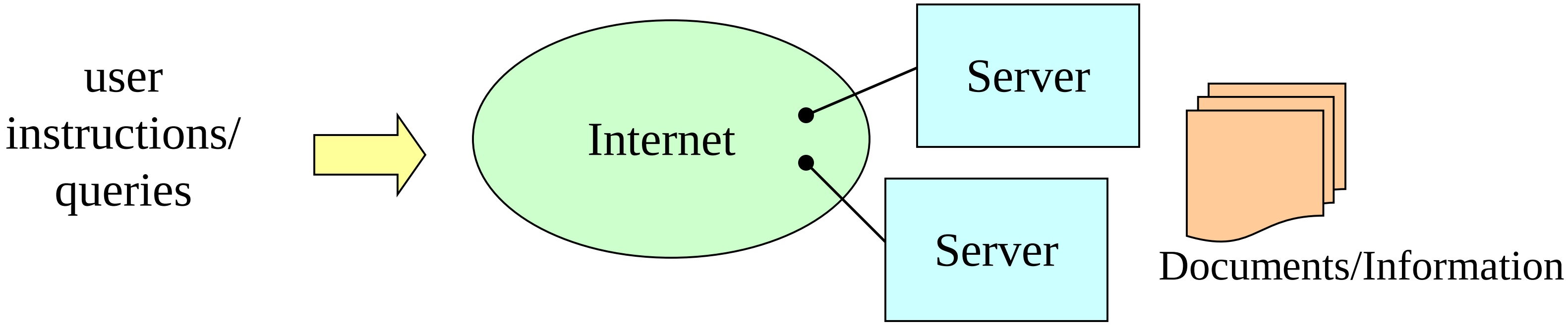
10.0 Speech-based Information Retrieval

Text/Speech-based Information Retrieval

- **Text-based information retrieval extremely successful**

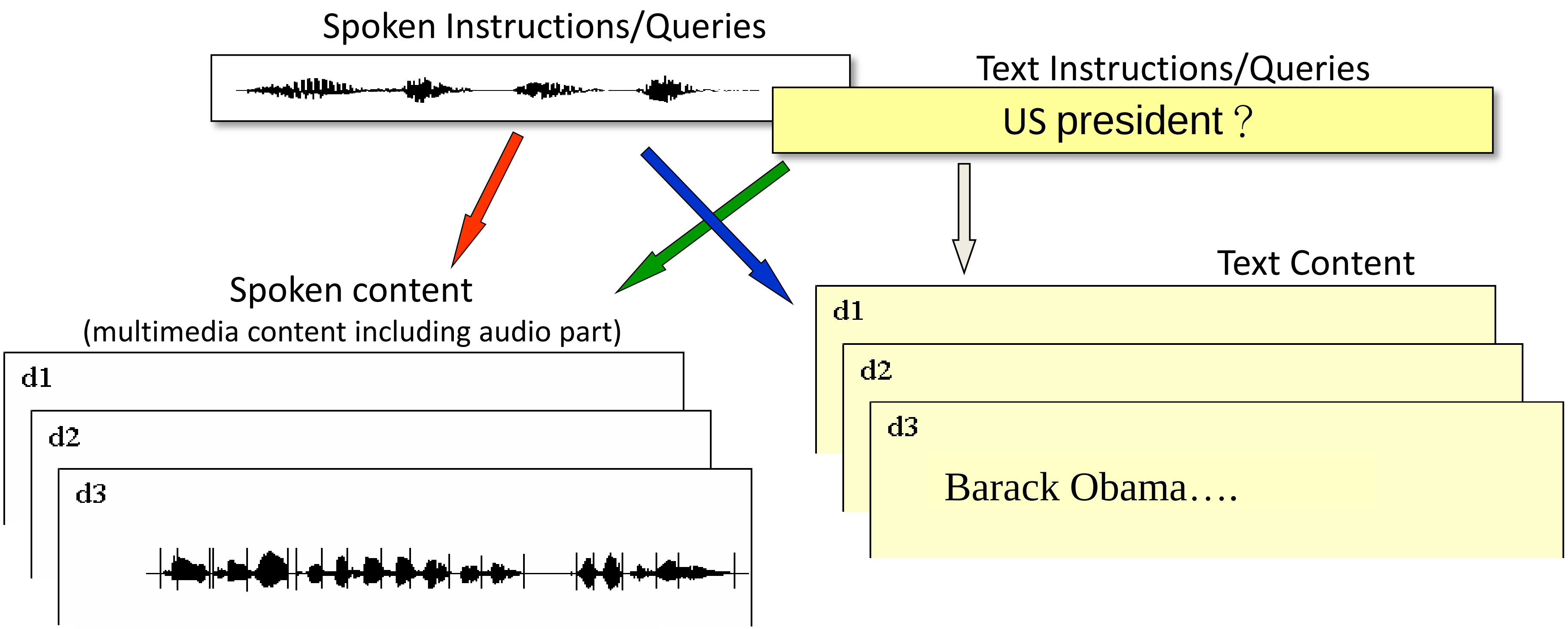


  - information desired by the users can be obtained very efficiently
  - all users like it
  - producing very successful industry

- **All roles of texts can be accomplished by voice**
  - spoken content or multimedia content with voice in audio part
  - voice instructions/queries via handheld devices

- **Speech-based information retrieval**

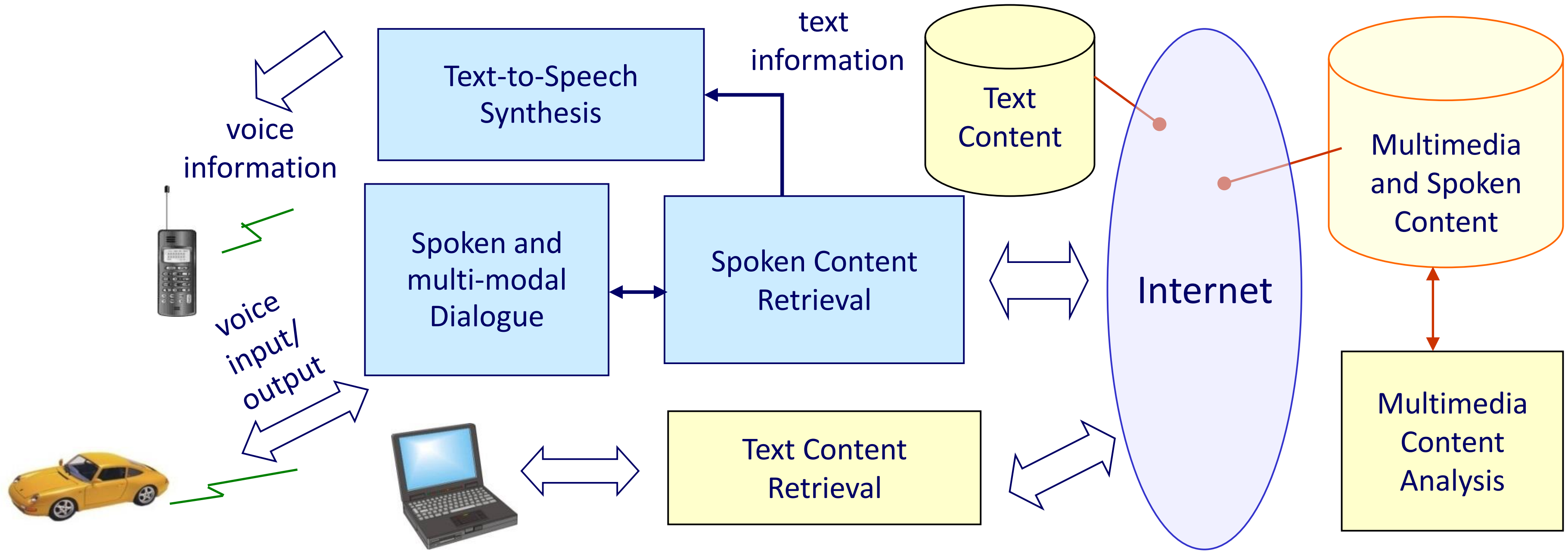
Speech-based Information Retrieval



- **User instructions and/or network content can be in form of voice**
  - text queries/spoken content : spoken document retrieval, spoken term detection
  - spoken queries/text content : voice search
  - spoken queries/spoken content : query by example

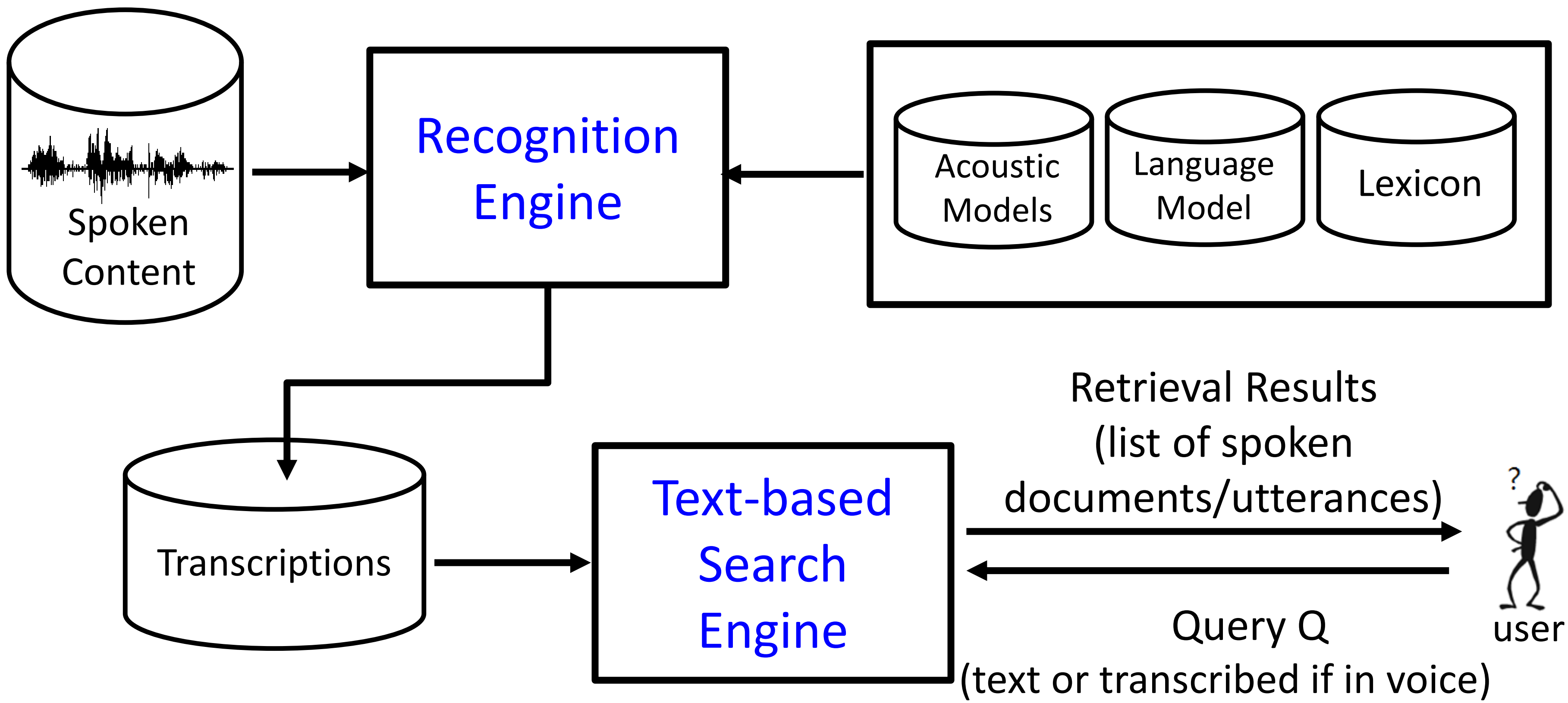
[spoken content retrieval]

Wireless and Multimedia Technologies are Creating An Environment for Speech-based Information Retrieval



- Many hand-held devices with multimedia functionalities available
- Unlimited quantities of multimedia content fast growing over the Internet
- User-content interaction necessary for retrieval can be accomplished by spoken and multi-modal dialogues
- Network access is primarily text-based today, but almost all roles of texts can be accomplished by voice

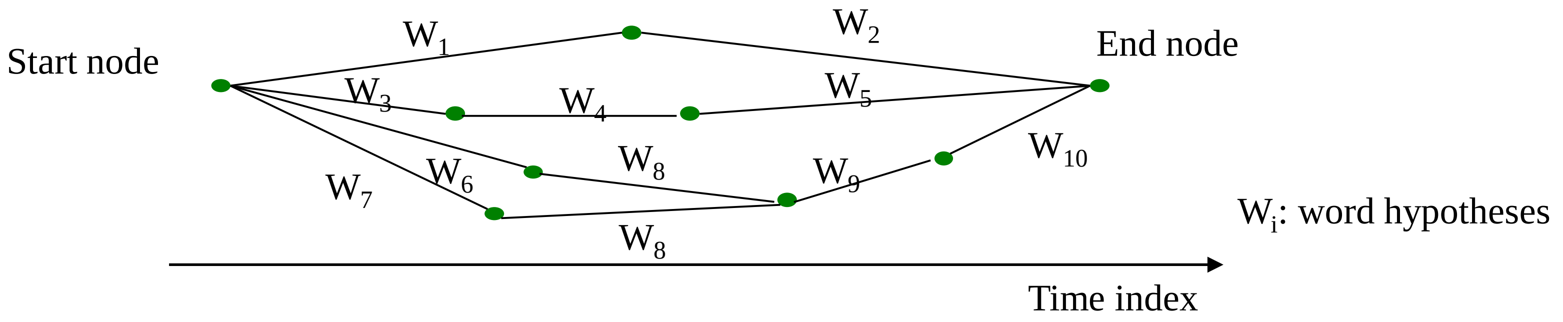
Basic Approach for Spoken Content Retrieval



- **Transcribe the spoken content**
- **Search over the transcriptions as they are texts**
- **Recognition errors cause serious performance degradation**

Lattices for Spoken Content Retrieval

- **Low recognition accuracies for spontaneous speech including Out-of-Vocabulary (OOV) words under adverse environment**
  - considering lattices with multiple alternatives rather than 1-best output



➢ higher probability of including correct words, but also including more noisy words

➢ correct words may still be excluded (OOV and others)

➢ huge memory and computation requirements

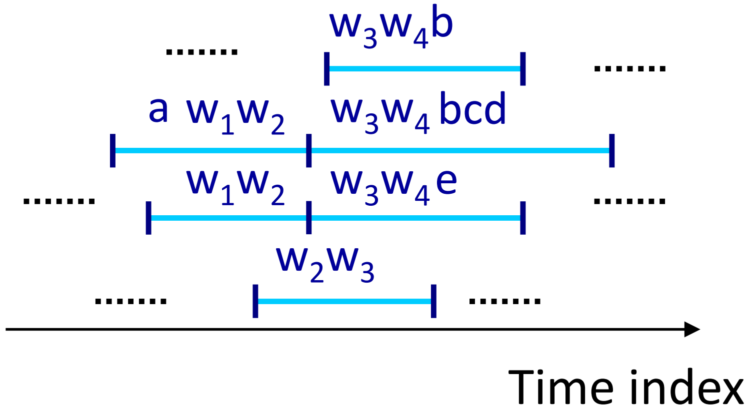
Other Approach Examples in addition to Lattices

- **Confusion Matrices**
  - use of confusion matrices to model recognition errors and expand the query/document, etc.
- **Pronunciation Modeling**
  - use of pronunciation models to expand the query, etc.
- **Fuzzy Matching**
  - query/content matching not necessarily exact

OOV or Rare Words Handled by Subword Units

- OOV Word $W=w_1w_2w_3w_4$ can't be recognized and never appears in lattice
  - w_i : subword units : phonemes, syllables…
  - a, b, c, d, e : other subword units

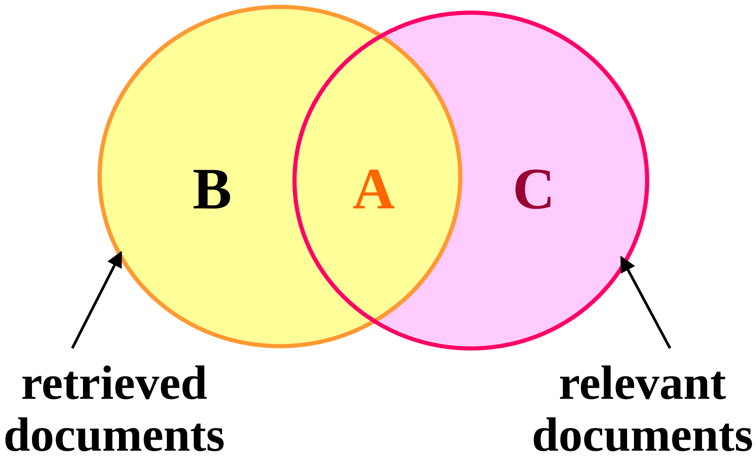
Lattice:



- $W=w_1w_2w_3w_4$ hidden at subword level
  - can be matched at subword level without being recognized
- Frequently Used Subword Units
  - Linguistically motivated units: phonemes, syllables/characters, morphemes, etc.
  - Data-driven units: particles, word fragments, phone multigrams, morphs, etc.

Performance Measures (1/2)

- **Recall and Precision Rates**



Precision rate = $\dfrac{A}{A+B}$

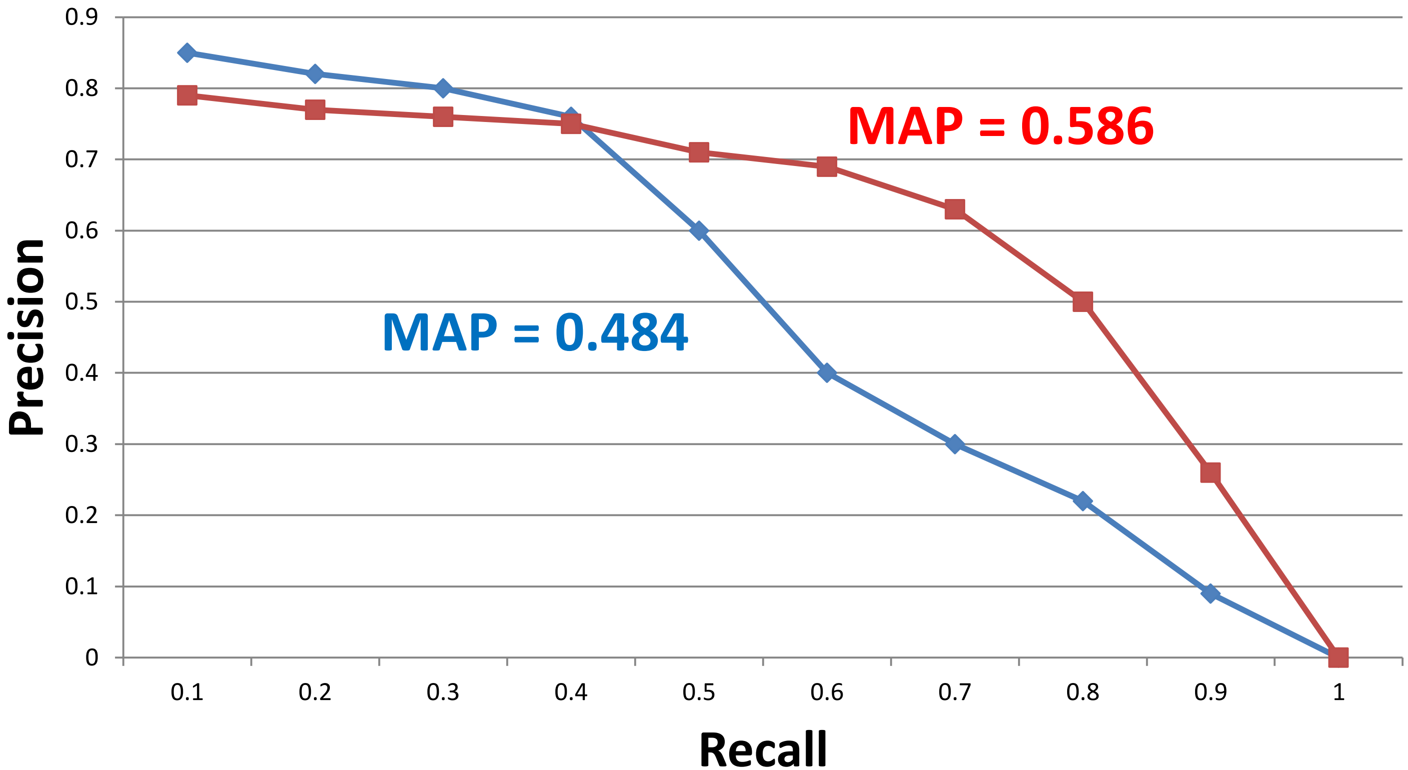
Recall rate = $\dfrac{A}{A+C}$

– recall rate may be difficult to evaluate, while precision rate is directly perceived by users

– recall-precision plot with varying thresholds

Performance Measures (2/2)

- MAP (mean average precision)
  - area under recall-precision curve
  - a performance measure frequently used for information retrieval



References

- **General or basic Spoken Content Retrieval**
  - http://www.superlectures.com/asru2011/lecture.php?lang=en&id=5
    Spoken Content Retrieval - Lattices and Beyond (Lin-shan Lee's talk at ASRU 2011)
  - Chelba, C., Hazen, T.J., Saraclar, M., "Retrieval and browsing of spoken content," Signal Processing Magazine, IEEE , vol.25, no.3, pp.39-49, May 2008
  - Martha Larson and Gareth J. F. Jones (2012) " Spoken Content Retrieval: A Survey of Techniques and Technologies ", Foundations and Trends in Information Retrieval: Vol. 5: No 4-5, pp 235-422
  - "An Introduction to Voice Search", Signal Processing Magazine, IEEE, Vol. 25, 2008
- **Text-based Information Retrieval**
  - http://nlp.stanford.edu/IR-book/
    Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze, Introduction to Information Retrieval, Cambridge University Press. 2008.

Vector Space Model

- **Vector Representations of query *Q* and document *d***
  - for each type *j* of indexing feature (e.g. syllable, word, etc.) a vector is generated
  - each component in this vector is the weighted statistics z_{jt} of a specific indexing term *t* (e.g. syllable s_i)

$$z_{jt} = \underbrace{\left(1 + \ln[c_t]\right)}_{\text{Term Frequency (TF)}} \cdot \underbrace{\ln\left(N/N_t\right)}_{\text{Inverse Document Frequency (IDF)}}$$

c_t: frequency counts for the indexing term t present in the query q or document d (for text), or sum of normalized recognition scores or confidence measures for the indexing term t (for speech)

N: total number of documents in the database

N_t: total number of documents in the database which include the indexing term t

IDF: the significance (or importance) or indexing power for the indexing term t

- **The Overall Relevance Score is the Weighted Sum of the Relevance Scores for all Types of Indexing Features**

$$R_j(\vec{Q}_j, \vec{d}_j) = \left(\vec{Q}_j \bullet \vec{d}_j\right) \Big/ \left(\left\|\vec{Q}_j\right\| \cdot \left\|\vec{d}_j\right\|\right)$$

$\vec{q}_j, \vec{d}_j$: vector representations for query q and document d with type j of indexing feature

$$S(Q, d) = \sum_j w_j \cdot R_j(\vec{Q}_j, \vec{d}_j)$$

w_j : weighting coefficients

Vector Space Model

mono-syllable

$$\begin{array}{c} s_1 \\ s_2 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ s_N \end{array} \begin{bmatrix} \vdots \\ 0 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ 0 \\ \vdots \\ \vdots \\ \vdots \end{bmatrix}$$

$(j = 1)$

bi-syllable

$$\begin{array}{c} s_1 s_2 \\ s_1 s_3 \\ s_1 s_4 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ s_N s_N \\ \end{array} \begin{bmatrix} \vdots \\ \vdots \\ \vdots \\ \vdots \\ 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \\ \vdots \end{bmatrix}$$

$(j = 2)$

character

$$\begin{array}{c} c_1 \\ c_2 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ c_M \end{array} \begin{bmatrix} \\ \\ \\ \\ \\ \\ \\ \\ \end{bmatrix}$$

$(j = 3)$

word

$$\begin{array}{c} w_1 \\ w_2 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ w_L \end{array} \begin{bmatrix} \\ \\ \\ \\ \\ \\ \\ \\ \end{bmatrix}$$

$(j = 4)$

<u>賽</u> <u>德</u> <u>克</u>•<u>巴</u> <u>萊</u>

ㄙ ㄉ ㄎ ㄅ ㄌ
ㄞ ㄜ ㄜ ㄚ ㄞ

Difficulties in Speech-based Information Retrieval for Chinese Language

- **Even for Text-based Information Retrieval, Flexible Wording Structure Makes it Difficult to Search by Comparing the Character Strings Alone**
  - name/title 李登輝→李前總統登輝，李前主席登輝(President T.H Lee)
  - arbitrary abbreviation 北二高→北部第二高速公路(Second Northern Freeway)
    華航→中華航空公司(China Airline)
  - similar phrases 中華文化→中國文化(Chinese culture)
  - translated terms 巴塞隆那→巴瑟隆納(Barcelona)
- **Word Segmentation Ambiguity Even for Text-based Information Retrieval**
  - 腦科(human brain studies) →電腦科學(computer science)
  - 土地公(God of earth) →土地公有政策(policy of public sharing of the land)
- **Uncertainties in Speech Recognition**
  - errors (deletion, substitution, insertion)
  - out of vocabulary (OOV) words, etc.
  - very often the key phrases for retrieval are OOV

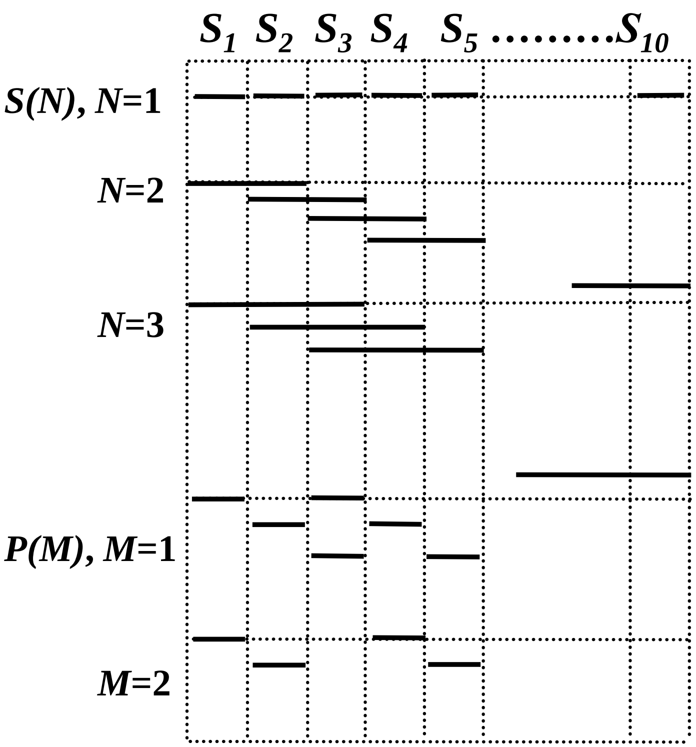
Syllable-Level Indexing Features for Chinese Language

- **A Whole Class of Syllable-Level Indexing Features for Better Discrimination**
  – Overlapping syllable segments with length *N*

Syllable Segments	*Examples*
S(N), N=1	*(s_1) (s_2) …(s_{10})*
S(N), N=2	*(s_1 s_2) (s_2 s_3)…(s_9 s_{10})*
S(N), N=3	*(s_1 s_2 s_3) (s_2 s_3 s_4)…(s_8 s_9 s_{10})*
S(N), N=4	*(s_1 s_2 s_3 s_4) (s_2 s_3 s_4 s_5)…(s_7 s_8 s_9 s_{10})*
S(N), N=5	*(s_1 s_2 s_3 s_4 s_5) (s_2 s_3 s_4 s_5 s_6)…(s_6 s_7 s_8 s_9 s_{10})*

  – Syllable pairs separated by *M* syllables

Syllable Pair Separated by M syllables	*Examples*
P(M), M=1	*(s_1 s_3) (s_2 s_4)……(s_8 s_{10})*
P(M), M=2	*(s_1 s_4) (s_2 s_5)……(s_7 s_{10})*
P(M), M=3	*(s_1 s_5) (s_2 s_6)……(s_6 s_{10})*
P(M), M=4	*(s_1 s_6) (s_2 s_7)……(s_5 s_{10})*



- **Character- or Word-Level Features can be Similarly Defined**

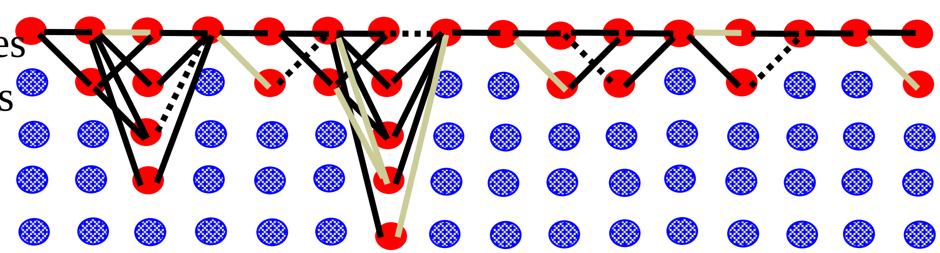
Syllable-Level Statistical Features

- **Single Syllables**
  - all words are composed by syllables, thus partially handle the OOV problem
  - very often relevant words have some syllables in common
  - each syllable usually shared by more than one characters with different meanings, thus causing ambiguity
- **Overlapping Syllable Segments with Length *N***
  - capturing the information of polysyllabic words or phrases with flexible wording structures
  - majority of Chinese words are bi-syllabic
  - not too many polysyllabic words share the same pronunciation
- **Syllable Pairs Separated by *M* Syllables**
  - tackling the problems arising from the flexible wording structure, abbreviations, and deletion, insertion, substitution errors in speech recognition

Improved Syllable-level Indexing Features

- **Syllable-aligned Lattices and syllable-level utterance verification**
  - Including multiple syllable hypothesis to construct syllable-aligned lattices for both query and documents
  - Generating multiple syllable-level indexing features from syllable lattices
  - filtering out indexing terms with lower acoustic confidence scores
- **Infrequent term deletion (ITD)**
  - Syllable-level statistics trained with text corpus used to prune infrequent indexing terms
- **Stop terms (ST)**
  - Indexing terms with the lowest IDF scores are taken as the stop terms

syllables with higher acoustic confidence scores
syllables with lower acoustic confidence scores
syllable pairs S(N), N=2 pruned by ITD
syllable pairs S(N), N=2 pruned by ST



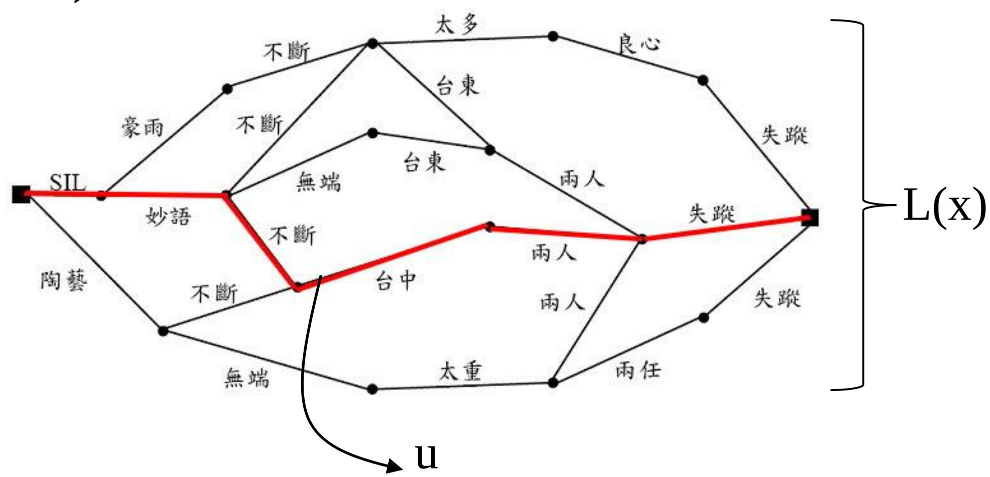
Expected Term Frequencies

- **E(t,x): expected term frequency for term t in the lattice of an utterance x**

$$E(t,x) = \sum_{u \in L(x)} N(t,u) P(u \mid x)$$

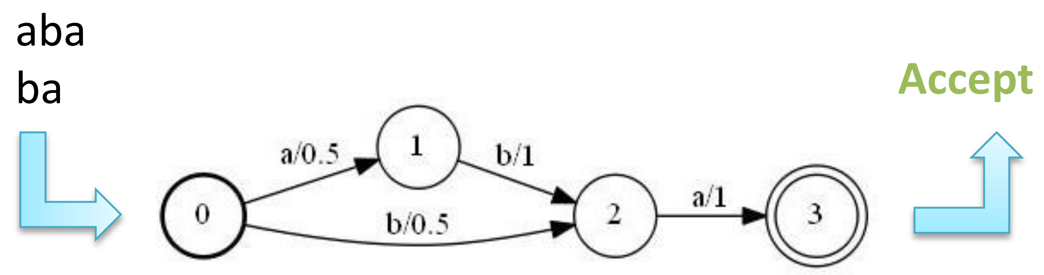
  - u: a word sequence (path) in the lattice of an utterance x
  - P(u|x): posterior probability of the word sequence u given x
  - N(t,u): the occurrence count of term t in word sequence u
  - L(x): all the word sequences (paths) in the lattice of an utterance x

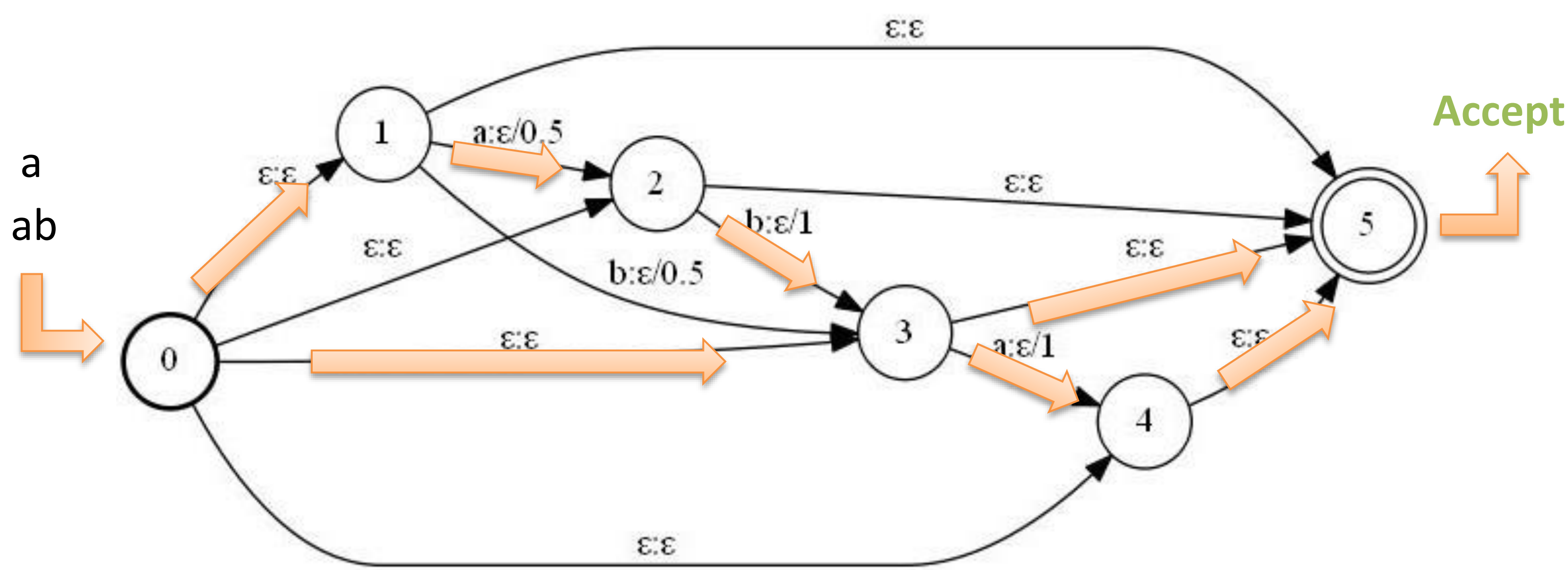
lattice of utterance x



WFST for Retrieval (1/4)

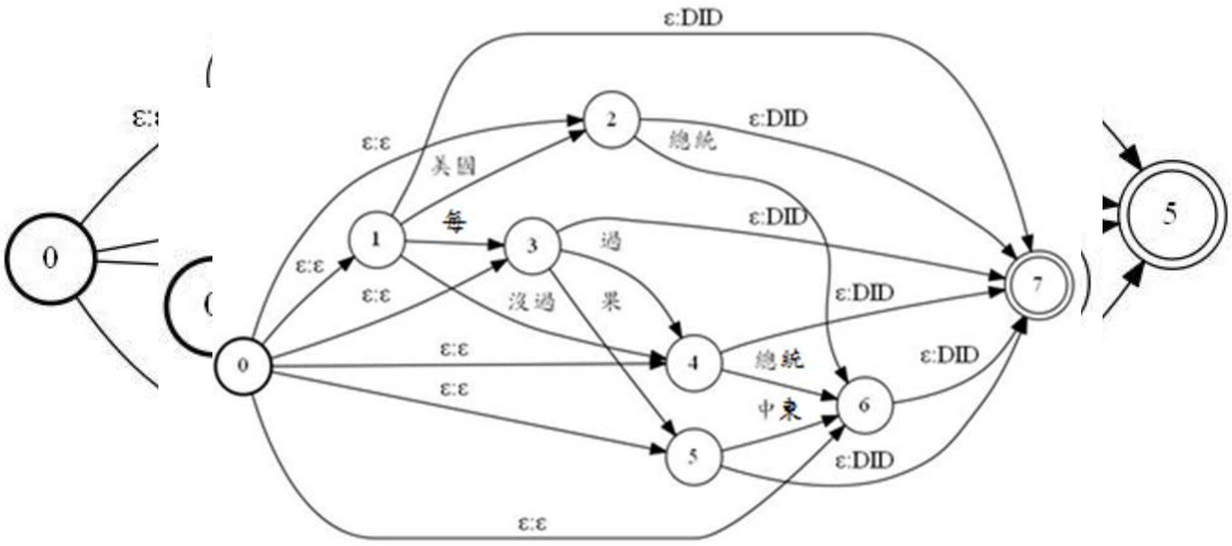
- **Factor Automata**
  - The finite state machines accepting all substrings of the original machine
  - retrieval is to have all substrings considered





WFST for Retrieval (2/4)

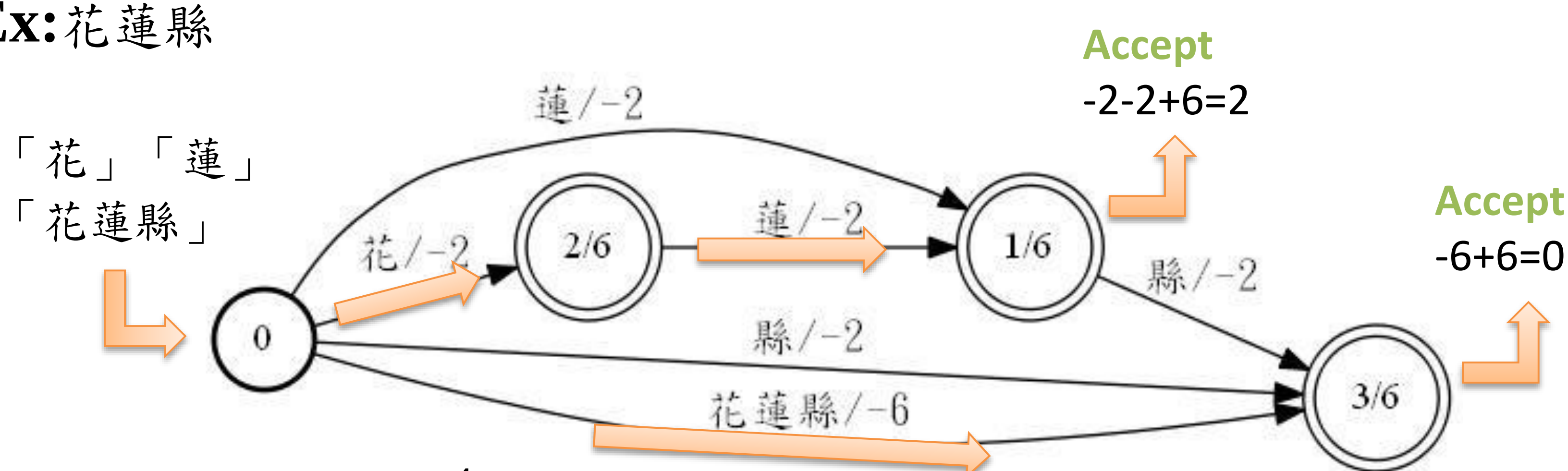
- **The index transducer of text document**
  - Every substring of the document is transduced to the corresponding document ID (e.g., 3014)
- **For spoken documents, the index transducers are generated from lattices directly**
- **The index transducer of the whole corpus**
  - Union of all transducers of all utterances



WFST for Retrieval (3/4)

- **Query Transducer**
  - Split the query string into words, characters, syllables, etc.
  - Generate the query transducer
  - Factorize the automaton
  - Distribute weights over different transitions

- **Ex:** 花蓮縣

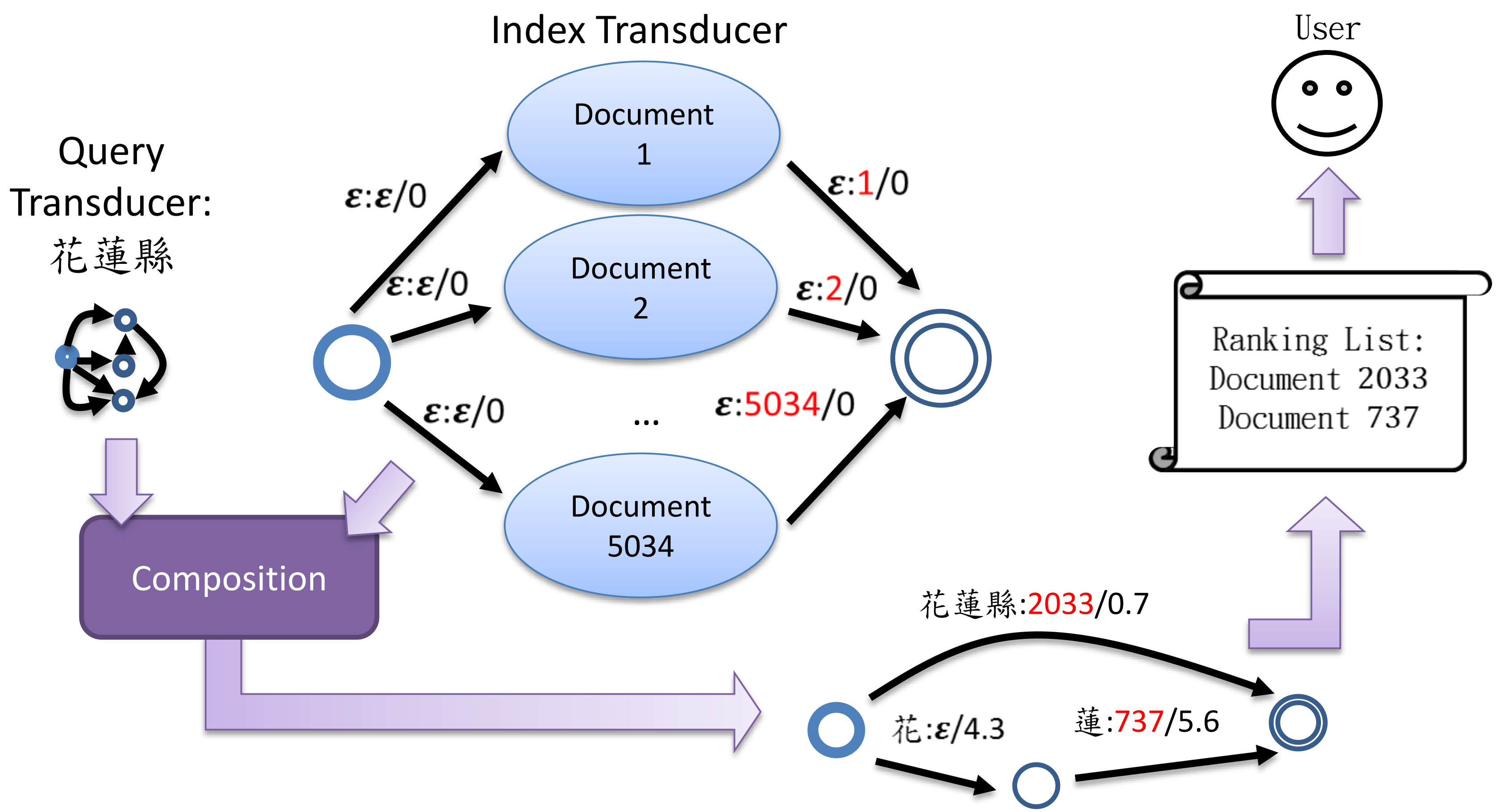


$\{蓮\} \quad : 4 = -\log(10^{-4})$

$\{花蓮\}: 2 = -\log(10^{-2})$

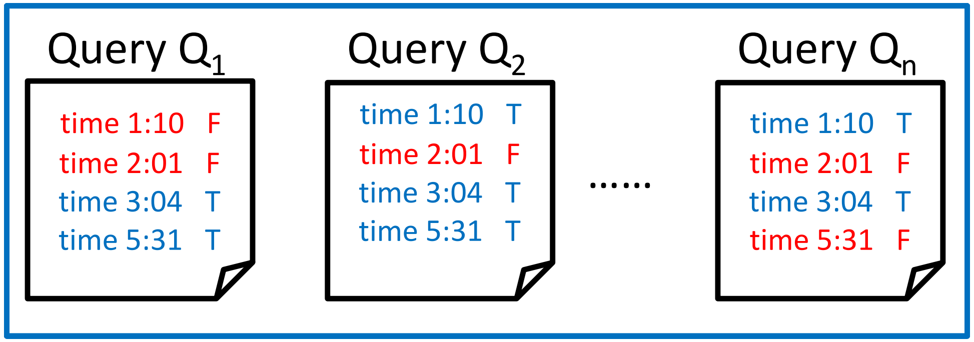
$\{花蓮縣\}: 0 = -\log(1.0)$

WFST for Retrieval (4/4)



Improved Retrieval by Training

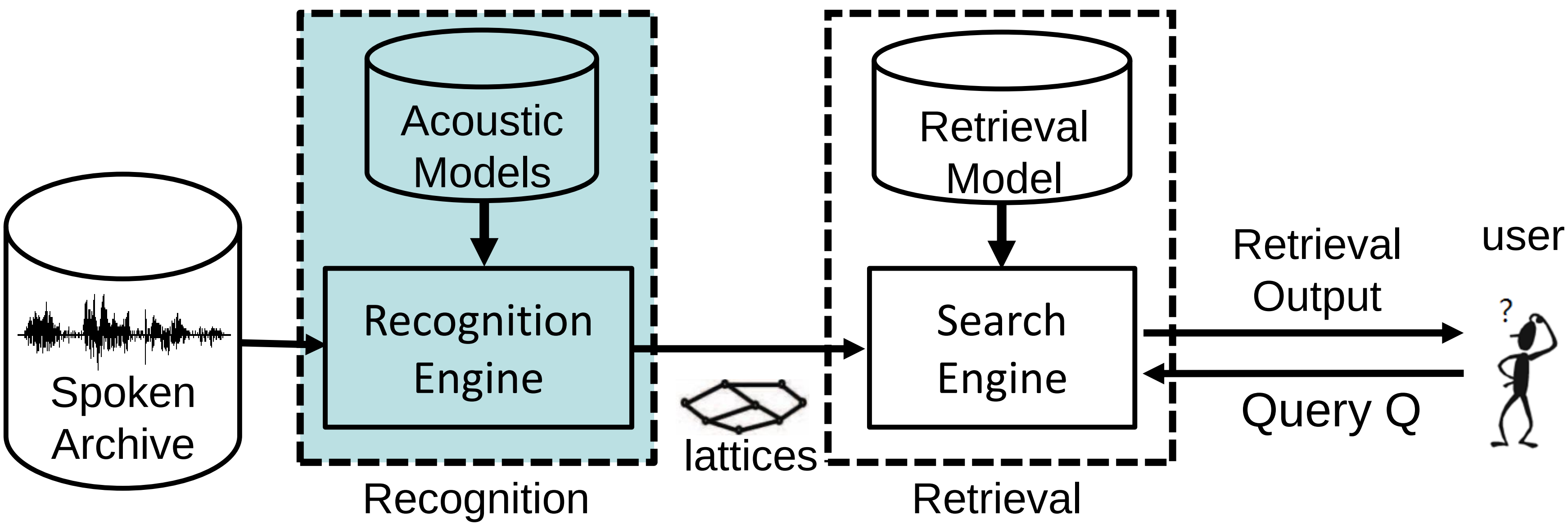
- Improve the retrieval with some training data
  - Training data: a set of queries and associated relevant/irrelevant utterances



  - Can be collected from user data
    - e.g. click-through data
- Improve text-based search engine
  - e.g. learn weights for different clues (such as different recognizers, different subword units …)
- Optimize the recognition models for retrieval performance
  - Considering retrieval and recognition processes as a whole
  - Re-estimate HMM parameters

HMM Parameter Re-estimation

- **Retrieval considered on top of recognition output in the past**
  - recognition and retrieval as two cascaded stages
  - retrieval performance relying on recognition accuracy
- **Considering retrieval and recognition processes as a whole**
  - acoustic models re-estimated by optimizing retrieval performance
  - acoustic models better matched to each respective data set



HMM Parameter Re-estimation

- **Objective Function for re-estimating HMM**

$$\hat{\lambda} = \arg\max_{\lambda} \sum_{Q \in Q_{train}} \sum_{x_t, x_f} \left[S(Q, x_t \mid \lambda) - S(Q, x_f \mid \lambda) \right]$$

λ: set of HMM parameters, $\hat{\lambda}$: re-estimated parameters for retrieval

Q_{train}: training query set

x_t, x_f: positive/negative examples for query Q

$S(Q,x|\lambda)$: relevance score of utterance x given query Q and model parameters set λ
(Since S(Q,x) is obtained from lattice, it depends on HMM parameters λ.)

Find new HMM parameters for recognition
➡ such that the relevance scores of positive and negative examples are better separated.

References

- **WFST for Retrieval**

  - Cyril Allauzen, Mehryar Mohri, and Murat Saraclar, “General indexation of weighted automata: application to spoken utterance retrieval,” in Proceedings of the Workshop on Interdisciplinary Approaches to Speech Indexing and Retrieval at HLT-NAACL, Stroudsburg, PA, USA, 2004, SpeechIR ’04, pp. 33–40, Association for Computational Linguistics.
  - D. Can and M. Saraclar, “Lattice indexing for spoken term detection,” IEEE Transactions on Audio, Speech, and Language Processing, vol. 19, no. 8, pp. 2338–2347, 2011.

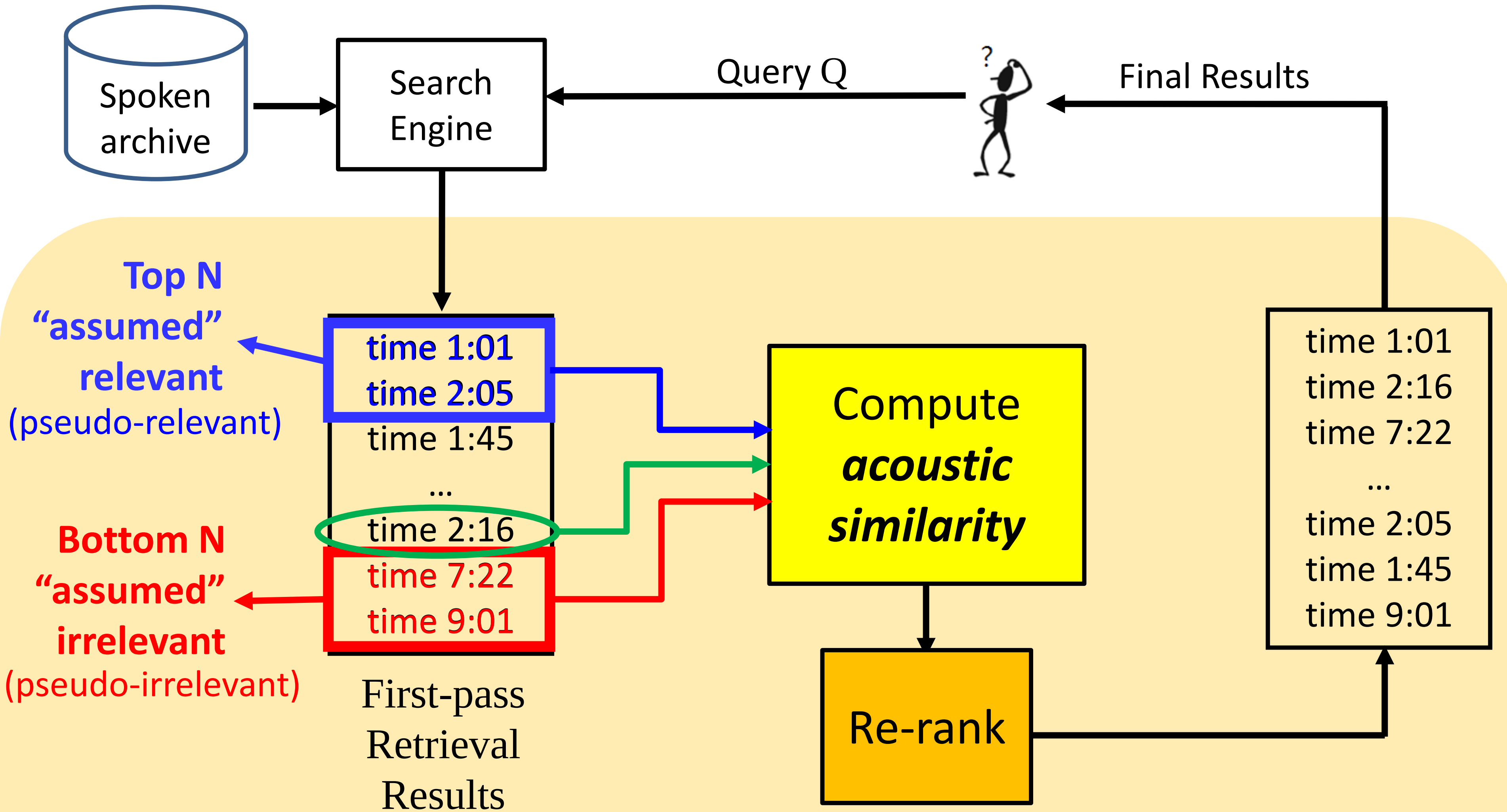
References

- **Spoken Content in Mandarin Chinese**
  - “Discriminating Capabilities of Syllable-based Features and Approaches of Utilizing Them for Voice Retrieval of Speech Information in Mandarin Chinese”, IEEE Transactions on Speech and Audio Processing, Vol.10, No.5, July 2002, pp.303-314.
- **Training Retrieval Systems**
  - Click-through data
    - Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '02)
  - Improve text-based search engine
    - “Improved Lattice-based Spoken Document Retrieval by Directly Learning from the evaluation Measures”, IEEE International Conference on Acoustics, Speech and Signal Processing, 2009
  - Re-estimate HMM parameters
    - "Integrating Recognition and Retrieval With Relevance Feedback for Spoken Term Detection," *Audio, Speech, and Language Processing, IEEE Transactions on* , vol.20, no.7, pp.2095-2110, Sept. 2012

Pseudo-relevance Feedback (PRF) (1/3)

- Collecting training data can be expensive
- Pseudo-relevance feedback (PRF):
  - Generate training data automatically
  - Procedure:
    - Generate first-pass retrieval results
    - assume the top N objects on the first-pass retrieval results are relevant (pseudo relevant)
    - assume the bottom M objects on the first-pass retrieval results are irrelevant (pseudo irrelevant)
    - Re-ranking: scores of objects similar to the pseudo-relevant/irrelevant objects increased/decreased

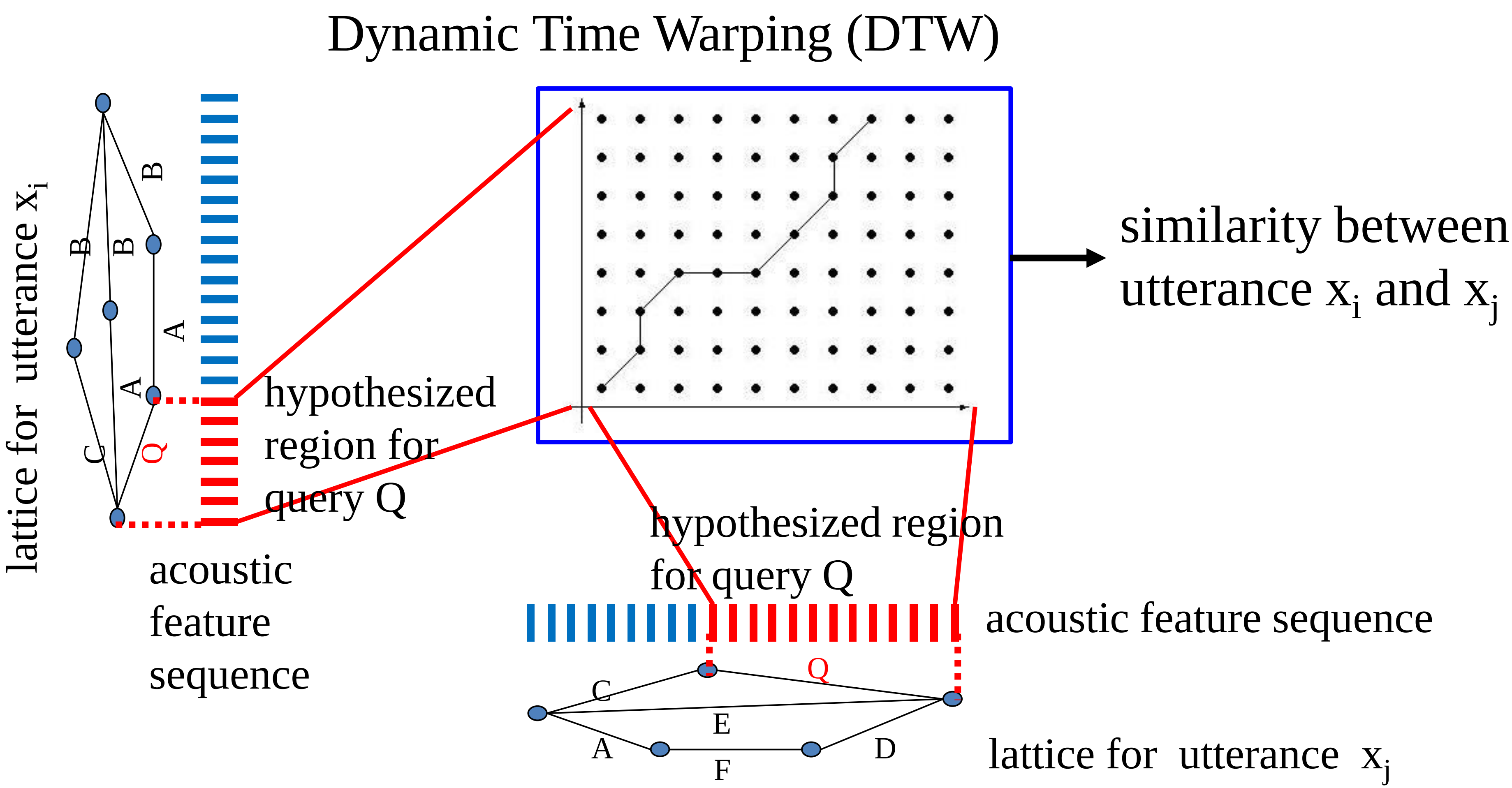
Pseudo-relevance Feedback (PRF) (2/3)



Re-rank: increase/decrease the score of utterances having higher **acoustic similarity** with pseudo-relevant/-irrelevant utterances

Pseudo-relevance Feedback (PRF) (3/3)

- **Acoustic similarity** between two utterances x_i and x_j

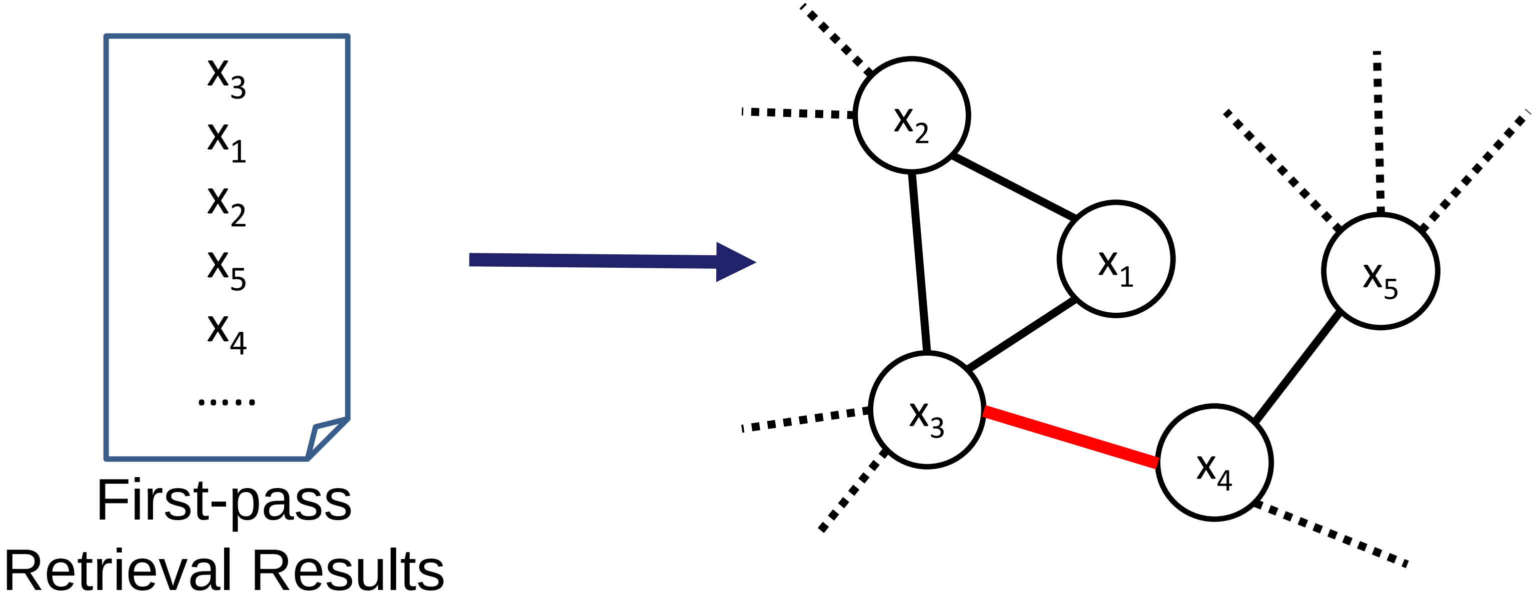


Improved PRF – Graph-based Approach (1/4)

- Graph-based approach
  - only the top N/bottom N utterances are taken as references in PRF
  - not necessarily reliable
  - considering the acoustic similarity structure of all utterances in the first-pass retrieval results globally using a graph

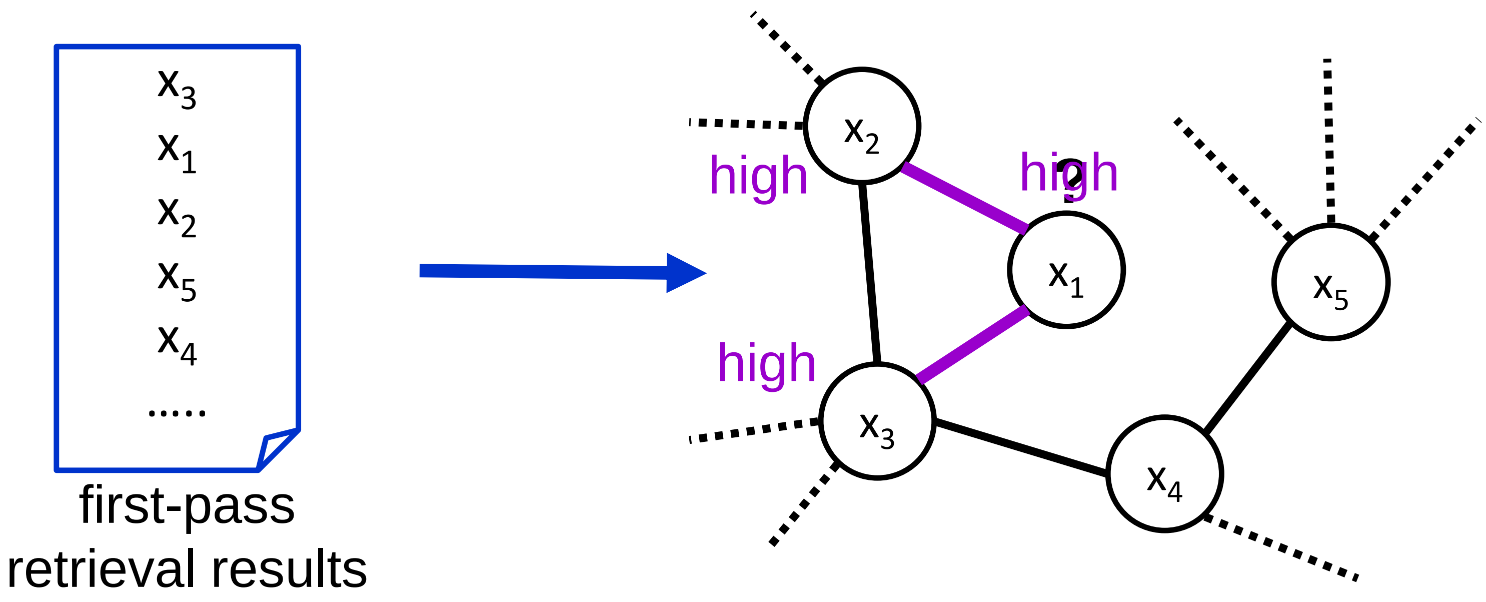
Improved PRF – Graph-based Approach (2/4)

- Construct a graph for all utterances in the first-pass retrieval results
  - nodes : utterances
  - edge weights: acoustic similarities between utterances



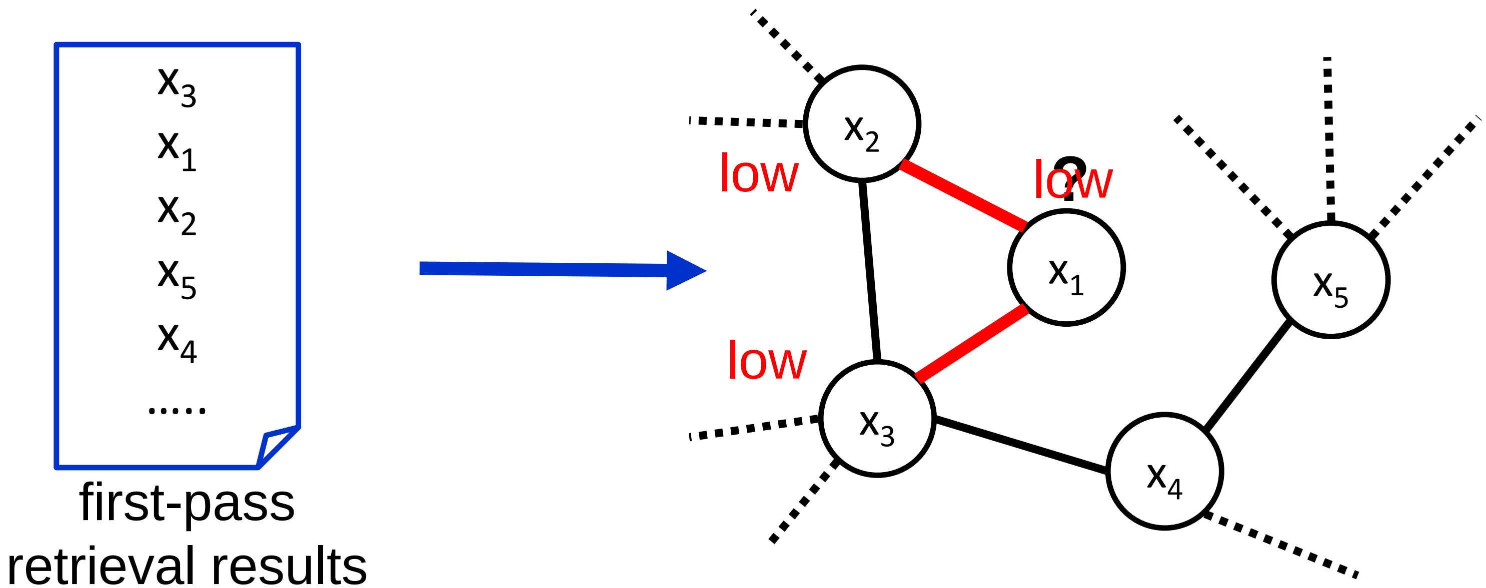
Improved PRF – Graph-based Approach (3/4)

- Utterances strongly connected to (similar to) utterances with high relevance scores should have relevance scores increased



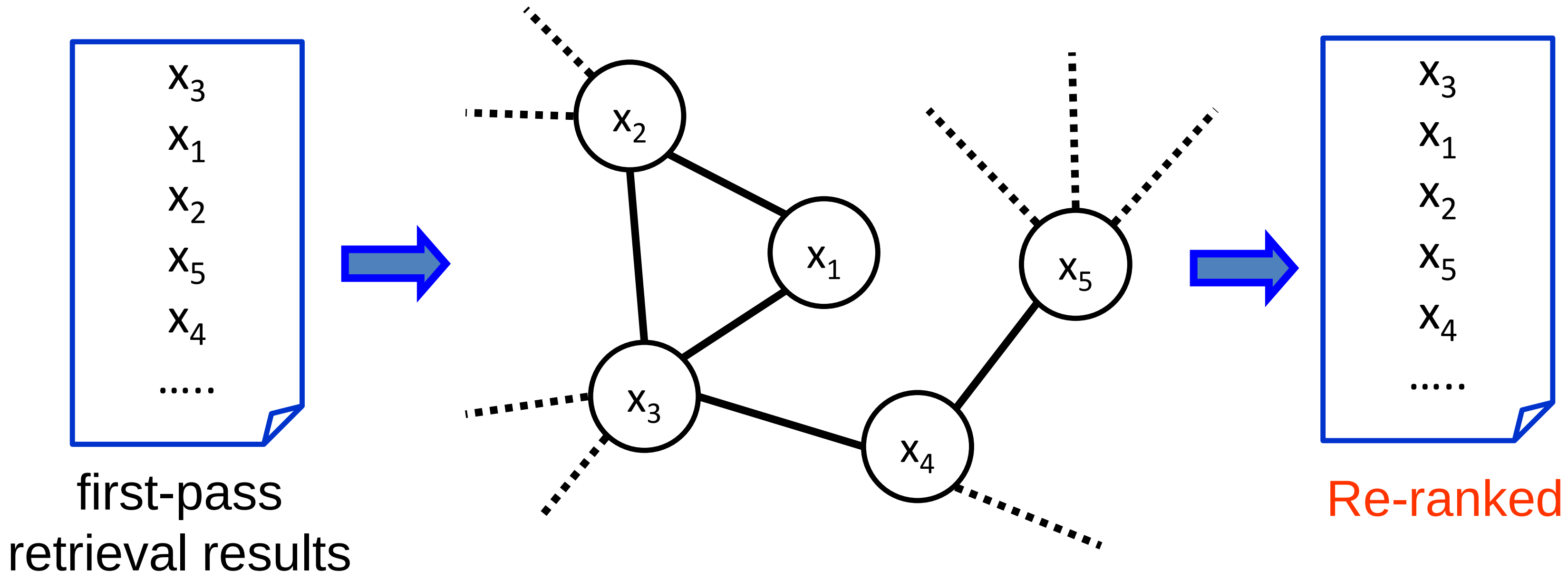
Improved PRF – Graph-based Approach (3/4)

- Utterances strongly connected to (similar to) utterances with low relevance scores should have relevance scores reduced



Improved PRF – Graph-based Approach (4/4)

- **Relevance scores propagate on the graph**
  - relevance scores smoothed among strongly connected nodes

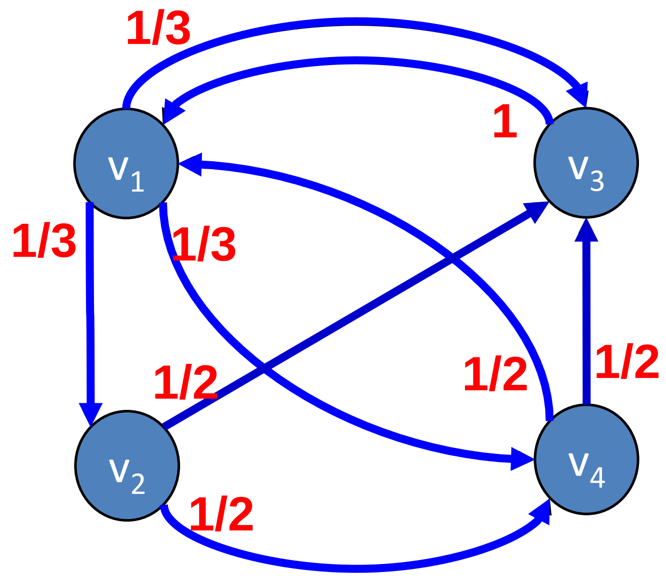


PageRank and Random Walk (1/2)

- **Object ranking by their relations**
  - Rank web pages for Google search
- **Basic Idea**
  - Objects having high connectivity to other high-score objects are popular (given higher scores)

$$
\begin{array}{cc}
 & \text{from} \\
P = & \begin{bmatrix} 0 & 0 & 1 & \frac{1}{2} \\ \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{2} & 0 & 0 \end{bmatrix} \text{to}
\end{array}
$$

Transition matrix



PageRank and Random Walk (2/2)

- The score of each object is related to the score of its neighbors and its prior score

- Final steady state

$$s_i = \alpha \sum_j p_{ji} s_j + (1-\alpha) v_i$$

final score

interpolation weight

Score propagation

Prior score

- In matrix form

$$\begin{aligned} \vec{s} &= \alpha P \vec{s} + (1-\alpha)\vec{v} \qquad , \vec{s} = [s_1, s_2, \cdots]^{\mathrm{T}} \quad , \vec{v} = [v_1, v_2, \cdots]^{\mathrm{T}} \\ &= \alpha P \vec{s} + (1-\alpha)\vec{v} e^{\mathrm{T}} \vec{s} \\ &= [\alpha P + (1-\alpha)\vec{v} e^{\mathrm{T}}]\vec{s} = P' \vec{s} \quad , e^{\mathrm{T}} = [1, 1, 1, \cdots, 1], e^{\mathrm{T}} \vec{s} = \textstyle\sum_i s_i = 1 \end{aligned}$$

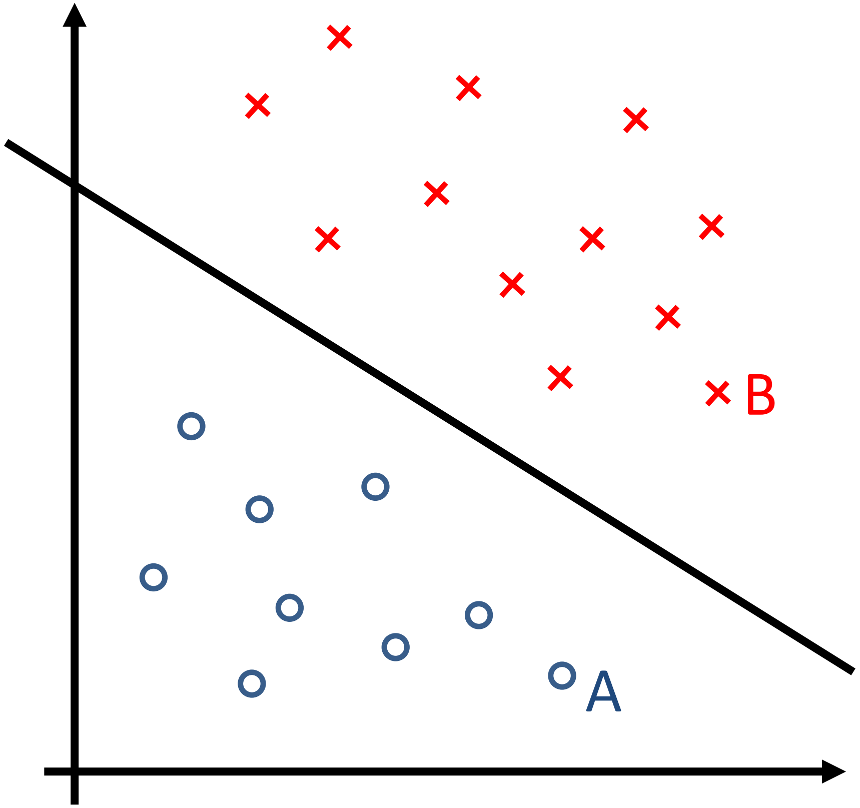
  - $\vec{s}$ is the solution to the eigenvalue problem

References

- **For Graph and Random walk**
  - Kurt Bryan[1], Tanya Leise, "The $25,000,000,000 eigenvector: the linear algebra behind google"
  - Amy. N. Langville, Carl.D. Meyer, "Deeper inside PageRank", Internet Mathematics, Vol. 1
  - "Improved Spoken Term Detection with Graph-Based Re-Ranking in Feature Space", in ICASSP 2011
  - "Open-Vocabulary Retrieval of Spoken Content with Shorter/Longer Queries Considering Word/Subword-based Acoustic Feature Similarity", Interspeech, 2012

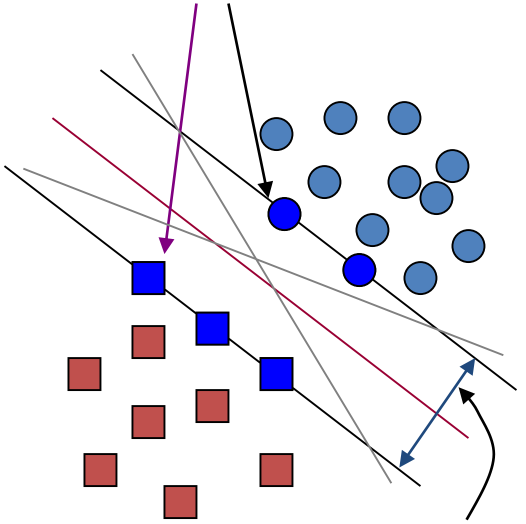
Support Vector Machine (SVM) (1/2)

- **Problem definition**
  - suppose there are two classes of objects (positive and negative)
  - goal: classify new objects given training examples
- **Represent each object as an N-dimensional feature vector**
  - o: positive example
  - x: negative example
- **Find a hyperplane separating positive and negative examples**
- **Classify new objects by this hyperplane**
  - point A is positive, point B is negative

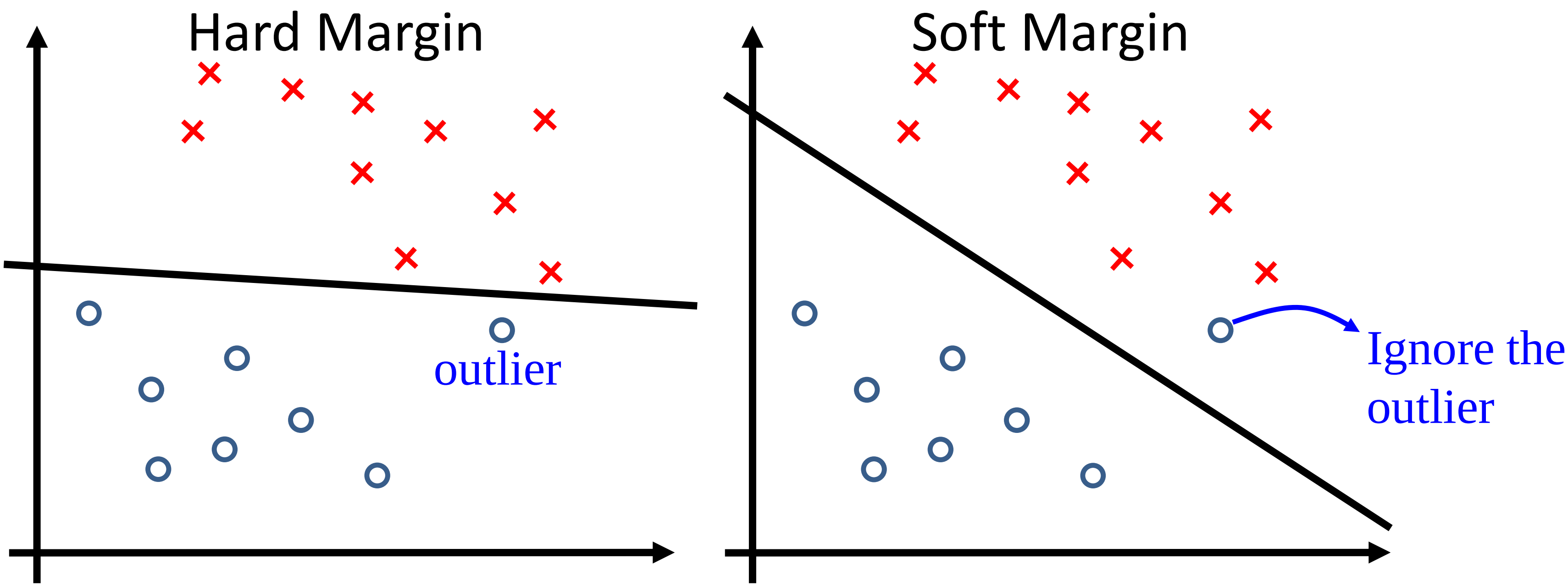


Support Vector Machine (SVM) (2/2)

- **Many hyperplanes can separate positive and negative examples**
- **Choose the one maximizing the "margin"**
  - margin: the minimum distance between the examples and the hyperplane
- **Some noise may change the feature vectors of the testing objects**
  - large margin may minimize the chance of misclassification



SVM – Soft Margin

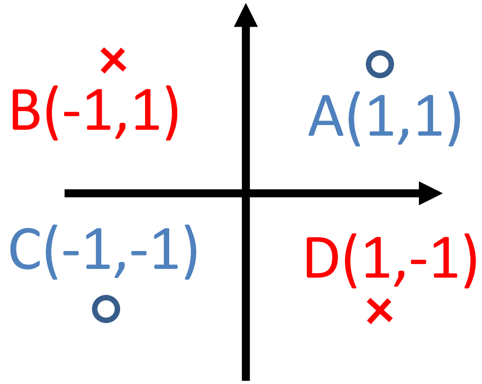


- **Hard Margin:**
  - If some training examples are outliers, separating all positive/negative examples may not be the best solution
- **Soft Margin:**
  - Tolerate some non-separable cases (outliers)

SVM – Feature Mapping

- **Original feature vectors (Non-separable)**

$$\begin{bmatrix} x \\ y \end{bmatrix}$$



- **Map original feature vectors onto a higher-dimensional space**

$$\begin{bmatrix} x^2 \\ y^2 \\ xy \end{bmatrix}$$

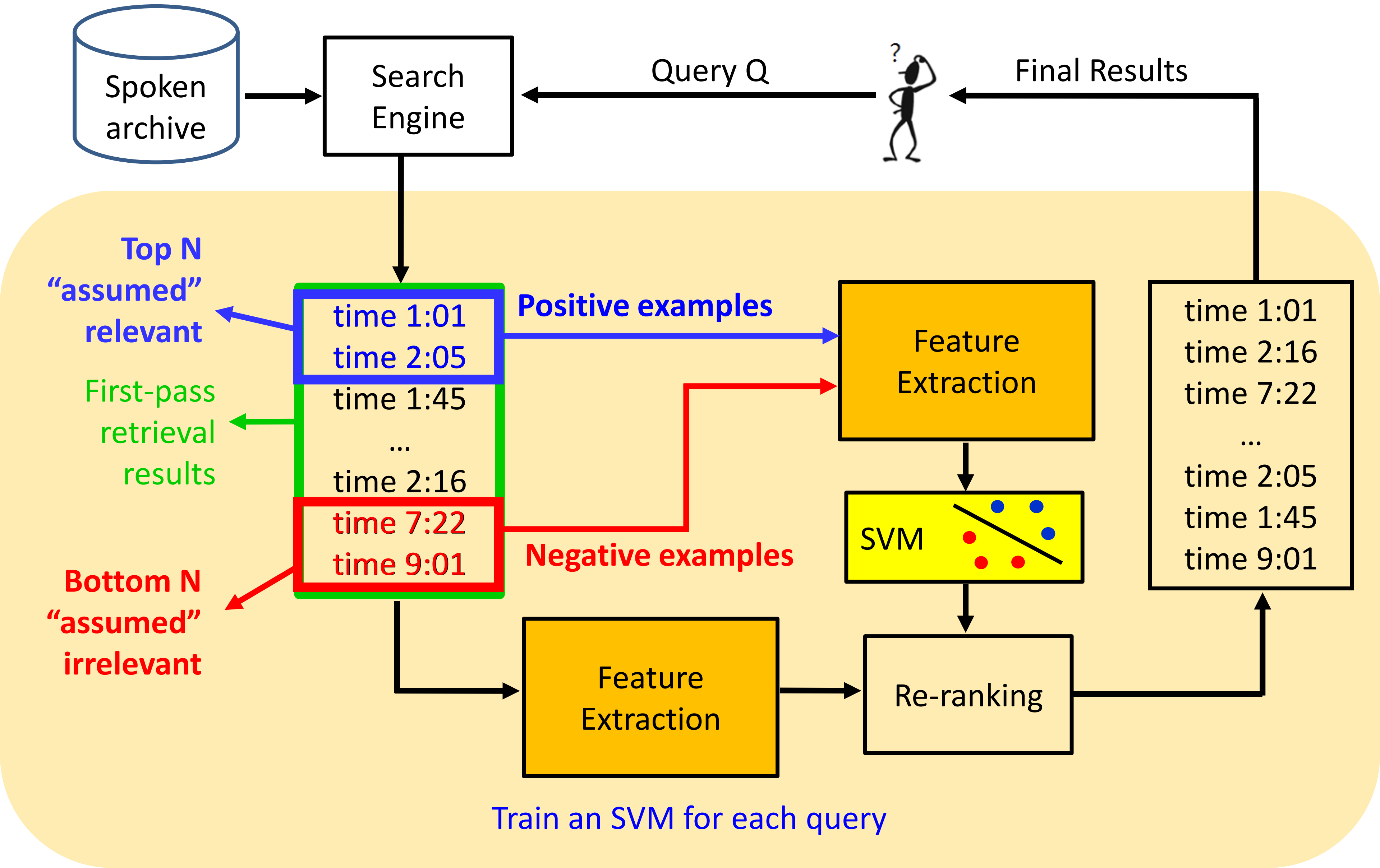
B(1,1,-1) A(1,1,1)

C(1,1,1) D(1,1,-1)

(Can be separated by hyperplane z=xy=0)

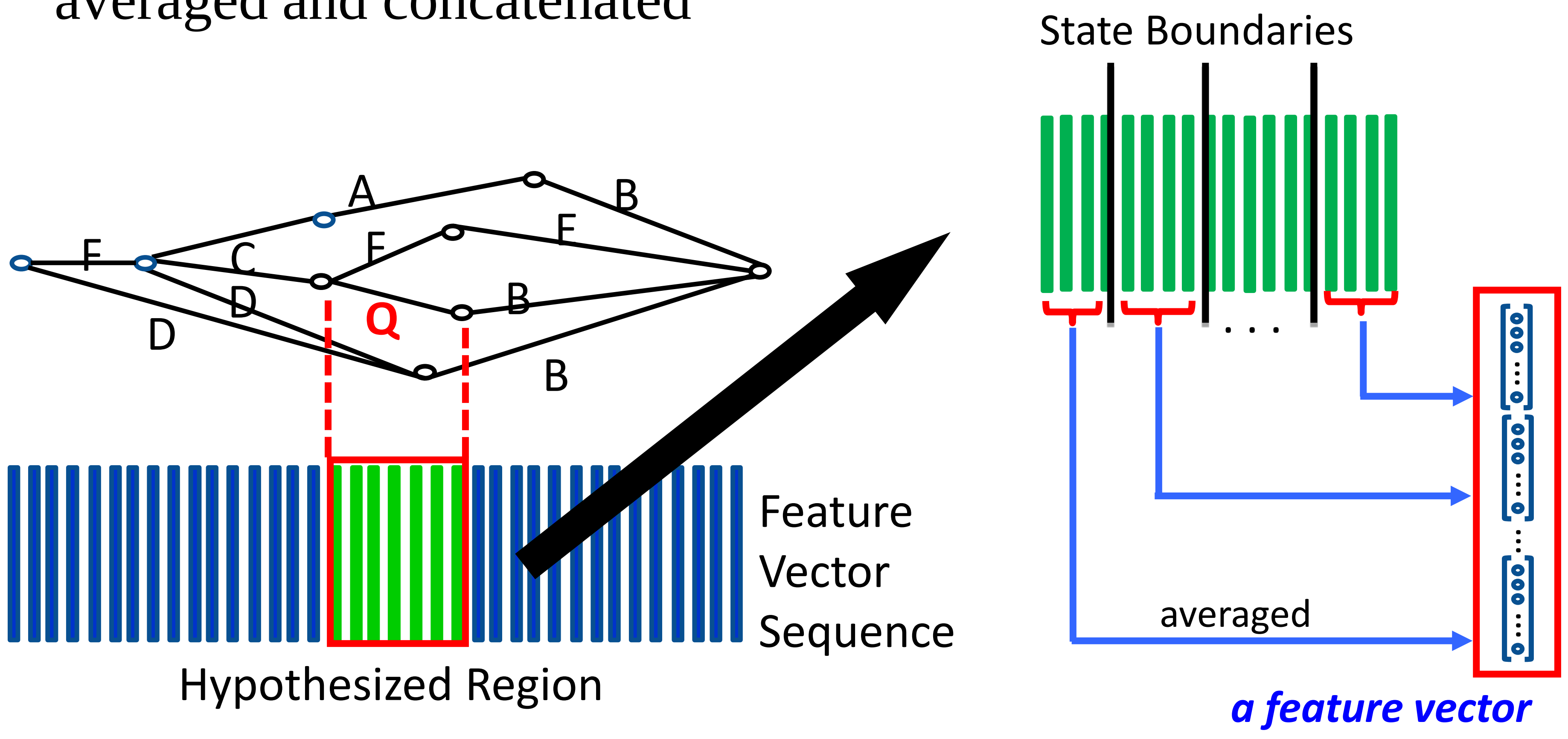
- **If positive and negative examples are not linearly separable in the original feature vector form, map their feature vectors onto a higher-dimensional space where they may become separable**

Improved PRF – SVM(1/3)



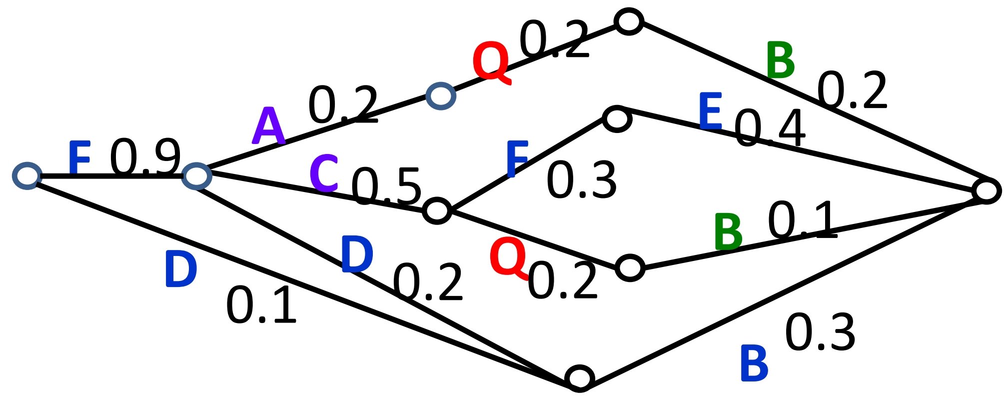
Improved PRF – SVM (2/3)

- Representing each utterance by its hypothesized region segmented by HMM states, with feature vectors in each state averaged and concatenated



Improved PRF – SVM (3/3)

- Context consistency
  - the same term usually have similar context; while quite different context usually implies the terms are different
- Feature Extraction



V - dimensional vector
(*V* : lexicon size)

A	B	C	D	…	Q
0.2	0.0	**0.5**	0.0	…	0.0

Immediate left context

A	B	C	D	…	Q
0.0	**0.3**	0.0	0.0	…	0.0

Immediate right context

A	B	C	D	…	Q
0.2	0.6	0.5	0.3	…	0.4

whole segment

Concatenated into a 3V - dimensional feature vector

References

- **SVM**
  - http://cs229.stanford.edu/materials.html
    (Lecture notes 3)
  - "A Tutorial on Support Vector Machines for Pattern Recognition," Data Mining and Knowledge Discovery, vol. 2, no. 2, pp. 121-167, 1998.
  - Bishop, C.M.
    <http://library.wur.nl/WebQuery/clc?achternaam==Bishop>, "Pattern recognition and machine learning." Chapter 7.
  - Nello Cristianini and John Shawe-Taylor. "An Introduction to Support Vector Machines: And Other Kernel-Based Learning Methods."
- **SVM Toolkit**
  - http://www.csie.ntu.edu.tw/~cjlin/libsvm/
    LibSVM
  - http://svmlight.joachims.org/
    SVMlight

References

- **Pseudo-relevance Feedback (PRF)**
  - "Improved Spoken Term Detection by Feature Space Pseudo-Relevance Feedback", Annual Conference of the International Speech Communication Association, 2010
- **SVM-based Reranking**
  - "Improved Spoken Term Detection Using Support Vector Machines Based on Lattice Context Consistency", International Conference on Acoustics, Speech and Signal Processing, Prague, Czech Republic, May 2011, pp. 5648-5651.
  - "Improved Spoken Term Detection Using Support Vector Machines with Acoustic and Context Features From Pseudo-Relevance Feedback", IEEE Workshop on Automatic Speech Recognition and Understanding, Hawaii, Dec 2011, pp. 383-388.
  - "Enhanced Spoken Term Detection Using Support Vector Machines and Weighted Pseudo Examples", IEEE Transactions on Audio, Speech and Language Processing , Vol. 21, No. 6, Jun 2013, pp. 1272-1284

Language Modeling Retrieval Approach (Text or Speech)

- Both query Q and spoken document d are represented as language models θ_Q and θ_d (consider unigram only below, may be smoothed (or interpolated) by a background model θ_b)
- Given query Q, rank spoken documents d according to $S_{LM}(Q,d)$

$$S_{LM}(Q,d) = -KL(\theta_Q \mid \theta_d)$$

  - Inverse of KL divergence (KL distance) between θ_Q and θ_d
  - The documents with document models θ_d similar to query model θ_Q are more likely to be relevant

Query model

$$P(t \mid \theta_Q) = \frac{N(t,Q)}{\sum_{t'} N(t',Q)}$$

$N(t, Q)$: Occurrence count or expected term frequency for term t in query Q

Document model

$$P(t \mid \theta_d) = \frac{N(t,d)}{\sum_{t'} N(t',d)}$$

$N(t, d)$: Occurrence count or expected term frequency for term t in document d

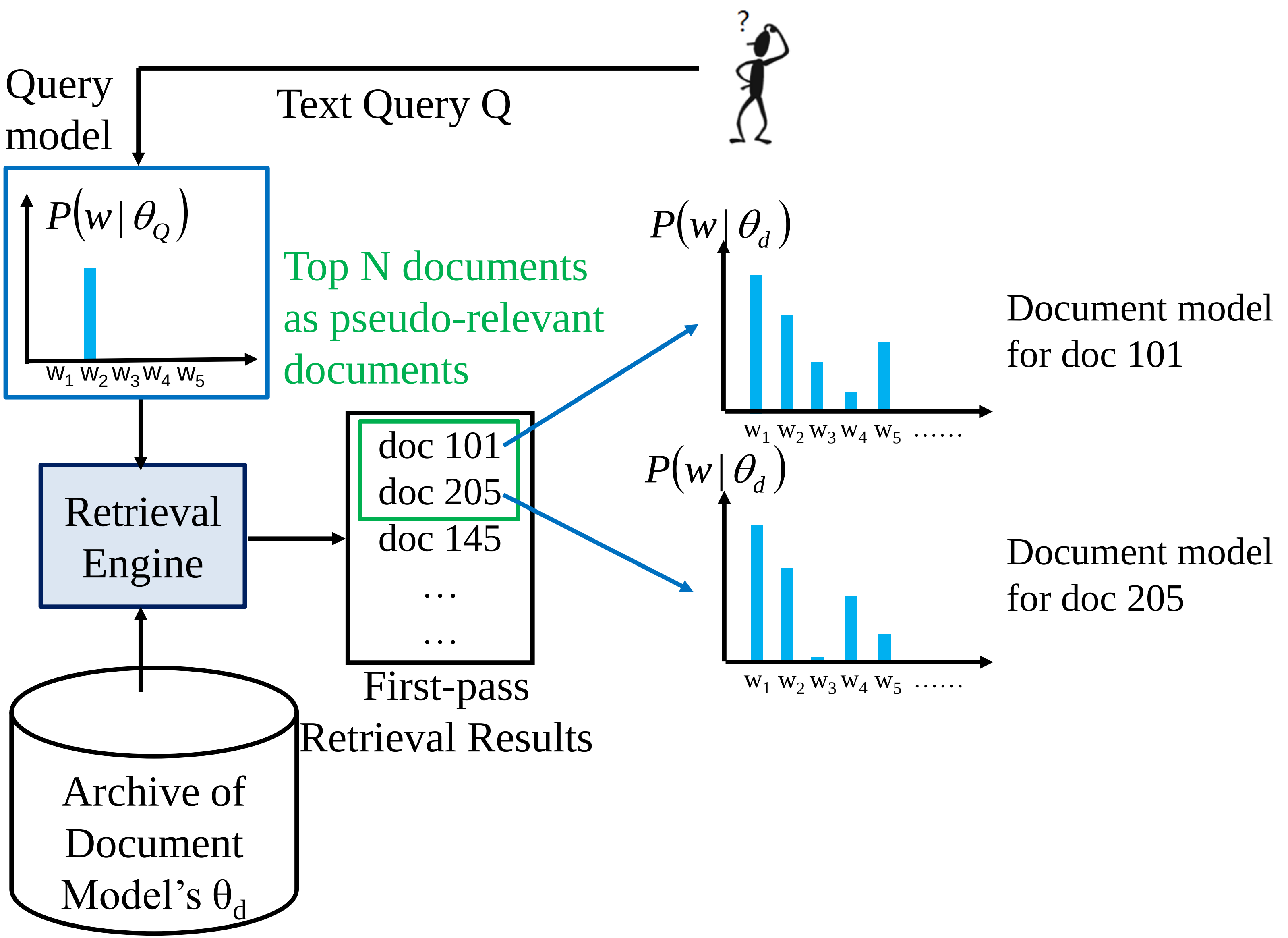
$$N(t,d) = \sum_{x \in d} E(t,x)$$

E(t, x): Expected term frequency for term t in the lattice of utterance x (for speech)

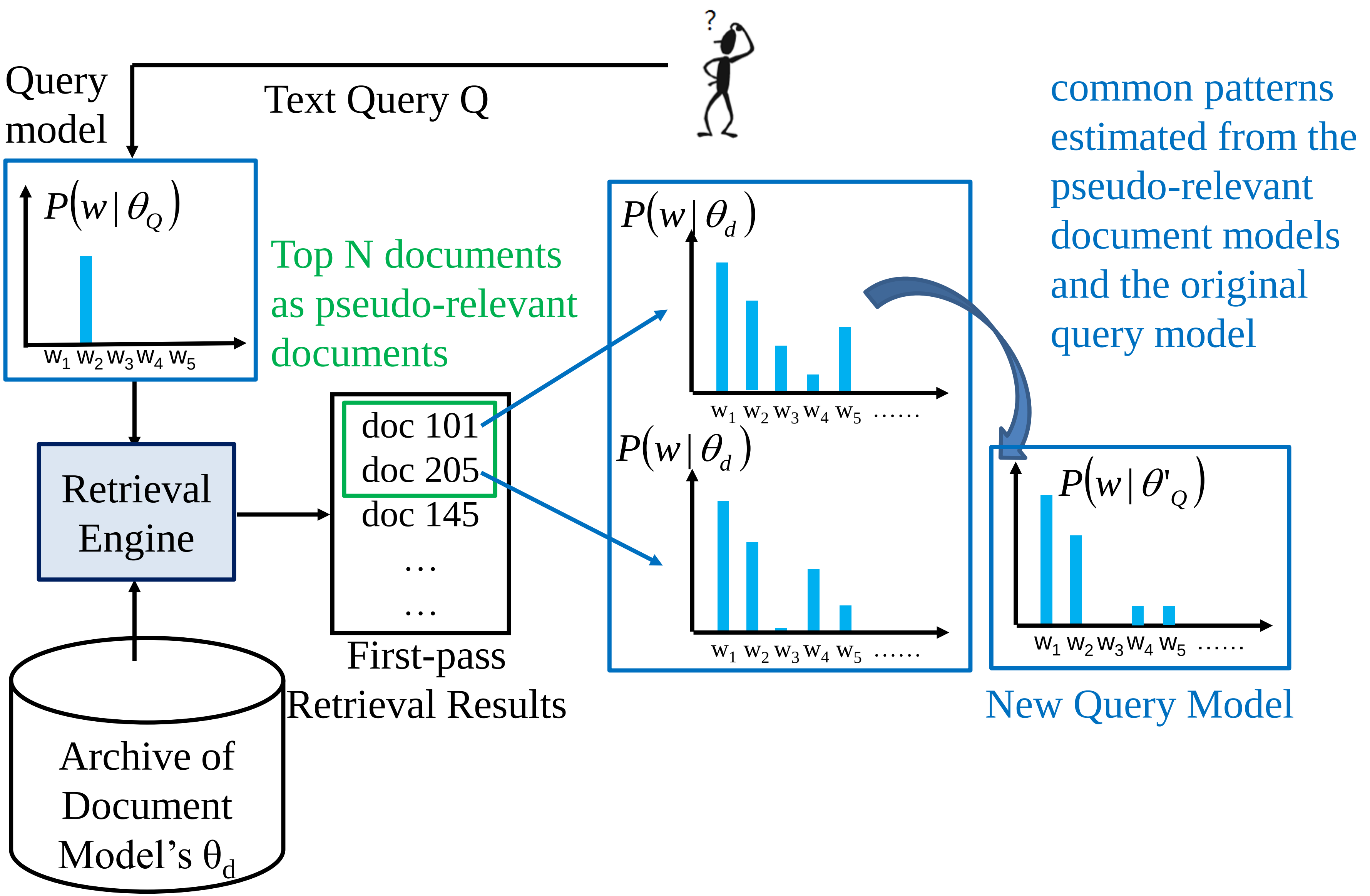
Semantic Retrieval by Query Expansion

- **Concept matching rather than Literal matching**
- **Returning utterances/documents semantically related to the query (e.g. Obama)**
  - not necessarily containing the query (e.g. including US and White House, but not Obama)
- **Expand the query (Obama) with semantically related terms (US and White House)**
- **Query expansion with language modeling retrieval approach**
  - Realized by PRF
  - Find common term distribution in pseudo-relevant documents and use it to construct a new query for 2nd-phase retrieval

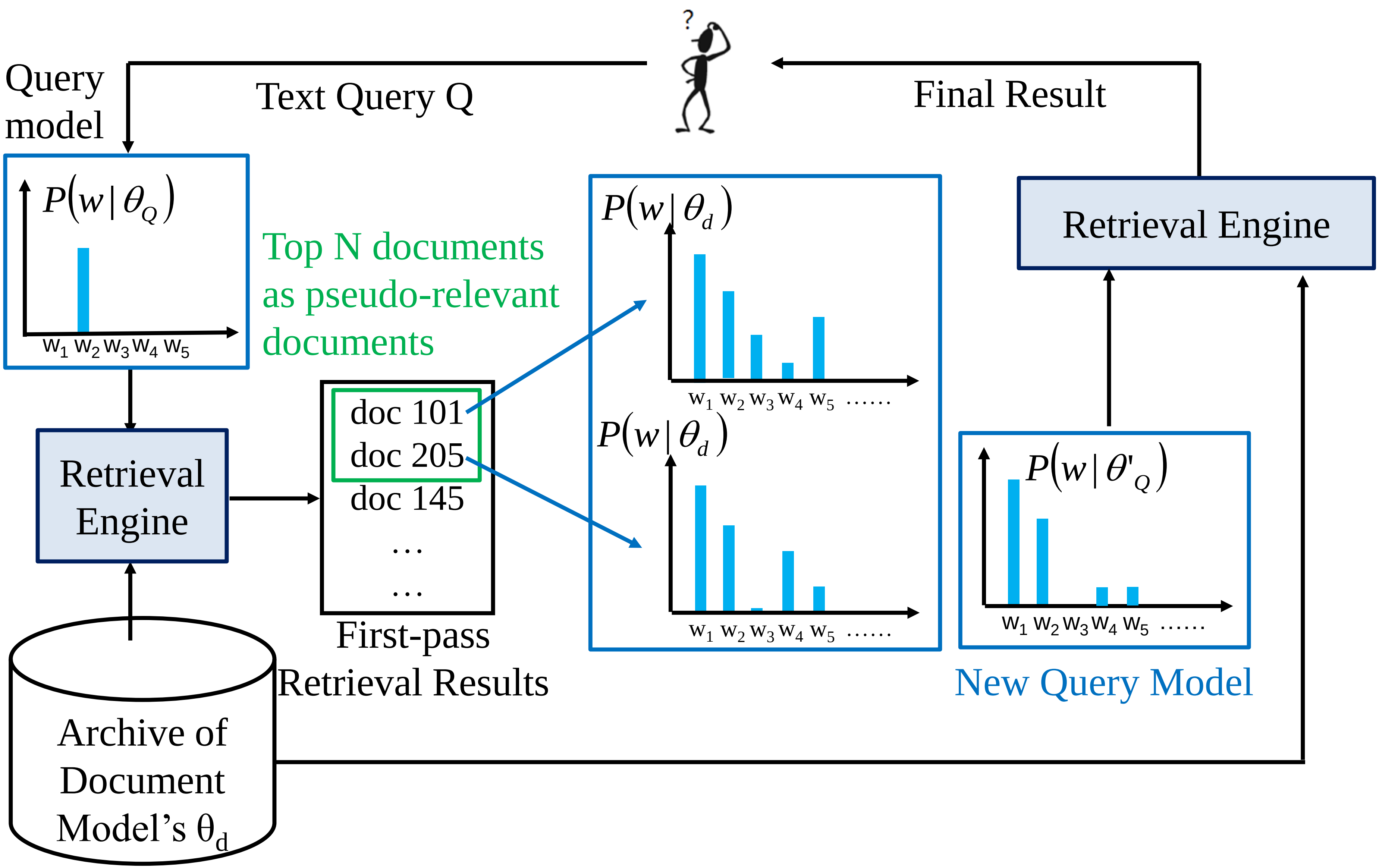
Semantic Retrieval by Query Expansion



Semantic Retrieval by Query Expansion



Semantic Retrieval by Query Expansion



Semantic Retrieval by Document Expansion

- **Document expansion**
  - Consider a document only has terms US and White House
  - Add some semantically related terms (Obama) into the document model
- **Document expansion for language modeling retrieval approach**

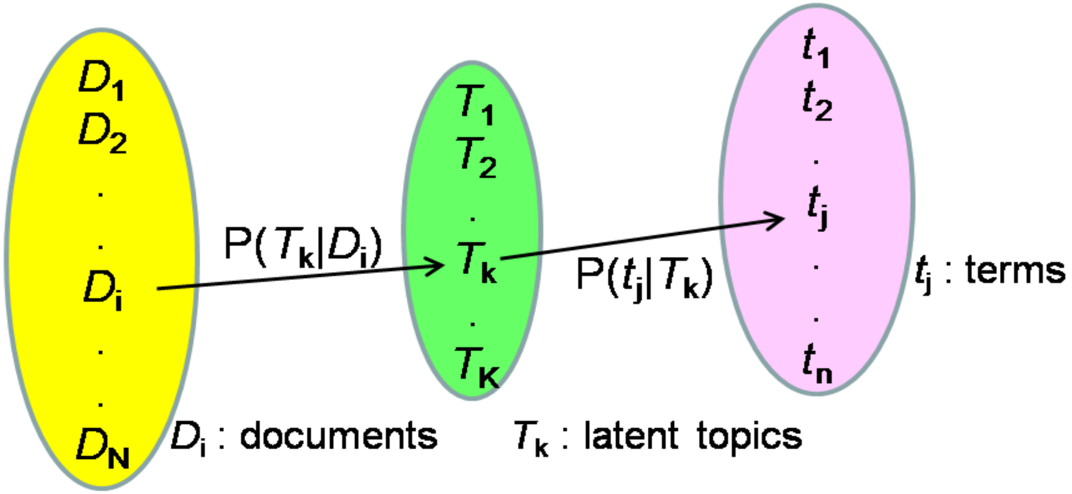
$$P(t \mid \theta_d') = \alpha P(t \mid \theta_d) + (1-\alpha)\sum_{i=1}^{K} P(t \mid T_i)P(T_i \mid d)$$

$P(T_i|d)$: probability of observing topic T_i given document d
$P(t|T_i)$: probability of observing term t given topic T_i
- Obtained by latent topic analysis (e.g. PLSA)

θ_d: original document model
α: interpolation weight
θ_d': expanded document model

Latent Topic Analysis

- **An example: Probabilistic Latent Semantic Analysis (PLSA)**
- **Creating a set of latent topics between a set of terms and a set of documents**



  – modeling the relationships by probabilistic models trained with EM algorithm

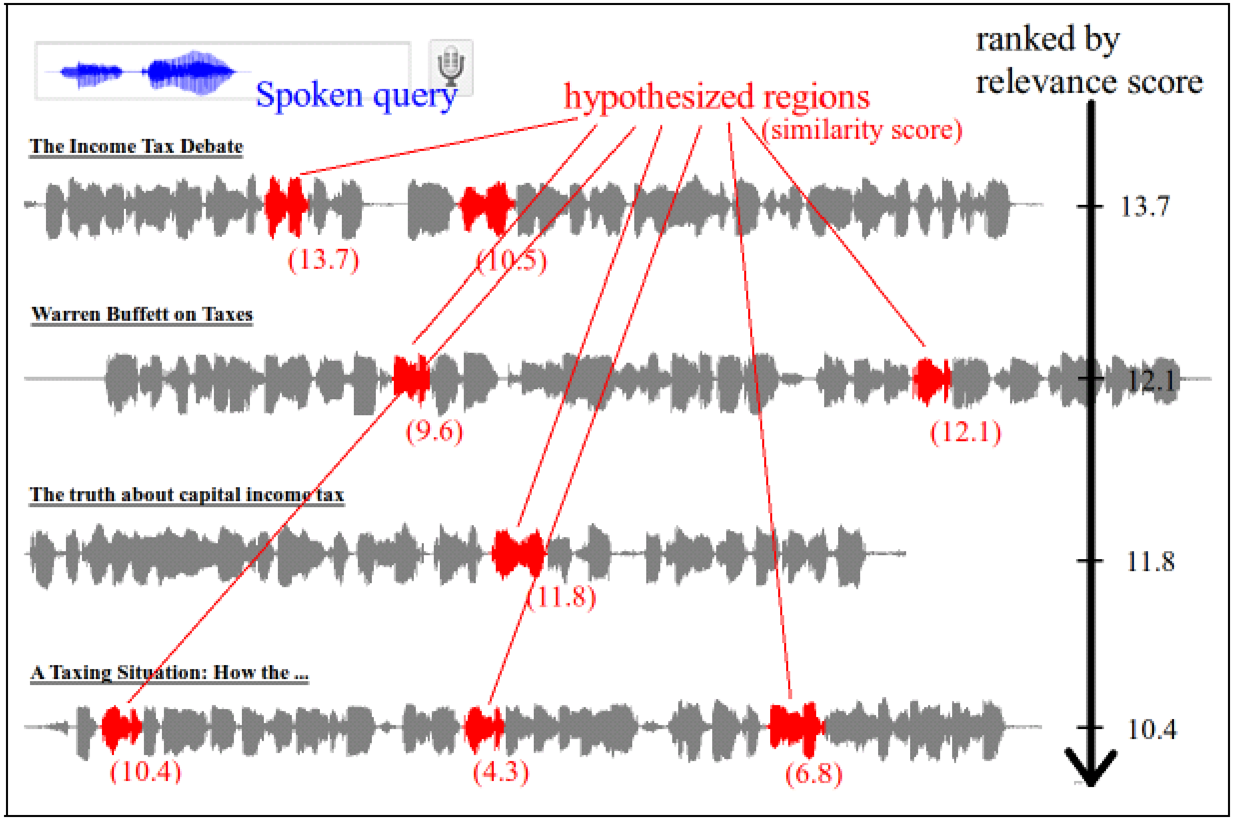
- **Other well-known approaches: Latent Semantic Analysis (LSA), Non-negative Matrix Factorization (NMF), Latent Dirichlet Allocation (LDA) … …**

References

- **Semantic Retrieval of Spoken Content**
  - "Improved Semantic Retrieval of Spoken Content by Language models Enhanced with Acoustic Similarity Graph", IEEE Workshop on Spoken Language Technology, 2012
  - T. K. Chia, K. C. Sim, H. Li, and H. T. Ng, "Statistical lattice-based spoken document retrieval," ACM Trans. Inf. Syst., vol. 28, pp. 2:1–2:30, 2010.

Unsupervised Spoken Term Detection (STD) with Spoken Queries

- Search speech by speech – no need to know which word is spoken
- No recognition, without annotated data, without knowledge about the language
- Bypass the difficulties of recognition : annotated data for the target domain, OOV words, recognition errors, noise conditions, etc.
  - relevance score ≡ highest similarity score within a document.

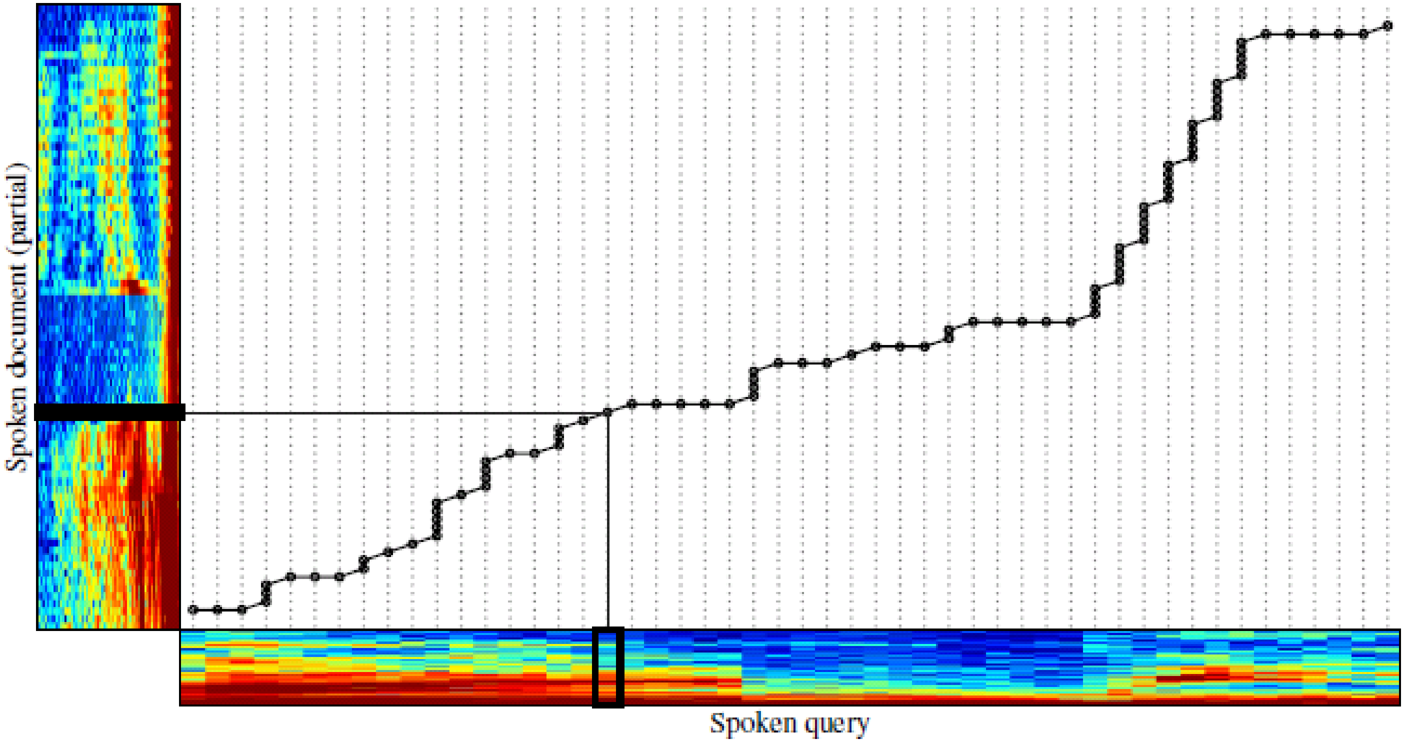


Two major approaches for Unsupervised STD

- **Template matching (signal-to-signal matching)**
  - Dynamic Time Warping (DTW) based, matching the signals directly
  - Precise but less compatible to signal variations (by different speakers, different acoustic conditions, etc.) with higher computation requirements
- **Model-based approach with automatically discovered patterns**
  - Representing signals by models and matching with these models
  - Discovering acoustic patterns and training corresponding models without annotated data

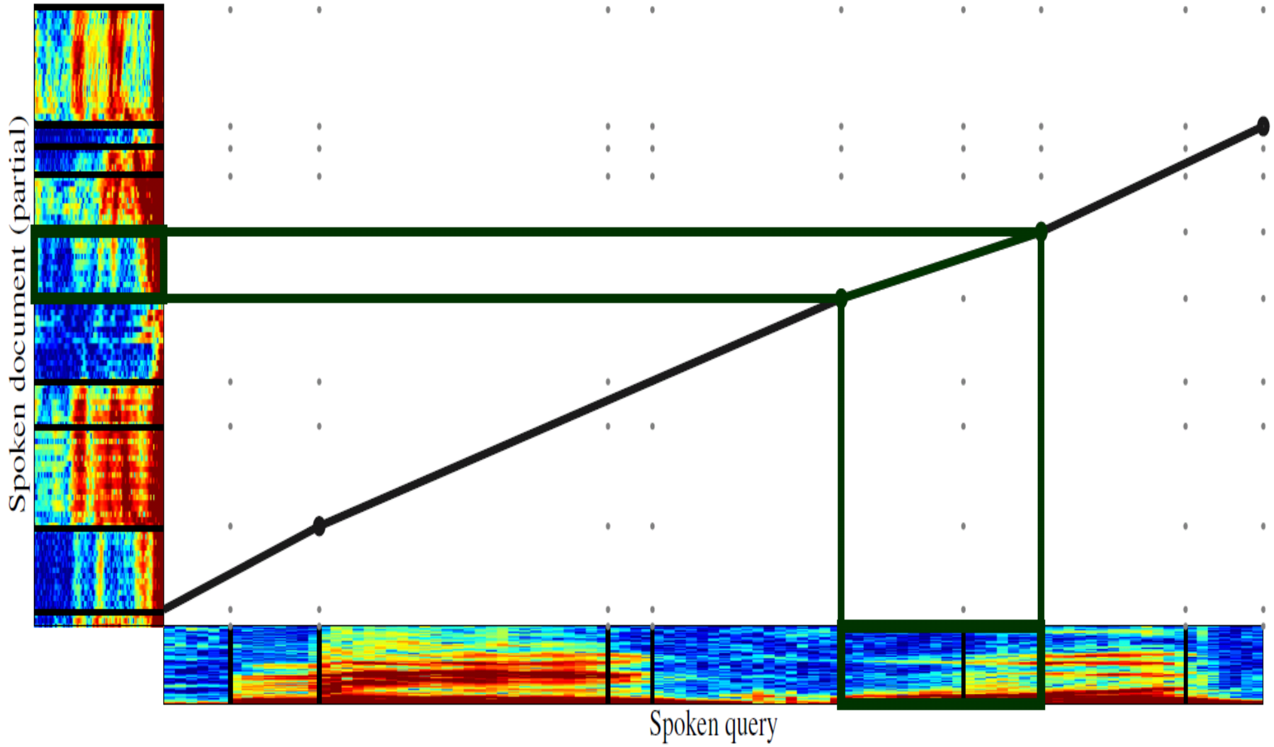
Template Matching

- **Dynamic time warping (DTW)**
  - Find possible speech regions that are similar to the query

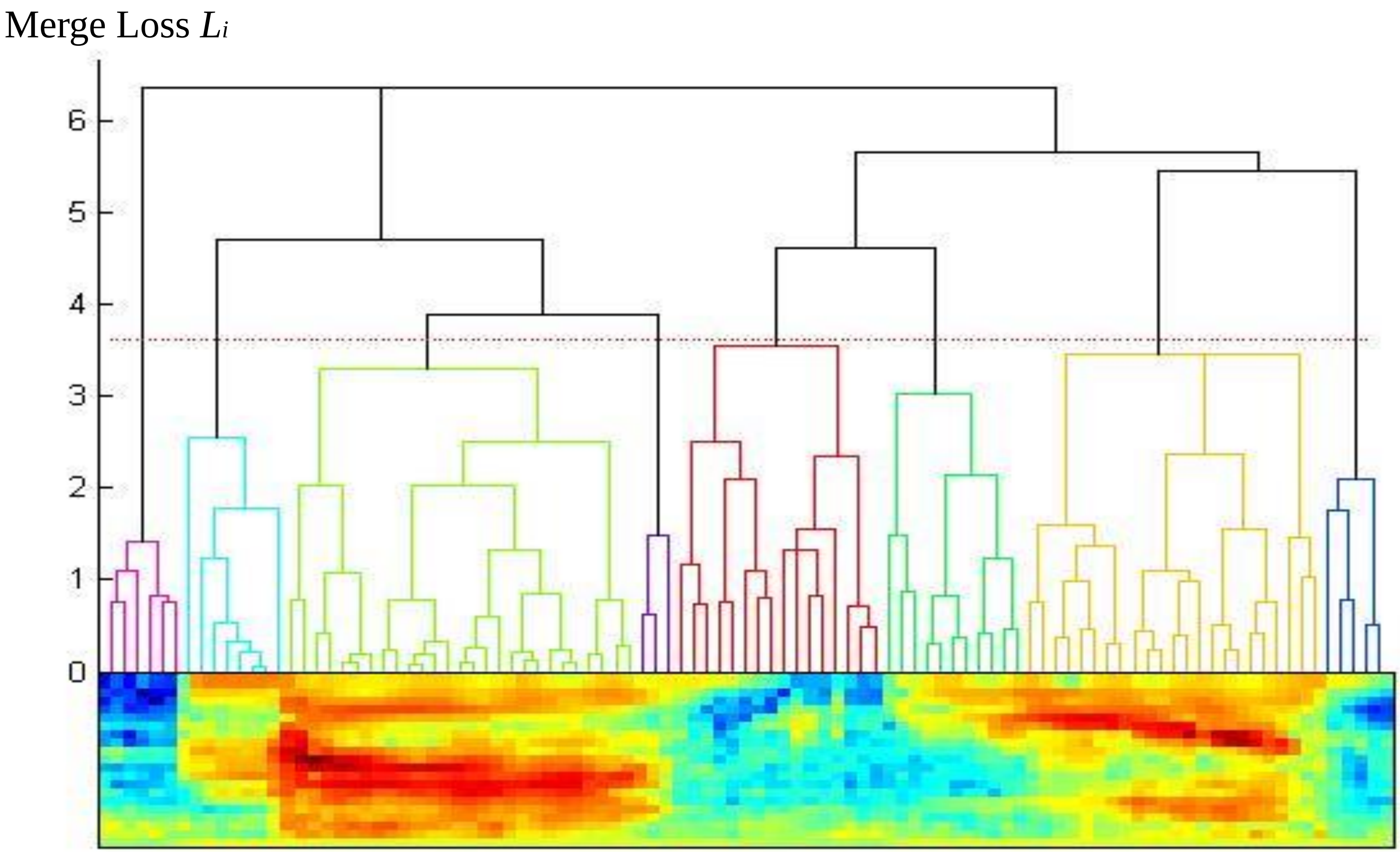


Template Matching

- **Segment-based DTW**
  - divide signals into segments of consecutive similar frames
  - segment-by-segment matching rather than frame-by-frame
  - Segment-based DTW (much faster but less precise) followed by frame-based DTW (slow but precise)



Hierarchical Agglomerative Clustering (HAC)

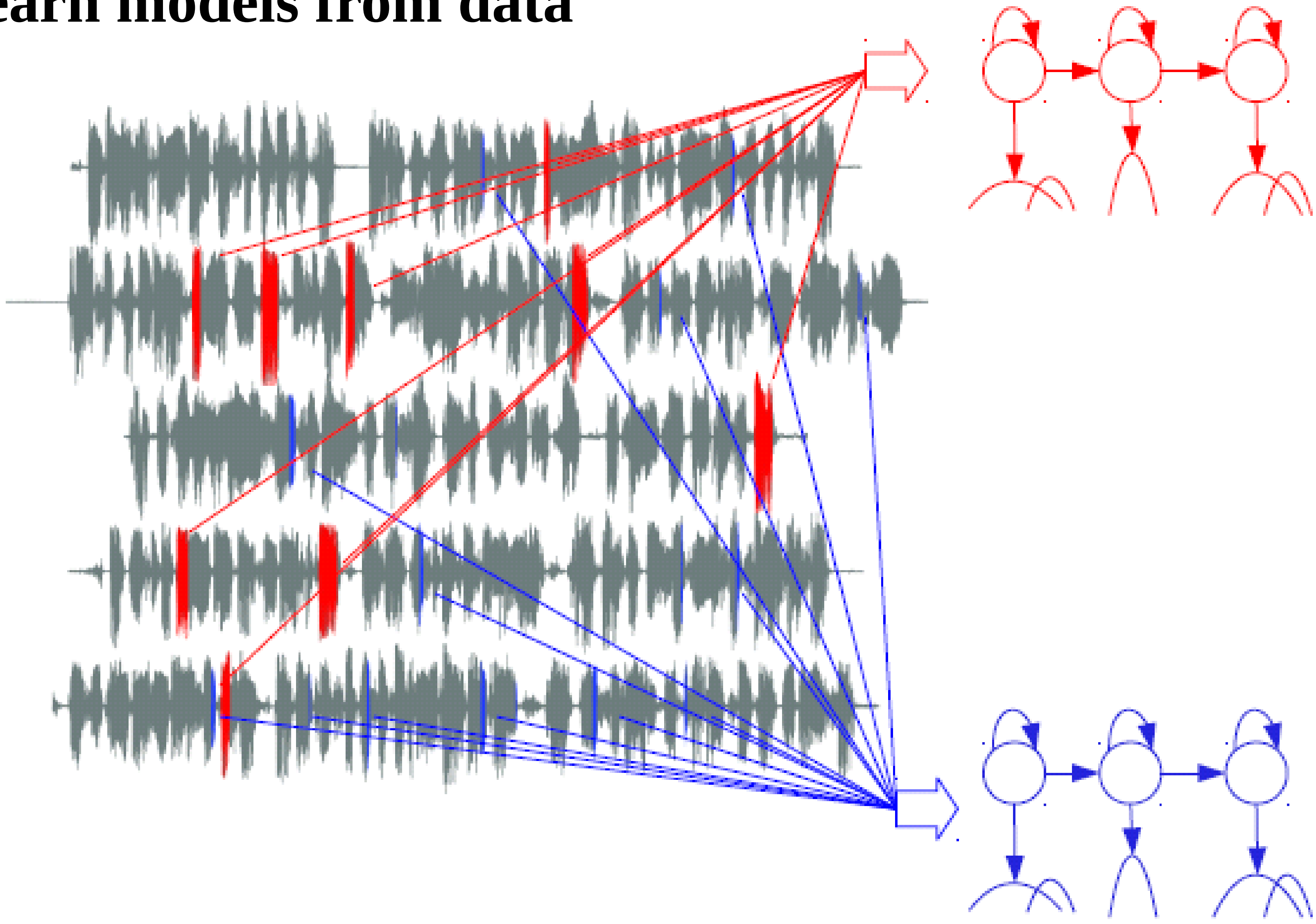


Hierarchical Agglomerative Clustering (HAC)

- **Initial Condition**
  - Each frame of signal (i.e. a MFCC vector) is a segment
- **Merge**
  - calculate the distance between each pair of adjacent segments
  - merge the pair with minimum distance into a single segment
  - represent the merged segment by a vector (e.g. the mean)
  - repeat

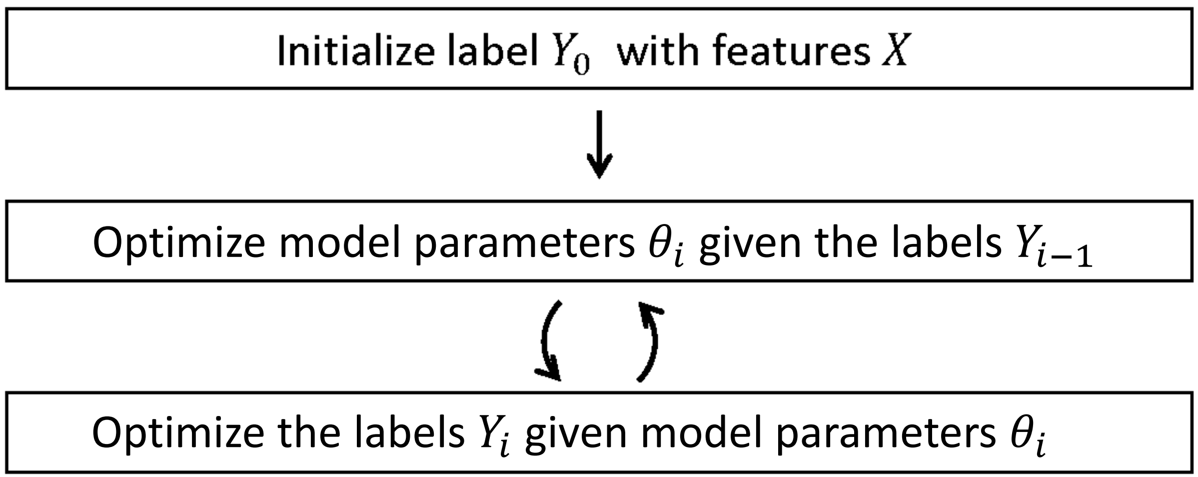
Model-based approach

- **Learn models from data**



Unsupervised Pattern Discovery

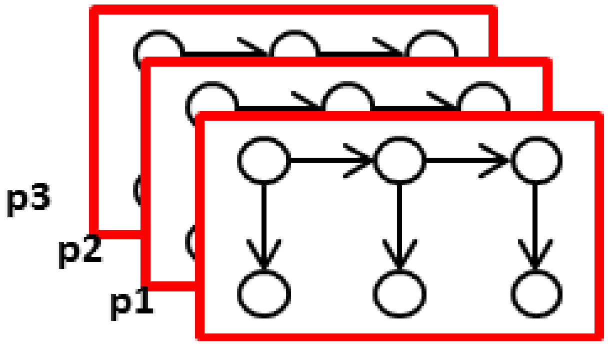
- **Unsupervised Discovery**
  - without annotated data
  - all patterns automatically learned from a set of corpora in unknown languages without linguistic knowledge



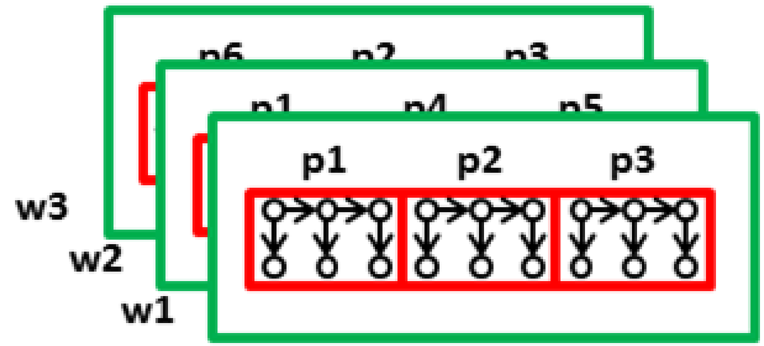
- **Initializing $\boldsymbol{Y_0}$**
  - signal segmentation (based on waveform-level features) followed by segment clustering
- **In each iteration i**
  - train the best set of HMM models θ_i based on Y_{i-1} and then obtain a new set of labels Y_i based on θ_i

Unsupervised Automatic Discovery of Linguistic Structure

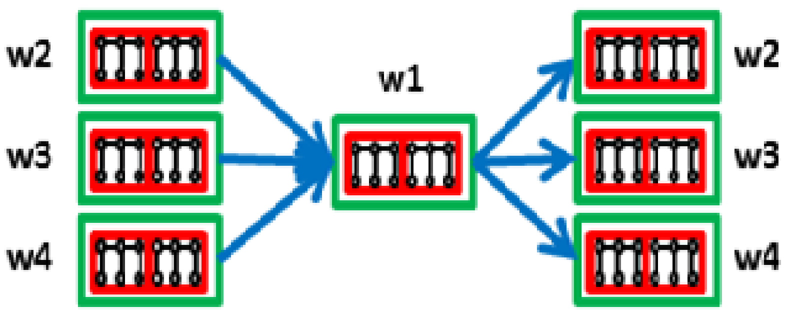
- **Hierarchical Linguistic Structure Automatically Discovered**
  - Subword-like pattern HMMs



  - Word-like pattern lexicon

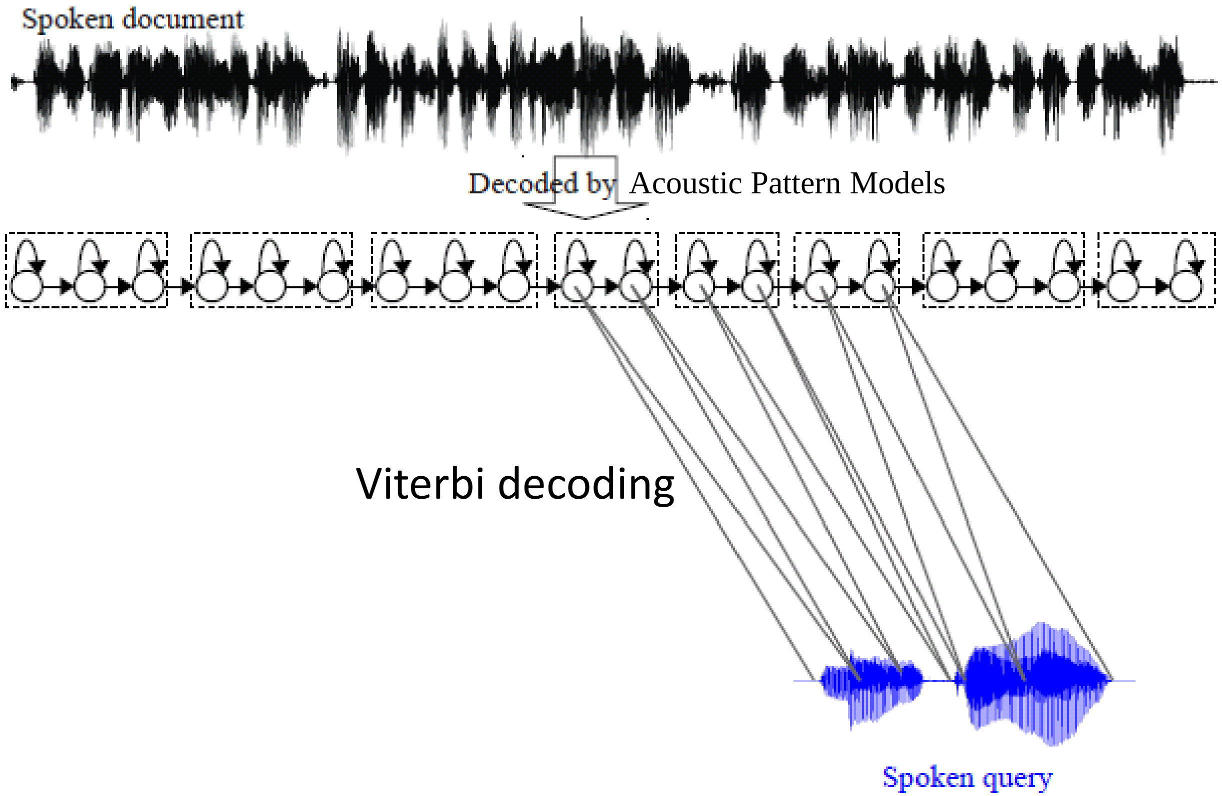


  - Word-like pattern language model



Search Based on Model of Acoustic patterns

- **Apply recognition-like approach with discovered models**



References

- **Unsupervised Discovery of Acoustic Patterns**
  – "Unsupervised Discovery of Linguistic Structure Including Two-level Acoustic Patterns Using Three Cascaded Stages of Iterative Optimization, " International Conference on Acoustics, Speech and Signal Processing, Vancouver, Canada, May 2013.
- **Unsupervised Spoken Term Detection**
  – "Integrating Frame-Based and Segment-Based Dynamic Time Warping for Unsupervised Spoken Term Detection with Spoken Queries", International Conference on Acoustics, Speech and Signal Processing, Prague, Czech Republic, May 2011, pp. 5652-5655.
  – "Toward Unsupervised Model-based Spoken Term Detection with Spoken Queries without Annotated Data, " International Conference on Acoustics, Speech and Signal Processing, May 2013
  – "Model-based Unsupervised Spoken Term Detection with Spoken Queries", IEEE Transactions on Audio, Speech, and Language Processing, Vol. 21, No. 7, Jul 2013, pp. 1330-1342.
- **HAC**
  – Unsupervised Optimal Phoneme Segmentation: Objectives, Algorithm and Comparisons, Yu Qiao, Naoya Shimomura, and Nobuaki Minematsu, ICASSP 2008

References

- **Mobile/Video Search**
  - "In-Car Media Search", IEEE Signal Processing Magazine, July 2011
  - "Speech and Multimodal Interaction in Mobile Search", IEEE Signal Processing Magazine, July 2011
  - "Reusing Speech Techniques for Video Semantic Indexing", IEEE Signal Processing Magazine, March 2013
- **Overall**
  - "Spoken Content Retrieval – Beyond Cascading Speech Recognition with Text Retrieval", IEEE/ACM Transactions on Audio, Speech and Language Processing, June 2015