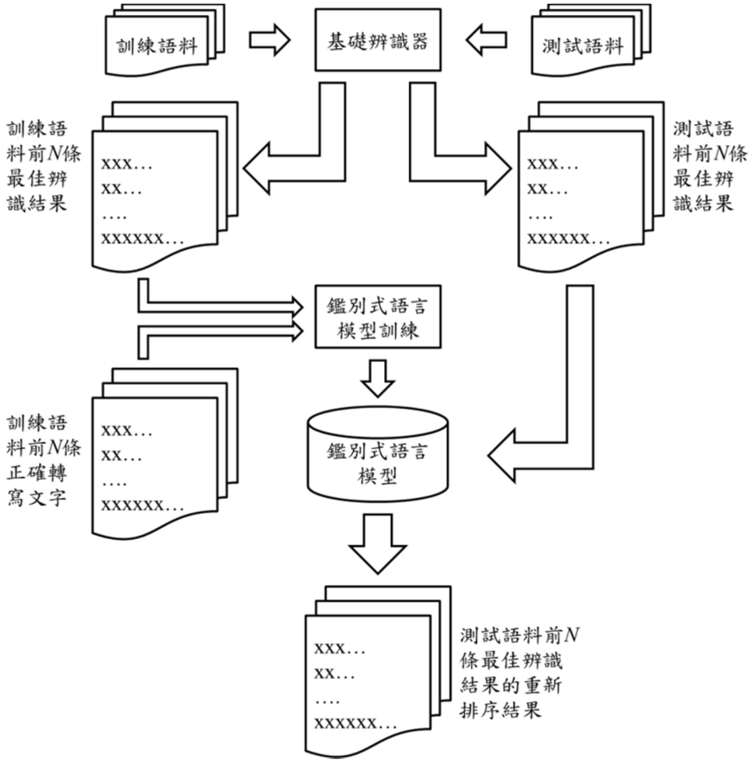
		word unigrams				word bigrams			
	$\log[P(W)P(W \mid x)]$	w_p	w_q	...	w_t	$w_p w_k$	...	$w_j w_m$	$w_l w_m$
Feature Vector $\mathbf{\Phi}(X,W)$	-2602.62	1	3	$\cdots$	0	2	$\cdots$	1	0
Parameter Vector of DLM $\boldsymbol{\alpha}$	1	0.01	0.12	$\cdots$	-0.25	-0.03	$\cdots$	0.78	0.52

B. Roark et al., "Discriminative n-gram language modeling," *Computer Speech and Language, 21, 2007.*

Discriminative Language Model

- Schematic Illustration





Discriminative Language Model

- Training of a DLM model
  - Fulfilled by finding a parameter vector $\boldsymbol{\alpha}$ that minimizes a loss function having to do with the margin between the score of the reference transcript W_i^R and that of any other hypothesis W_i for each training utterance X_i

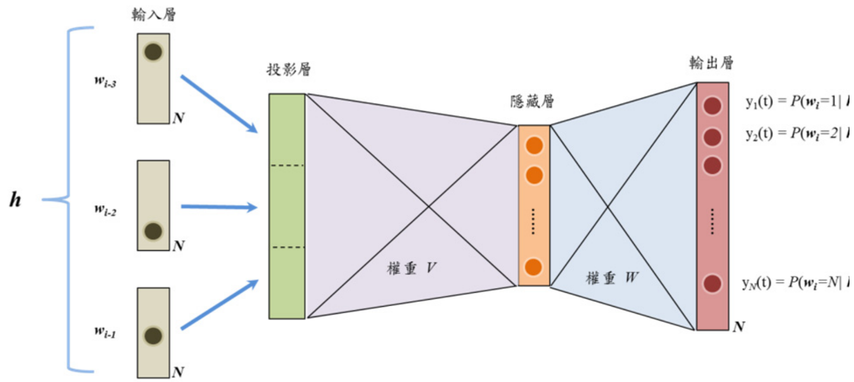
The Training Objectives of Various DLM Methods

Methods	Training Objectives
Perceptron	$F_{Perc}(\boldsymbol{\alpha}) = \frac{1}{2}\sum_{i=1}^{L}\left(\left(\boldsymbol{\Phi}(X_i, W_i^R) - \boldsymbol{\Phi}(X_i, W_i^*)\right)\bullet\boldsymbol{\alpha}\right)$
GCLM	$F_{GCLM}(\boldsymbol{\lambda}) = -\sum_{i=1}^{L}\log\frac{\exp\left(\boldsymbol{\Phi}(X_i, W_i^R)\bullet\boldsymbol{\alpha}\right)}{\sum_{W_i\in\mathbf{GEN}(X_i)}\exp\left(\boldsymbol{\Phi}(X_i, W_i)\bullet\boldsymbol{\alpha}\right)}$
WGCLM	$F_{WGCLM}(\boldsymbol{\lambda}) = -\sum_{i=1}^{L}\log\frac{\exp\left(\boldsymbol{\Phi}(X_i, W_i^R)\bullet\boldsymbol{\alpha}\right)}{\sum_{W_i\in\mathbf{GEN}(X_i)}\omega_{i,W_i}\exp\left(\boldsymbol{\Phi}(X_i, W_i)\bullet\boldsymbol{\alpha}\right)}$
MERT	$F_{MERT}(\boldsymbol{\lambda}) = \sum_{i=1}^{L}\sum_{W_i\in\mathbf{GEN}(X_i)}\frac{\varpi_{i,W_i}\exp\left(\boldsymbol{\Phi}(X_i, W_i)\bullet\boldsymbol{\alpha}\right)^{\beta}}{\sum_{W_s\in\mathbf{GEN}(X_i)}\exp\left(\boldsymbol{\Phi}(X_i, W_s)\bullet\boldsymbol{\alpha}\right)^{\beta}}$

1. B. Chen, J.-W. Liu, "Discriminative language modeling for speech recognition with relevance information," *ICME*, 2011
2. M.-H. Lai et al., "Empirical comparisons of various discriminative language models for speech recognition," *ROCLING*, 2011

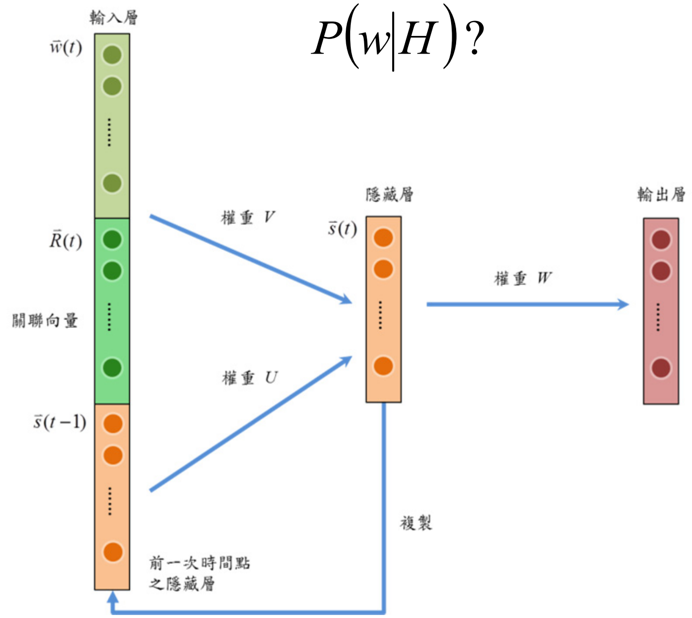
Neural Network Language Model (NNLM)

- Schematic Illustrations
  - Feed-forward neural networks



  - Recurrent neural networks

$P(w|H)$?

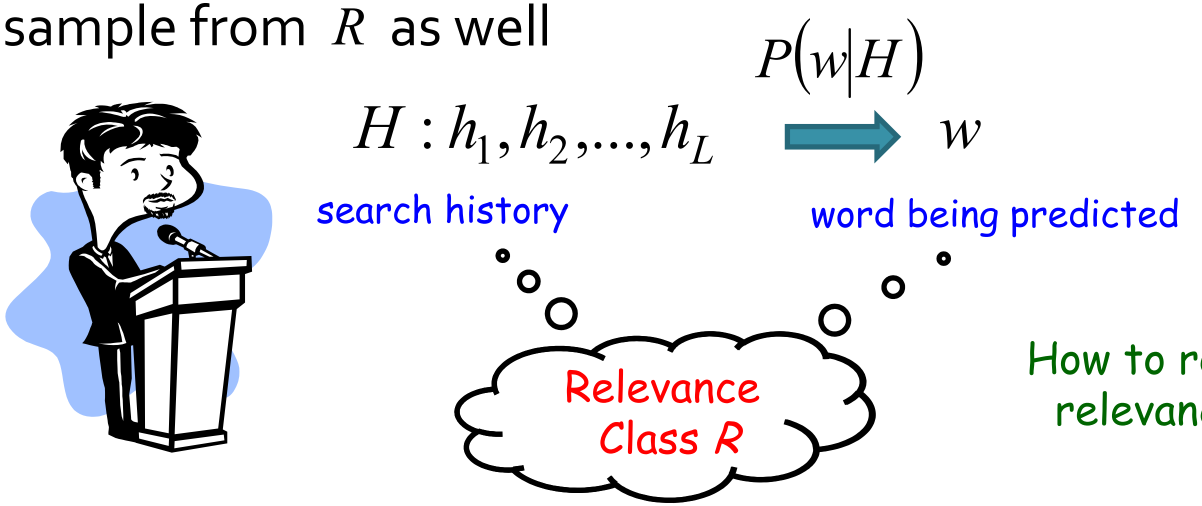


- Research Issues
  - Encoding of words (and history)
  - Leveraging extra information cues
  - Discriminative training of NNLM

1. T. Mikolov et al., "Recurrent neural network based language model," *Interspeech 2010*
2. B.-X. Huang et al., "Recurrent neural network-based language modeling with relevance information," *ROCLING 2012*

Relevance Model (RM)

- Investigate a novel use of relevance information cues to dynamically complement (or adapt) the conventional *n*-gram models, assuming that
  - During speech recognition, a search history $H = h_1, h_2, \ldots, h_L$ is a sample from a relevance class R describing some semantic content
  - Assume that a probable word w that immediately succeeds H is a sample from R as well



How to represent the relevance class R?

B. Chen and K.-Y. Chen, "Leveraging relevance cues for language modeling in speech recognition," to appear in *Information Processing & Management*, 2013

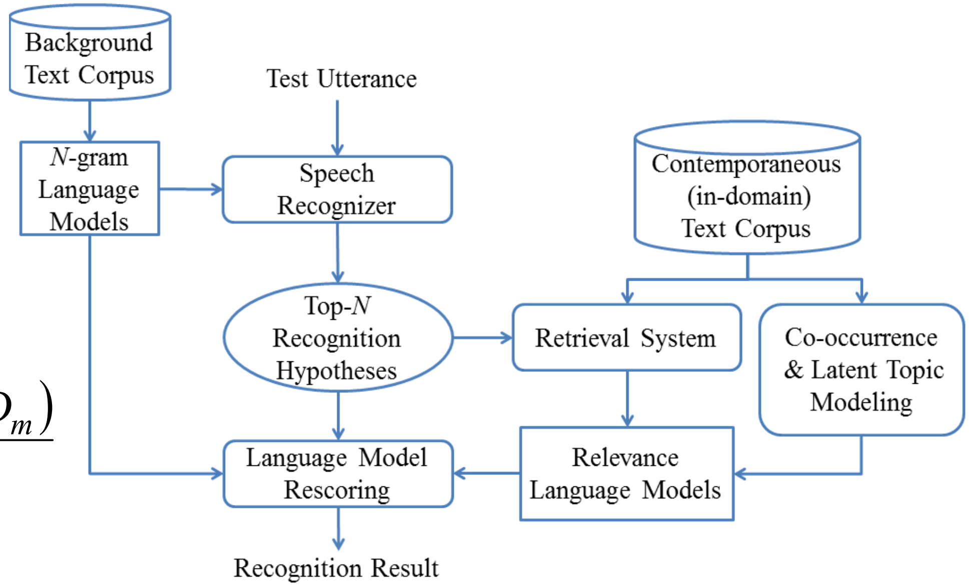
Relevance Model

- Leverage the top-*M* relevant documents of the search history to approximate the relevance class R
  - Take H as a query to retrieve relevant documents
  - **R**elevance **M**odel: Multinomial view (*bag-of-words modeling*) of R

$$P_{\text{RM}}(w|H)=\frac{P_{\text{RM}}(H,w)}{P_{\text{RM}}(H)}$$

$$=\frac{\sum_{m=1}^{M} P(D_m)P(H,w\mid D_m)}{\sum_{m=1}^{M} P(D_m)P(H\mid D_m)}$$

$$=\frac{\sum_{m=1}^{M} P(D_m)P(w\mid D_m)\prod_{l=1}^{L} P(h_l\mid D_m)}{\sum_{m=1}^{M} P(D_m)\prod_{l=1}^{L} P(h_l\mid D_m)}$$



$$P_{\text{Adapt}}(w|H)=\lambda\cdot P_{\text{RM}}(w|H)+(1-\lambda)\cdot P_{\text{BG}}(w|h_{L-1},h_L)$$

Relevance Model

- Further incorporation of latent topic information
  - A shared set of latent topic variables $\{T_1, T_2, \ldots, T_K\}$ is used to describe "*word-document*" co-occurrence characteristics

$$P(w \mid D_m) = \sum_{k=1}^{K} P(w \mid T_k) P(T_k \mid D_m)$$

$$P_{\mathrm{TRM}}(H, w) = \sum_{m=1}^{M} \sum_{k=1}^{K} P(D_m) P(T_k \mid D_m) P(w \mid T_k) \prod_{l=1}^{L} P(h_l \mid T_k)$$

- Alternative modeling of pairwise word associations

$$P_{\mathrm{PRM}}(h_l, w) = \sum_{m=1}^{M} P(D_m) P(h_l \mid D_m) P(w \mid D_m)$$

$$P_{\mathrm{PRM}}(w \mid H) = \sum_{l=1}^{L} \alpha_l \cdot P_{\mathrm{PRM}}(w \mid h_l)$$

$$P_{\mathrm{TPRM}}(h_l, w) = \sum_{m=1}^{M} \sum_{k=1}^{K} P(D_m) P(T_k \mid D_m) P(h_l \mid T_k) P(w \mid T_k)$$

Relevance Model

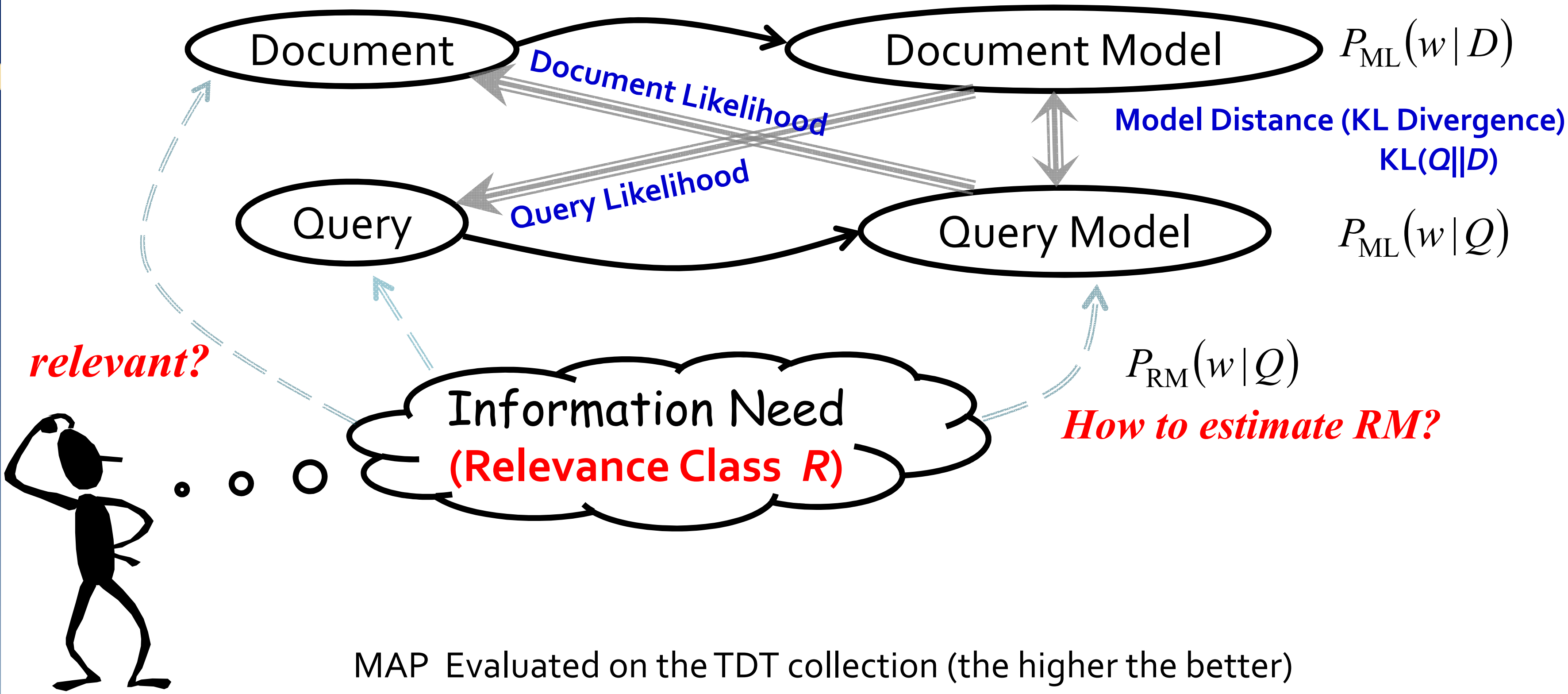
- Tested on a large vocabulary broadcast new recognition task
  - Character error rate (CER) results (the lower the better)

n-gram	RM	TRM	PRM	TPRM	PLSA	LDA	Cache	TBLM
20.08	19.29	19.08	19.23	19.09	19.15	19.15	19.86	20.02

  - The various RM models achieve results compared to PLSA and LDA (topic models) and are considerably better than Cache and TBLM (trigger-based language model)
  - The various RM models are more efficient than PLSA and LDA
    - The various RM probabilities can be easily composed on the basis of the component probability distributions that were trained beforehand, without recourse to any complex inference procedure during the recognition (or rescoring) process
      - Computationally tractable and feasible for speech recognition

RM for Spoken Document Retrieval

- Schematic illustration



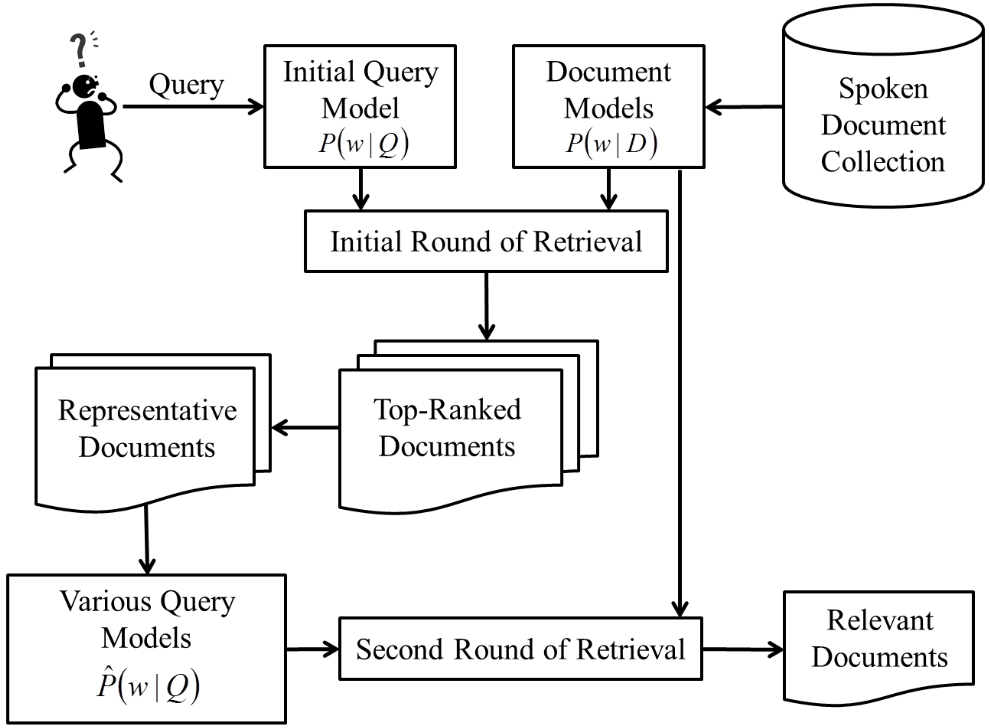
MAP Evaluated on the TDT collection (the higher the better)

ULM	RM	TRM	RM+NR	TRM+NR	PLSA	LDA
0.323	0.364	0.394	0.392	0.402	0.345	0.341

B. Chen et al., "Spoken document retrieval with unsupervised query modeling techniques," *IEEE Transactions on Audio, Speech and Language Processing*, 20(9), 2012.

RM for Spoken Document Retrieval

- Effective Pseudo-relevance Feedback
  - How to effectively glean useful cues from the top-ranked documents so as to achieve more accurate relevance (query) modeling?

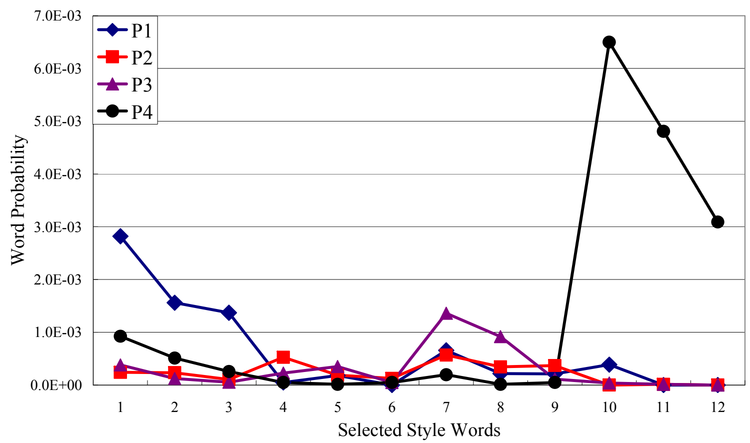


Considering relevance, non-relevance, diversity and density cues

Positional Language Model

- Are there any other alternatives beyond the above LMs?
- The table below shows the style words with higher rank of *TF-IDF* scores on four partitions of the broadcast news corpus
  - The corpus was partitioned by a left-to-right HMM segmenter

P1	P2	P3	P4
1繼續 Continue	4醫師 Doctor	7學生 Student	10公視 TV station name
2現場 Locale	5網路 Internet	8老師 Teacher	11綜合報導 Roundup
3歡迎 Welcome	6珊瑚 Coral	9酒 Rice wine	12編譯 Edit and translate



Positional Language Model

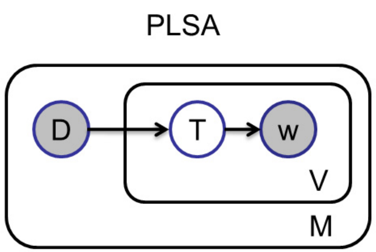
- Positional *n*-gram Model

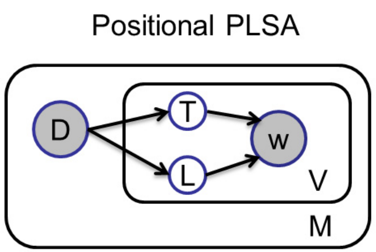
$$P_{POS}\left(w_i \mid w_{i-2}, w_{i-1}\right) = \sum_{s=1}^{S} \alpha_s P\left(w_i \mid w_{i-2}, w_{i-1}, L_s\right)$$

  - Where S is the number of partitions, α_S is the weight for a specific position L_S

- Positional PLSA (Probabilistic Latent Semantic) Model

$$P_{PosPLSA}\left(w_i \middle| H\right) = \sum_{s=1}^{S} \sum_{k=1}^{K} P\left(w_i \middle| T_k, L_s\right) P\left(L_s \middle| H\right) P\left(T_k \middle| H\right)$$





Graphical Model Representations

Conclusions

- Various language modeling approaches have been proposed and extensively investigated in the past decade, showing varying degrees of success in a wide array of applications (cross-fertilization between speech, NLP and IR communities)
- Among them, topic modeling, discovering the latent semantic (or topical) structures of document collections, is deemed to be the key for analysis and understanding of documents
- Modeling and computation are intertwined in developing new language models ("simple" is "elegant"?)
- "*Put language back into language modeling*" remains an important issue that awaits further studies (our ultimate goal?)

D. Blei, "Probabilistic topic models," *Communications of the ACM*, 55(4):77–84, 2012.

Thank You!