

Automatic Key Term Extraction from Spoken Course Lectures

Using Branching Entropy and Prosodic/Semantic Features

Yun-Nung (**Vivian**) Chen, Yu Huang,
Sheng-Yi Kong, Lin-Shan Lee

National Taiwan University, Taiwan





Introduction



Definition

- Key Term
 - Higher term frequency
 - Core content
- Two types
 - Keyword
 - Key phrase
- Advantage
 - Indexing and retrieval
 - The relations between key terms and segments of documents



Introduction

NTU VIRTUAL INSTRUCTOR

Lecture Search:

INTRODUCTION TO DIGITAL SPEECH PROCESSING

Chapter 1

Chapter 2

Chapter 3

Chapter 4

Chapter 5

Chapter 6

Chapter 7

Chapter 8

8-1

8-2

8-3

8-4

8-4 TIME-SYNCHRONOUS VITERBI SEARCH FOR LVCSR

LENGTH:

0:25:23.0

TIME SPAN OF THIS CHAPTER:

CH 1

CH 18

TIME SPAN OF THIS SLIDE:

8-1

8-9

Play Summary
(0:02:32.3)

Play Whole

KEY TERMS:

acoustic model	viterbi	lexicon
language model	phone	n gram
hmm	hmm	language
n gram	hidden markov model	phone
phone	delta	utterance
hidden markov model	markov model	trigram

Only those key terms in black on the top of each column are the key terms appearing in this slide

Time (Frame)- Synchronous Viterbi Search for Large-Vocabulary Continuous Speech Recognition

•MAP Principle

$$W^* = \arg \max_W [p(W|X)] = \arg \max_W \left[\frac{p(X|W)p(W)}{p(X)} \right] = \arg \max_W \left[\underset{\text{from IDMM}}{p(X|W)} \underset{\text{from Language Model}}{p(W)} \right]$$

$$p(X|W) = \sum_{\tilde{q}} p(X, \tilde{q}|W), \tilde{q}: \text{a state sequence}$$

•An Approximation

$$W^* = \arg \max_W [p(W) \sum_{\tilde{q}} p(X, \tilde{q}|W)] \approx \arg \max_W [p(W) \cdot \arg \max_{\tilde{q}} p(X, \tilde{q}|W)]$$

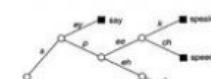
- the most likely word sequence is approximated by the most likely state sequence
- Viterbi search, a sub-optimal approach

•Viterbi Search—Dynamic Programming

- replacing the problem by a smaller sub-problem and formulating an iterative procedure
- time (frame)- synchronous; the best score at time t is updated from all states at time t-1

•Tree Lexicon as the Basic Working Structure

- each arc is an HMM
- each leaf node is a word
- search processes for a segment of utterance through some common units for different words can be shared
- the same tree copy reproduced at each leaf node in principle





Introduction

NTU VIRTUAL INSTRUCTOR

INTRODUCTION TO
DIGITAL SPEECH
PROCESSING

Chapter 1

Chapter 2

Chapter 3

Chapter 4

Chapter 5

Chapter 6

Chapter 7

Chapter 8

8-1

8-2

8-3

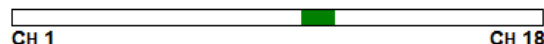
8-4

8-4 TIME-SYNCHRONOUS VITERBI SEARCH FOR L

LENGTH:

0:25:23.0

TIME SPAN OF THIS CHAPTER:



TIME SPAN OF THIS SLIDE:



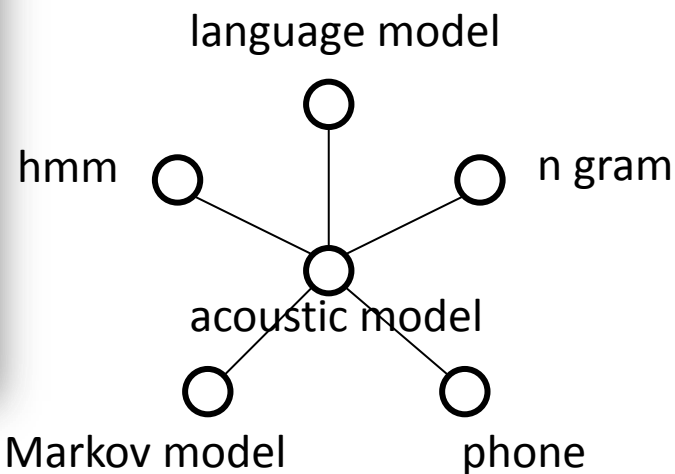
Play Summary
(0:02:32.3)

Play Whole

KEY TERMS:

acoustic model	viterbi	lexicon
language model	phone	n gram
hmm	hmm	language
n gram	hidden markov model	phone
phone	delta	utteran
hidden markov model	markov model	trigram

Only those key terms
each column are the
in this slide





Introduction

NTU VIRTUAL INSTRUCTOR

INTRODUCTION TO
DIGITAL SPEECH
PROCESSING

Chapter 1

Chapter 2

Chapter 3

Chapter 4

Chapter 5

Chapter 6

Chapter 7

Chapter 8

8-1

8-2

8-3

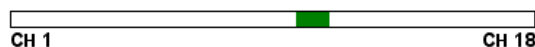
8-4

8-4 TIME-SYNCHRONOUS VITERBI SEARCH FOR L

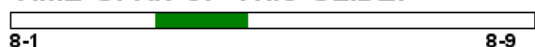
LENGTH:

0:25:23.0

TIME SPAN OF THIS CHAPTER:



TIME SPAN OF THIS SLIDE:



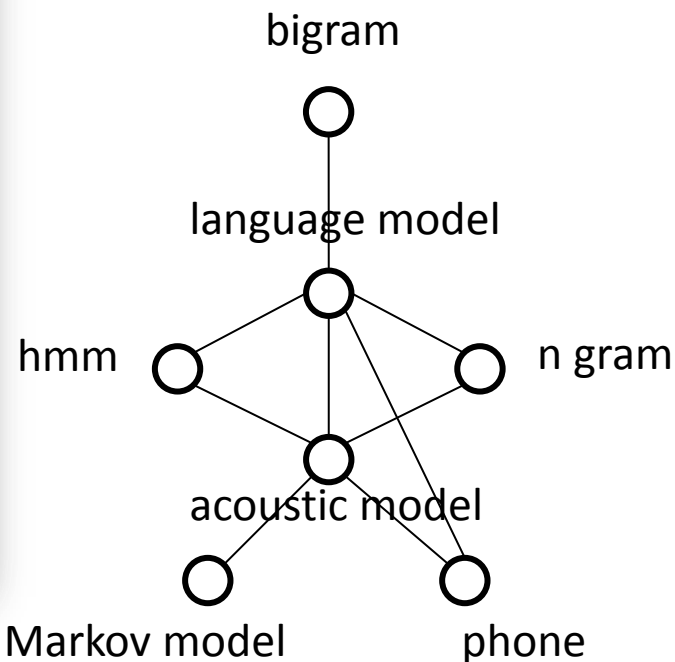
Play Summary
(0:02:32.3)

Play Whole

KEY TERMS:

language model	phone	hmm
n gram	hmm	phone
hmm	triphone	language model
acoustic model	language model	gaussian
bigram	n gram	markov model
phone	syllable	hidden markov n

Related
Key Ter



hidden Markov model

phone

Target: extract key terms from course lectures

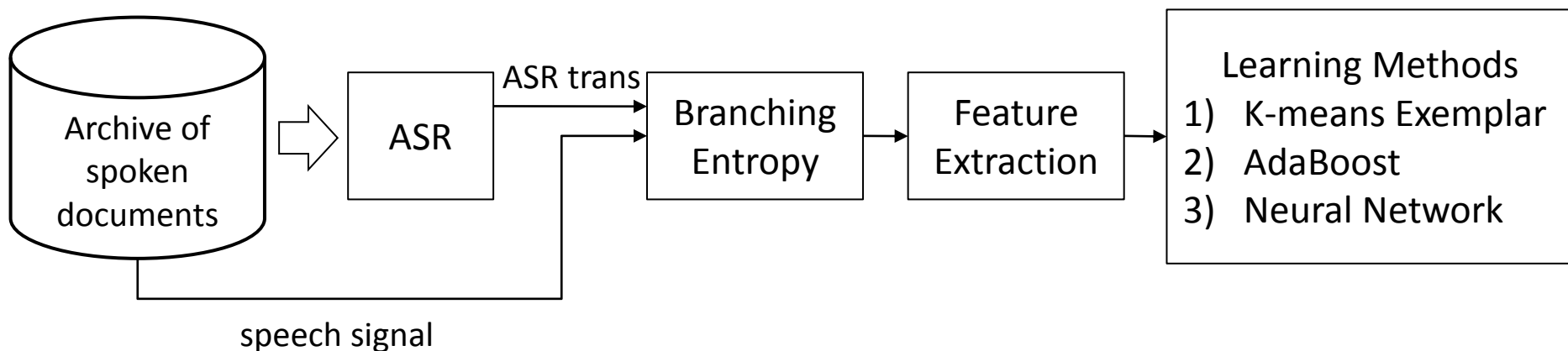


Proposed Approach



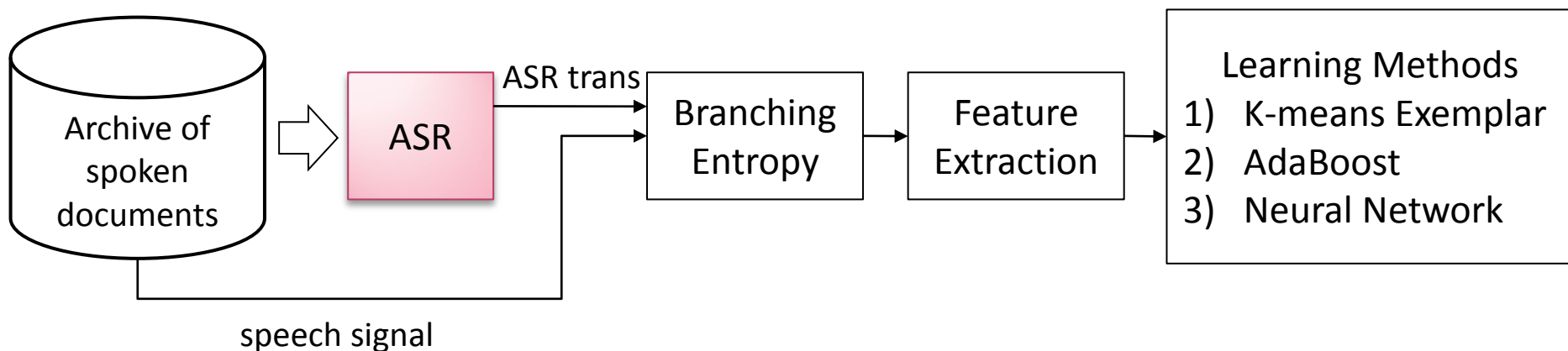
Automatic Key Term Extraction

▼ Original spoken documents



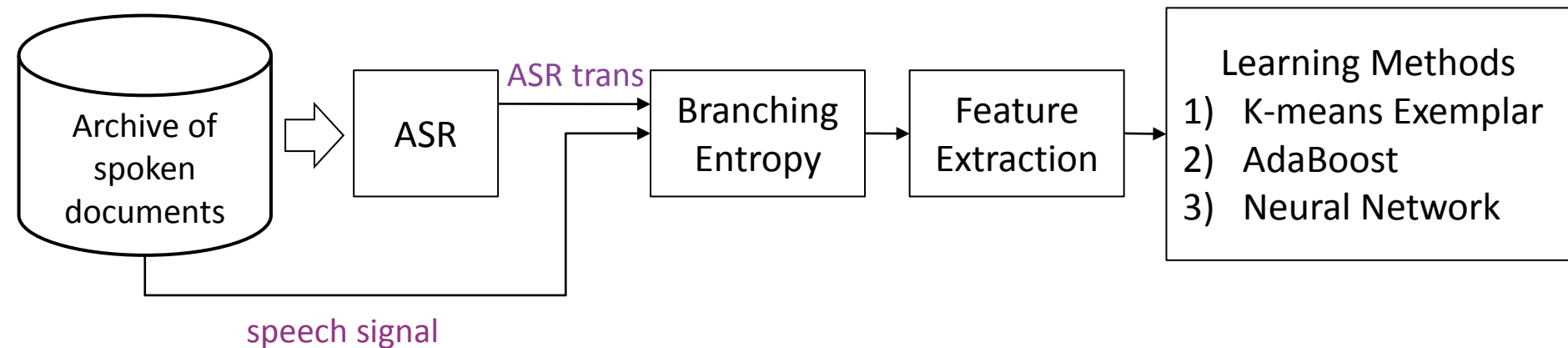


Automatic Key Term Extraction



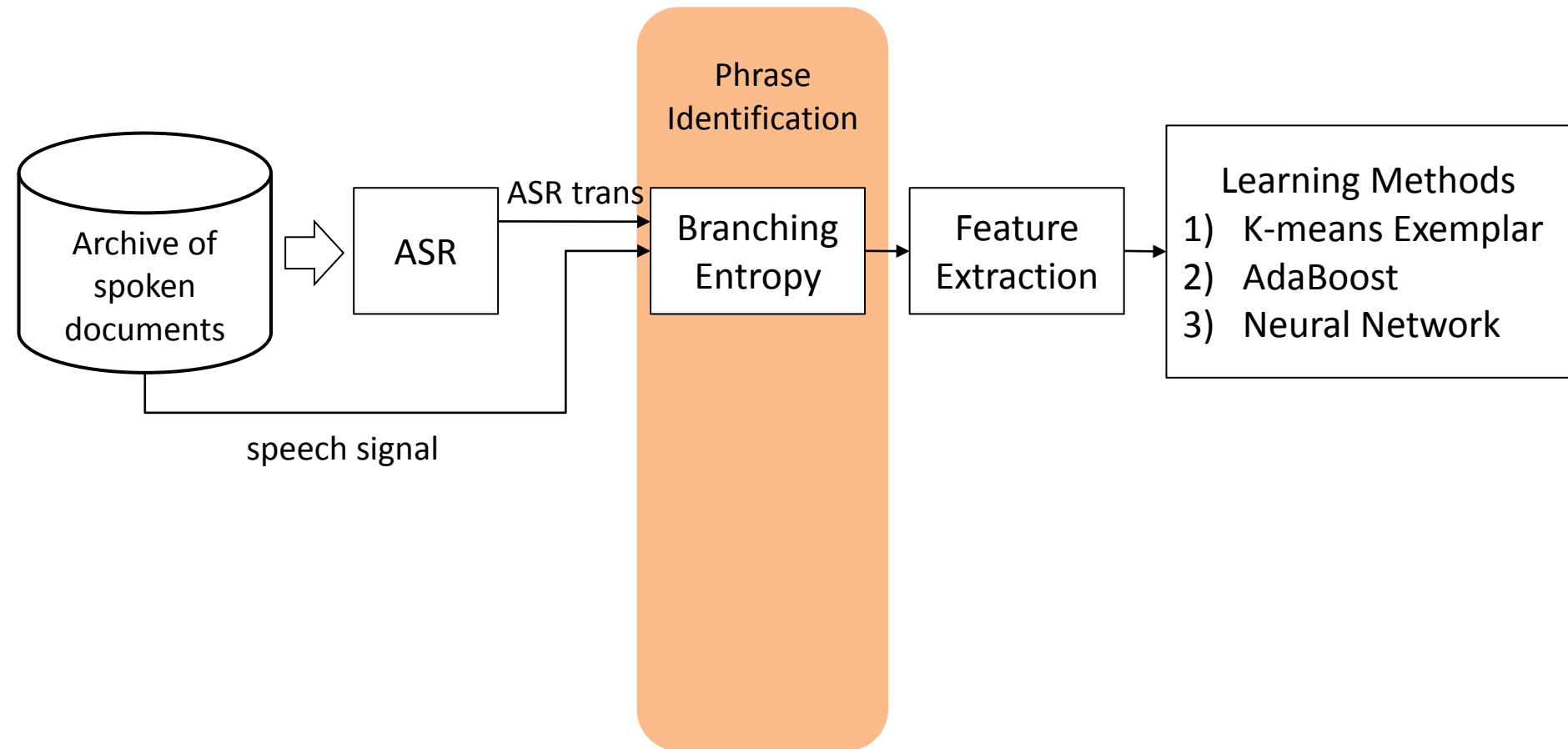


Automatic Key Term Extraction





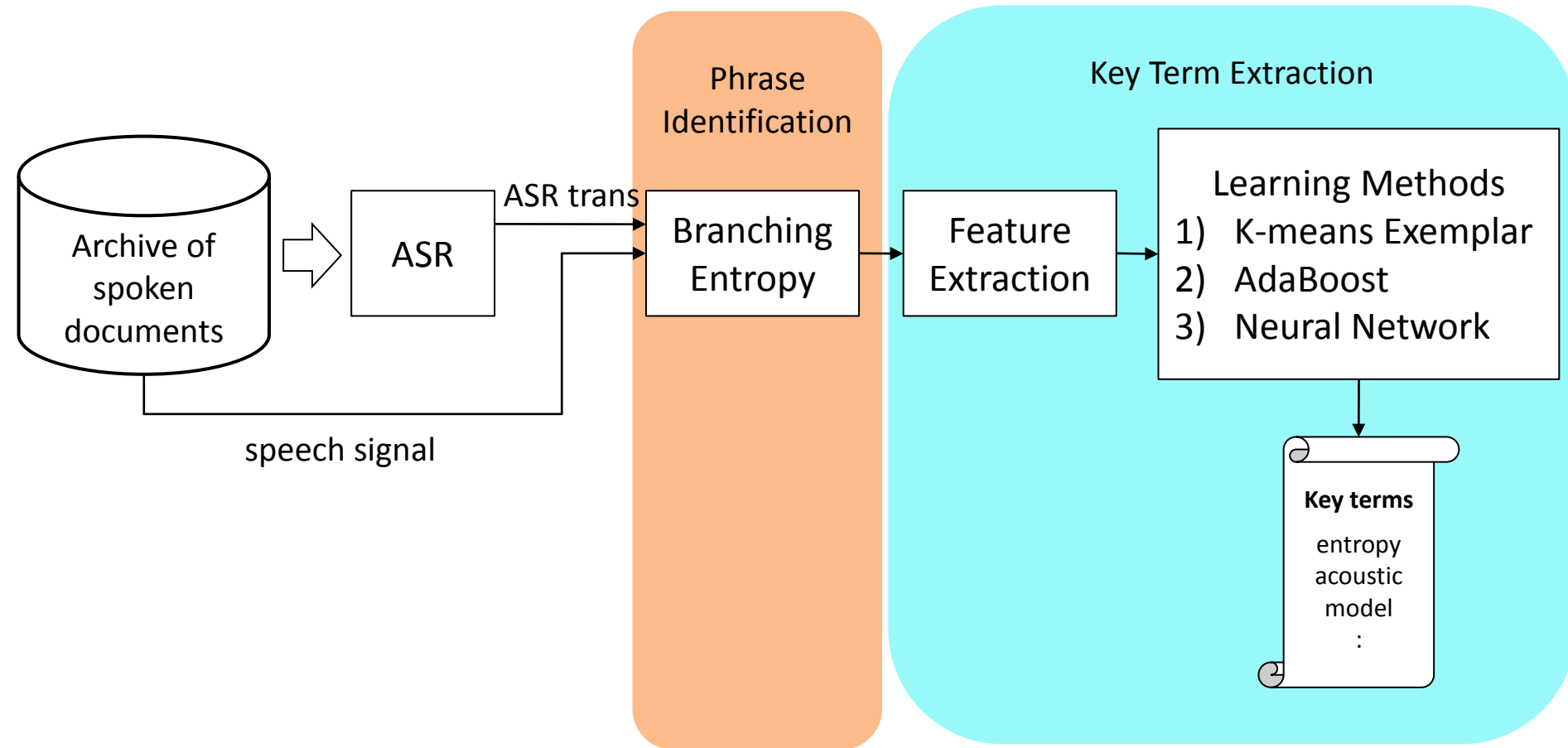
Automatic Key Term Extraction



First using branching entropy to identify phrases



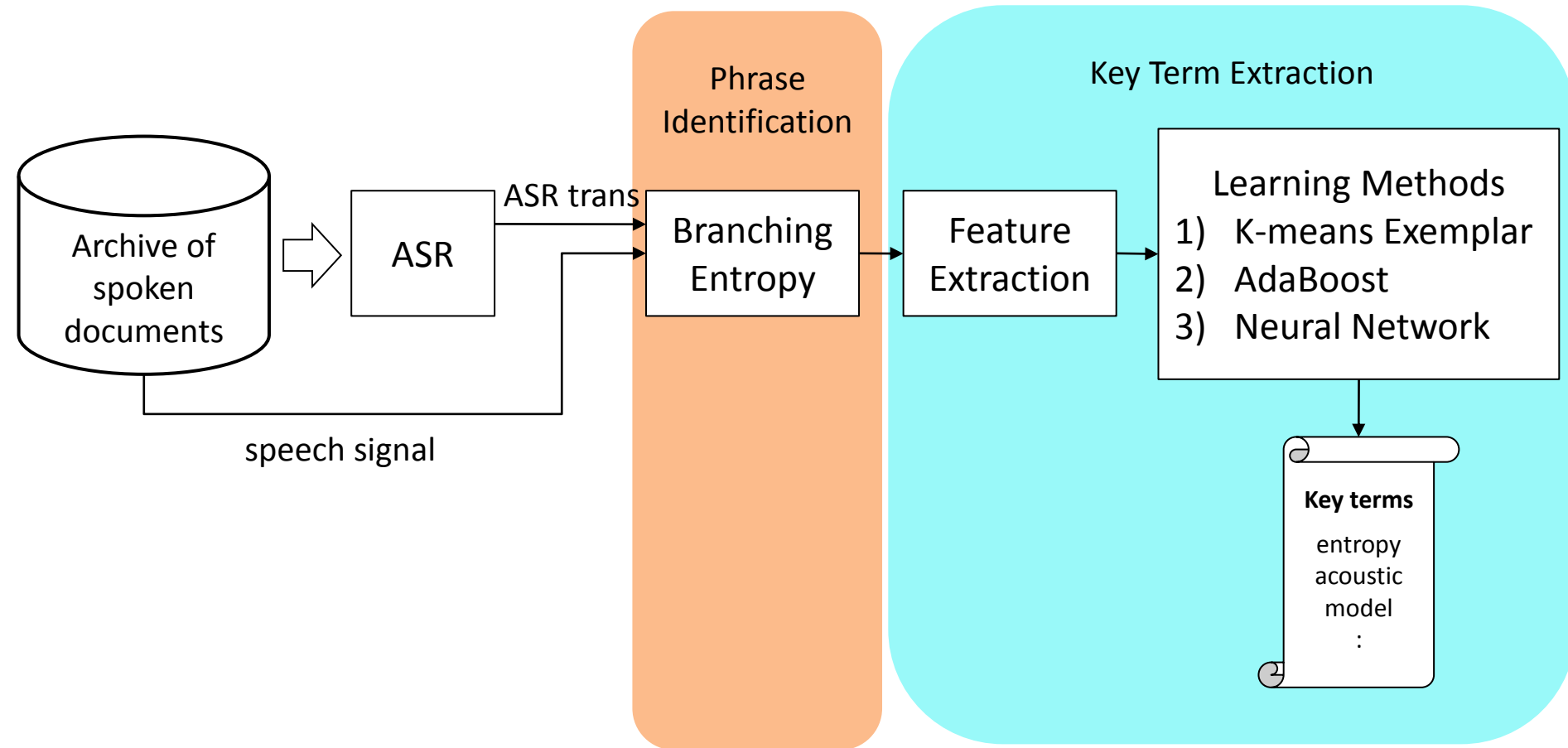
Automatic Key Term Extraction



Then using learning methods to extract key terms by some features



Automatic Key Term Extraction





Branching Entropy

How to decide the boundary of a phrase?



- “hidden” is almost always followed by the same word



Branching Entropy

How to decide the boundary of a phrase?



- “hidden” is almost always followed by the same word
- “hidden Markov” is almost always followed by the same word



Branching Entropy

How to decide the boundary of a phrase?



- “hidden” is almost always followed by the same word
- “hidden Markov” is almost always followed by the same word
- “hidden Markov model” is followed by many different words

Define branching entropy to decide possible boundary



Branching Entropy

How to decide the boundary of a phrase?



- Definition of Right Branching Entropy

- Probability of children x_i for X

$$p(x_i) = \frac{f_{x_i}}{f_X} \quad \begin{array}{l} X: w_1 \dots w_k \\ x_i: w_1 \dots w_k w_{(k+1)}^i \end{array}$$

- Right branching entropy for X

$$H_r(X) = - \sum_{i=1}^n p(x_i) \log_2 p(x_i)$$



Branching Entropy

How to decide the boundary of a phrase?



- Decision of Right Boundary
 - Find the right boundary located between X and x_i where

$$H_r(X) > \text{average } H_r(X)$$



Branching Entropy

How to decide the boundary of a phrase?





Branching Entropy

How to decide the boundary of a phrase?





Branching Entropy

How to decide the boundary of a phrase?





Branching Entropy

How to decide the boundary of a phrase?



- Decision of Left Boundary

- Find the left boundary located between \bar{X} and x_i where

$$H_l(\bar{X}) = - \sum_{i=1}^n p(x_i) \log_2 p(x_i)$$

\bar{X} : model Markov hidden

$$H_l(\bar{X}) > \text{average } H_l(\bar{X})$$

Using PAT Tree to implement



Branching Entropy

How to decide the boundary of a phrase?

- Implementation in the PAT tree
 - Probability of children x_i for X

$$p(x_i) = \frac{f_{x_i}}{f_X} \quad \begin{array}{l} X: w_1 \dots w_k \\ x_i: w_1 \dots w_k w_{(k+1)}^i \end{array}$$

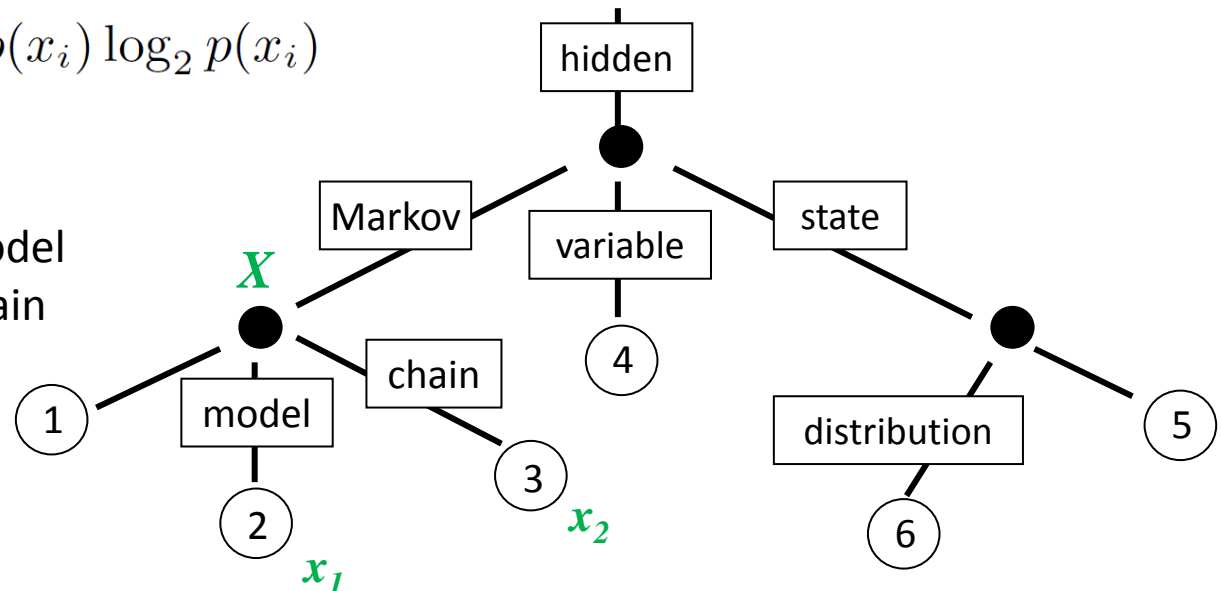
- Right branching entropy for X

$$H_r(X) = - \sum_{i=1}^n p(x_i) \log_2 p(x_i)$$

X : hidden Markov

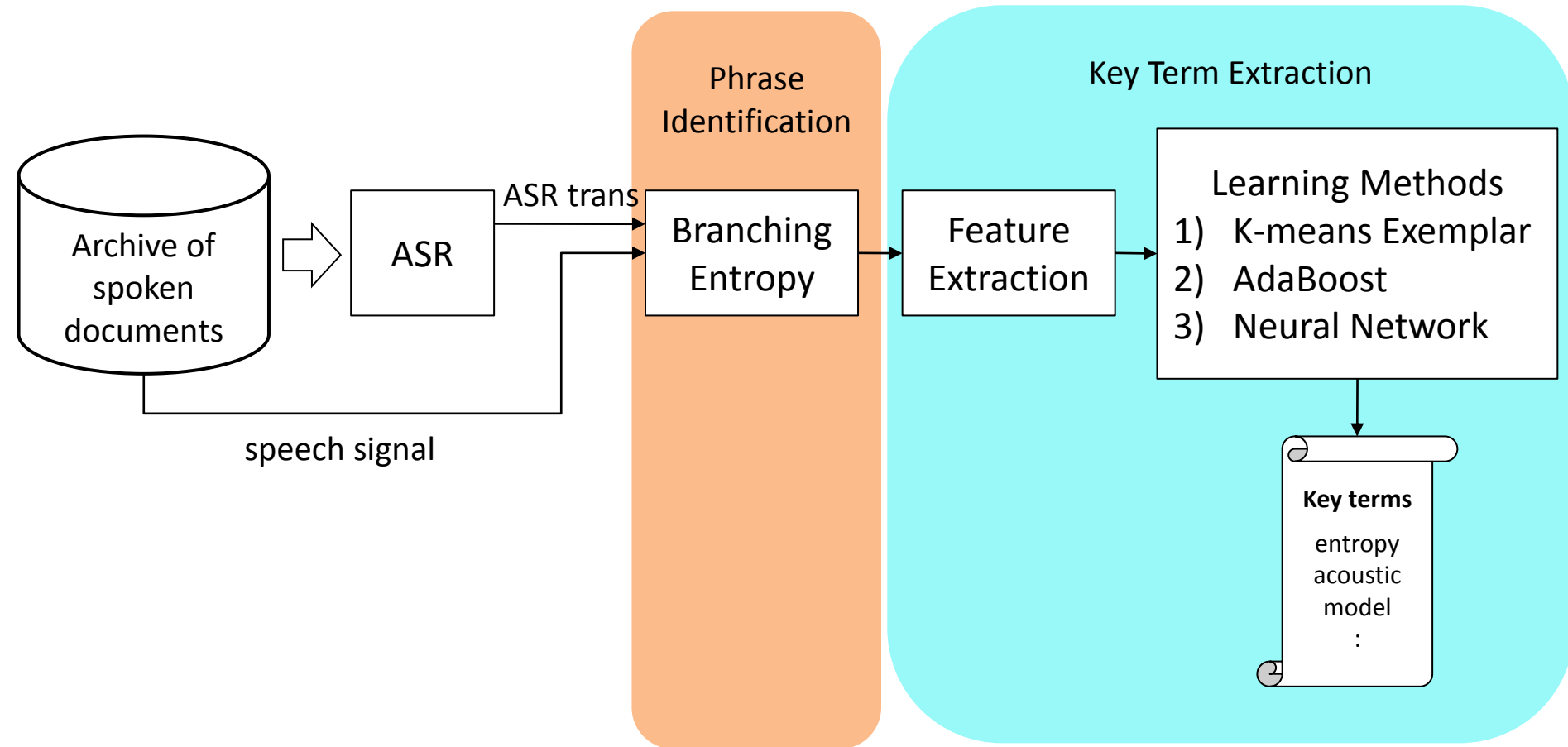
x_1 : hidden Markov model

x_2 : hidden Markov chain





Automatic Key Term Extraction



Extract prosodic, lexical, and semantic features for each candidate term

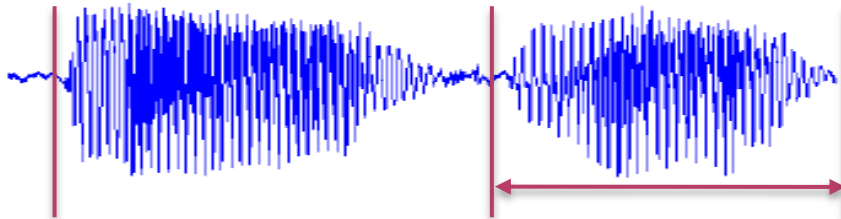


Feature Extraction

- Prosodic features

Speaker tends to use longer duration to emphasize key terms

- For each candidate term appearing at the first time



duration of phone "a" normalized by
avg duration of phone "a"

Feature Name	Feature Description
Duration (I – IV)	normalized duration (max, min, mean, range)

using 4 values for
duration of the term

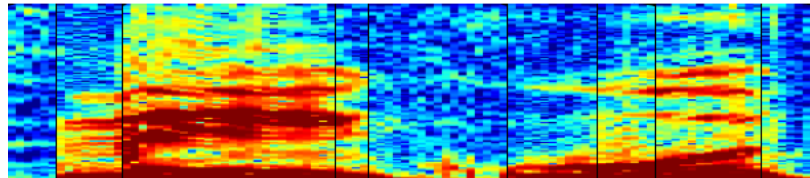
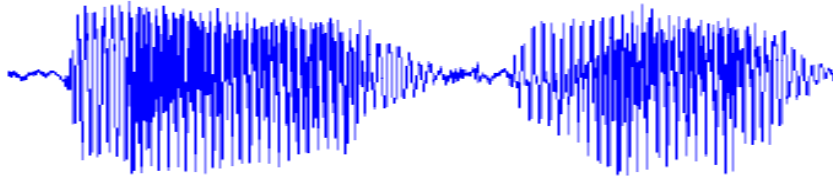


Feature Extraction

- Prosodic features

Higher pitch may represent significant information

- For each candidate term appearing at the first time



Feature Name	Feature Description
Duration (I – IV)	normalized duration (max, min, mean, range)

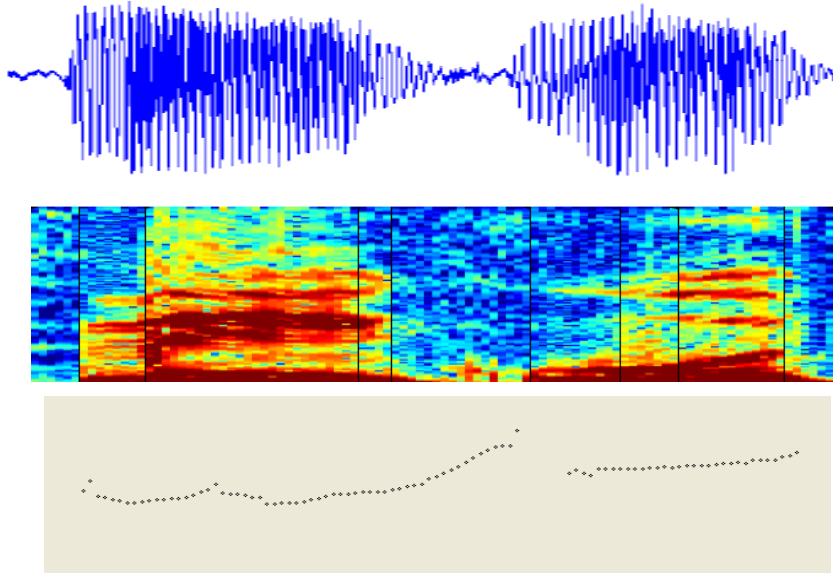


Feature Extraction

- Prosodic features

Higher pitch may represent significant information

- For each candidate term appearing at the first time



Feature Name	Feature Description
Duration (I – IV)	normalized duration (max, min, mean, range)
Pitch (I - IV)	F0 (max, min, mean, range)

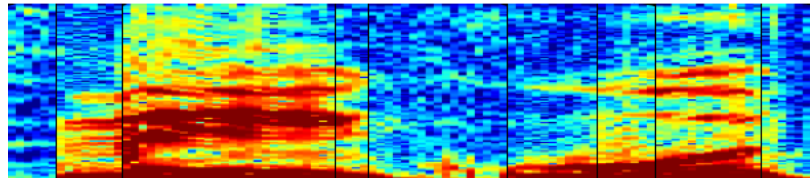
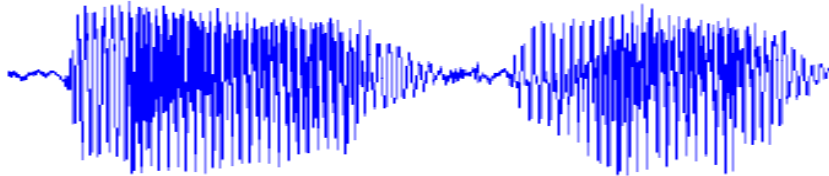


Feature Extraction

- Prosodic features

Higher energy emphasizes important information

- For each candidate term appearing at the first time



Feature Name	Feature Description
Duration (I – IV)	normalized duration (max, min, mean, range)
Pitch (I - IV)	F0 (max, min, mean, range)

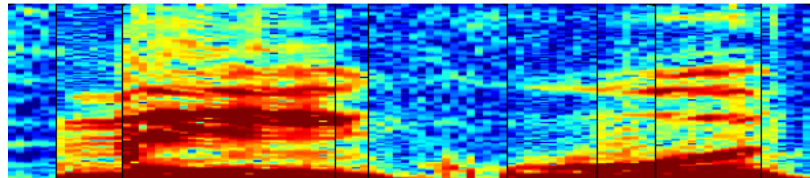
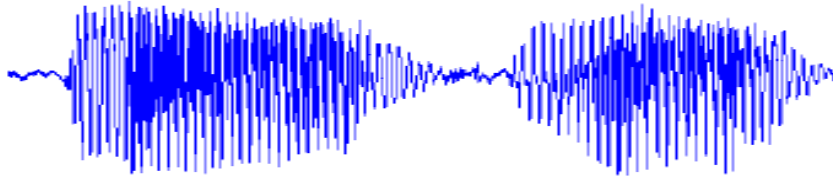


Feature Extraction

- Prosodic features

Higher energy emphasizes important information

- For each candidate term appearing at the first time



Feature Name	Feature Description
Duration (I – IV)	normalized duration (max, min, mean, range)
Pitch (I - IV)	F0 (max, min, mean, range)
Energy (I - IV)	energy (max, min, mean, range)



Feature Extraction

- Lexical features

Feature Name	Feature Description
TF	term frequency
IDF	inverse document frequency
TFIDF	$tf * idf$
PoS	the PoS tag

Using some well-known lexical features for each candidate term



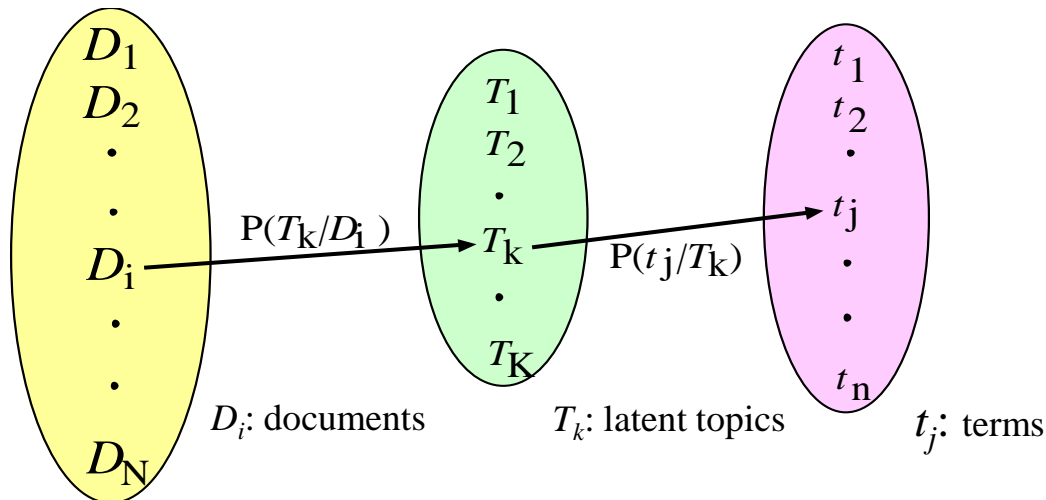
Feature Extraction

- Semantic features

Key terms tend to focus on limited topics

- Probabilistic Latent Semantic Analysis (PLSA)
 - Latent Topic Probability

$$P(T_k | t_i) = \frac{P(t_i | T_k) P(T_k)}{P(t_i)}$$





Feature Extraction

- Semantic features

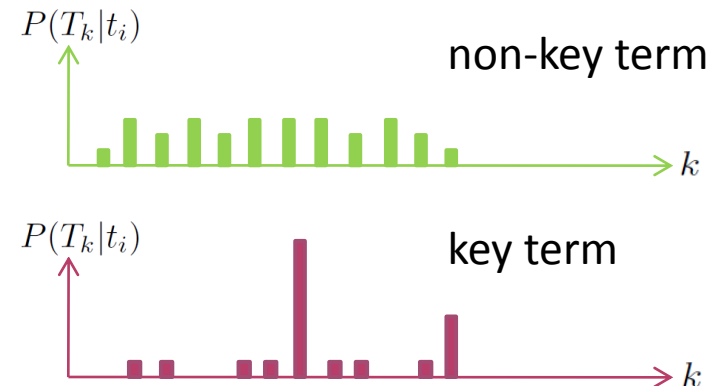
Key terms tend to focus on limited topics

- Probabilistic Latent Semantic Analysis (PLSA)

- Latent Topic Probability

$$P(T_k|t_i) = \frac{P(t_i|T_k)P(T_k)}{P(t_i)}$$

How to use it?



Feature Name	Feature Description
LTP (I - III)	Latent Topic Probability (mean, variance, standard deviation)

describe a probability distribution



Feature Extraction

- Semantic features

Key terms tend to focus on limited topics

- Probabilistic Latent Semantic Analysis (PLSA)

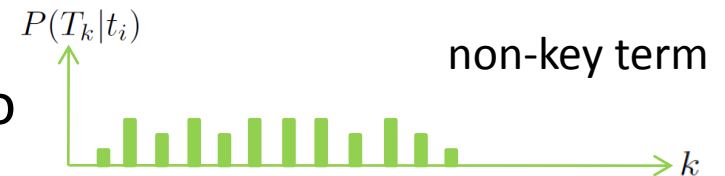
- Latent Topic Significance

Within-topic to out-of-topic ratio

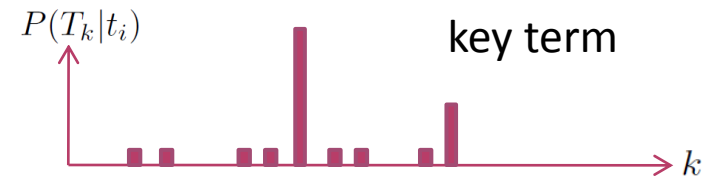
$$S_{t_i}(T_k) =$$

within-topic freq. $\sum_{d_j \in D} n(t_i, d_j) P(T_k | d_j)$

out-of-topic freq. $\sum_{d_j \in D} n(t_i, d_j) [1 - P(T_k | d_j)]$



non-key term



key term

Feature Name	Feature Description
LTP (I - III)	Latent Topic Probability (mean, variance, standard deviation)



Feature Extraction

- Semantic features

Key terms tend to focus on limited topics

- Probabilistic Latent Semantic Analysis (PLSA)

- Latent Topic Significance

Within-topic to out-of-topic ratio

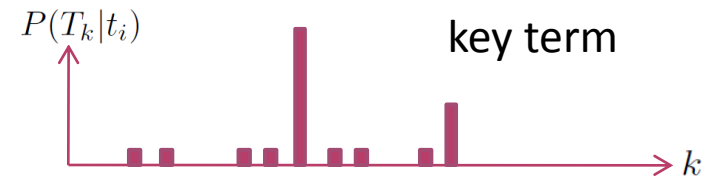
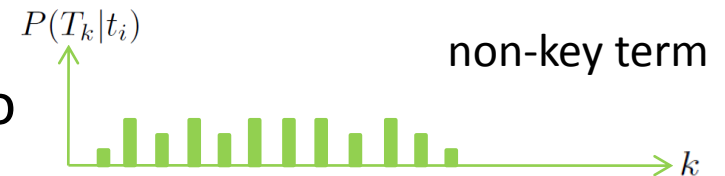
$$S_{t_i}(T_k) =$$

within-topic freq.

$$\sum_{d_j \in D} n(t_i, d_j) P(T_k | d_j)$$

out-of-topic freq.

$$\sum_{d_j \in D} n(t_i, d_j) [1 - P(T_k | d_j)]$$



Feature Name	Feature Description
LTP (I - III)	Latent Topic Probability (mean, variance, standard deviation)
LTS (I - III)	Latent Topic Significance (mean, variance, standard deviation)



Feature Extraction

- Semantic features

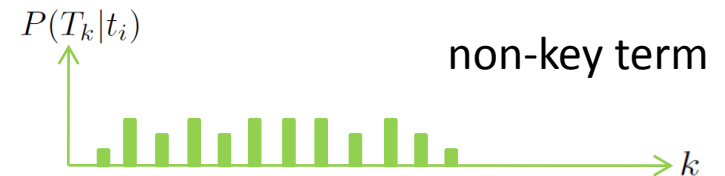
Key terms tend to focus on limited topics

- Probabilistic Latent Semantic Analysis (PLSA)

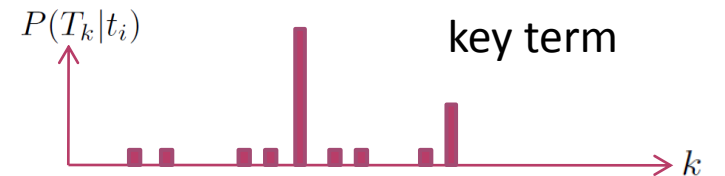
- Latent Topic Entropy

$$EN(t_i) =$$

$$- \sum_{k=1}^K P(T_k|t_i) \log P(T_k|t_i)$$



non-key term



key term

Feature Name	Feature Description
LTP (I - III)	Latent Topic Probability (mean, variance, standard deviation)
LTS (I - III)	Latent Topic Significance (mean, variance, standard deviation)



Feature Extraction

- Semantic features

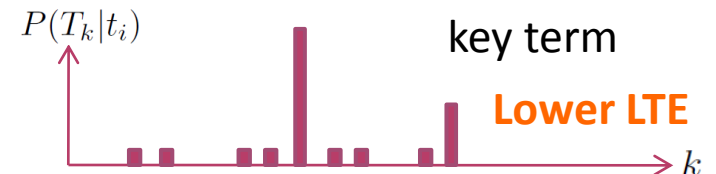
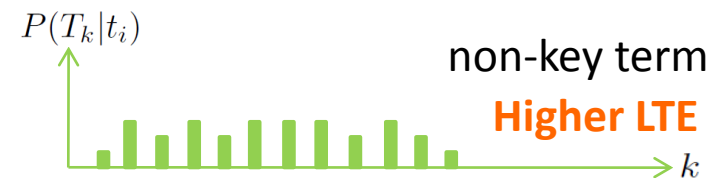
Key terms tend to focus on limited topics

- Probabilistic Latent Semantic Analysis (PLSA)

- Latent Topic Entropy

$$EN(t_i) =$$

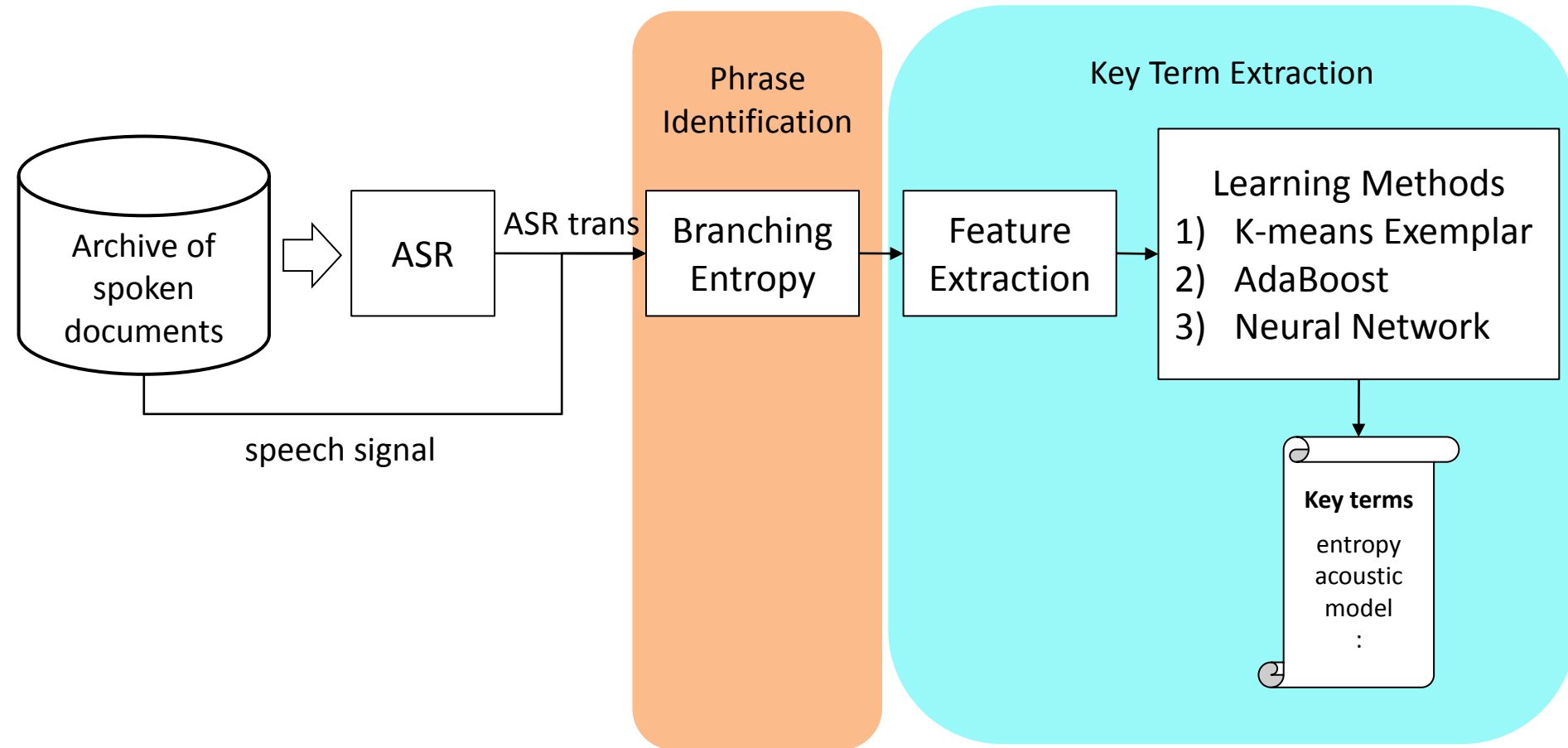
$$- \sum_{k=1}^K P(T_k|t_i) \log P(T_k|t_i)$$



Feature Name	Feature Description
LTP (I - III)	Latent Topic Probability (mean, variance, standard deviation)
LTS (I - III)	Latent Topic Significance (mean, variance, standard deviation)
LTE	term entropy for latent topic



Automatic Key Term Extraction



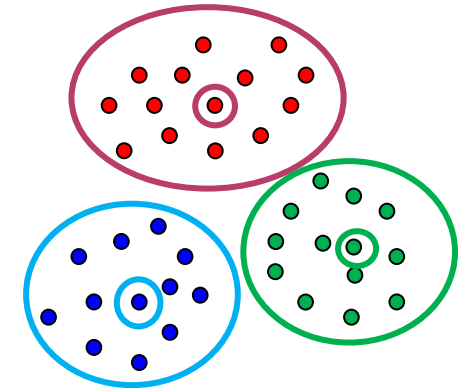
Using unsupervised and supervised approaches to extract key terms



Learning Methods

- Unsupervised learning
 - K-means Exemplar
 - Transform a term into a vector in LTS (Latent Topic Significance) space

$$v_i = (S_{t_i}(T_1), S_{t_i}(T_2), \dots, S_{t_i}(T_K))$$



- Run K-means

The terms in the same cluster focus on a single topic

- Find the centroid of each cluster to be the key term

The candidate term in the same group are related to the key term
The key term can represent this topic



Learning Methods

- Supervised learning
 - Adaptive Boosting
 - Neural Network

Automatically adjust the weights of features to produce a classifier



Experiments & Evaluation



Experiments

- Corpus
 - NTU lecture corpus
 - Mandarin Chinese embedded by English words



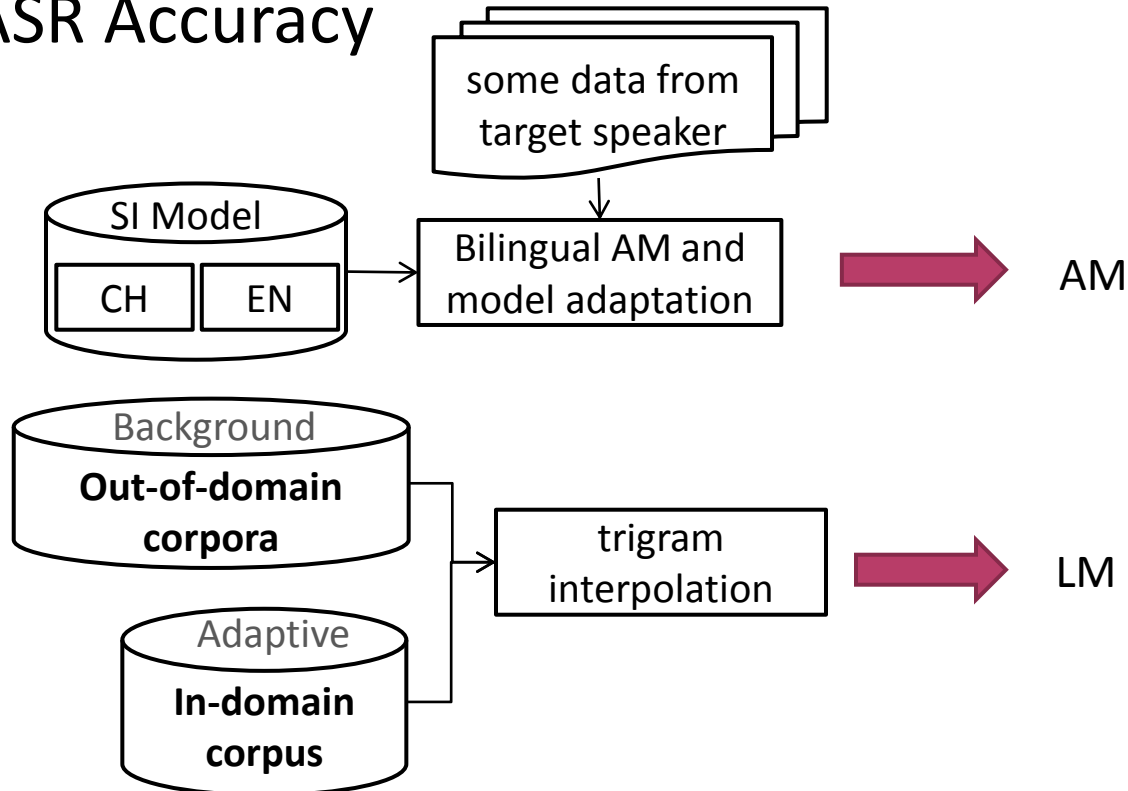
我們的solution是viterbi algorithm
(Our solution is viterbi algorithm)

- Single speaker
- 45.2 hours



Experiments

• ASR Accuracy



Language	Mandarin	English	Overall
Char Acc (%)	78.15	53.44	76.26



Experiments

- Reference Key Terms
 - Annotations from 61 students who have taken the course
 - If the k -th annotator labeled N_k key terms, he gave each of them a score of $\frac{1}{N_k}$, but 0 to others
 - Rank the terms by the sum of all scores given by all annotators for each term
 - Choose the top \bar{N} terms form the list (\bar{N} is average N_k)
 - $\bar{N} = 154$ key terms
 - 59 key phrases and 95 keywords



Experiments

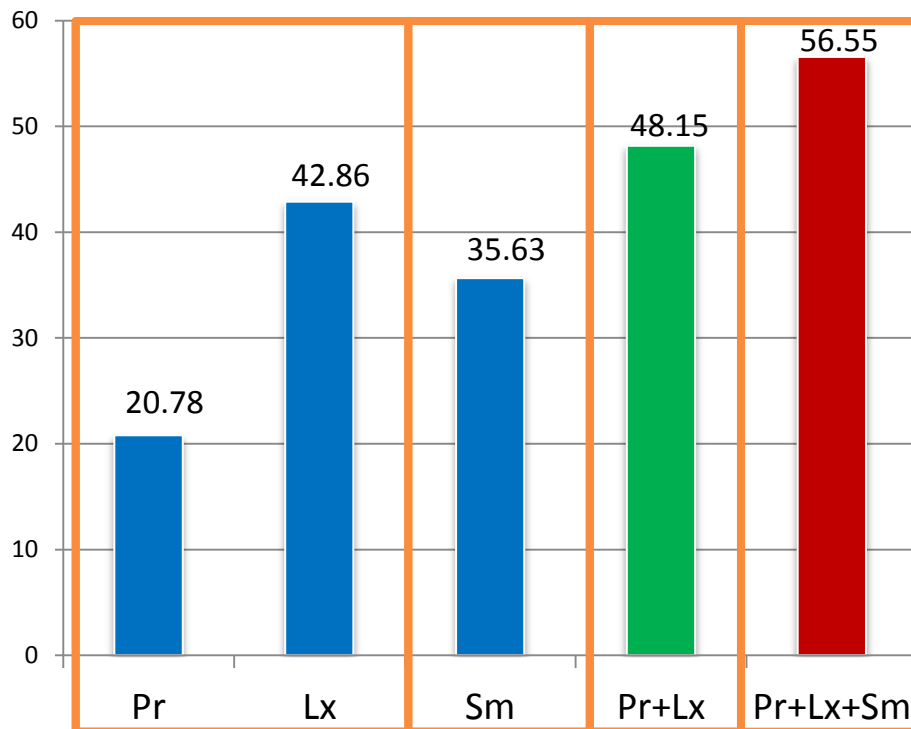
- Evaluation
 - Unsupervised learning
 - Set the number of key terms to be \bar{N}
 - Supervised learning
 - 3-fold cross validation



Experiments

- Feature Effectiveness
 - Neural network for keywords from ASR transcriptions

F-measure



Pr: Prosodic
Lx: Lexical
Sm: Semantic

Three sets of features are all useful



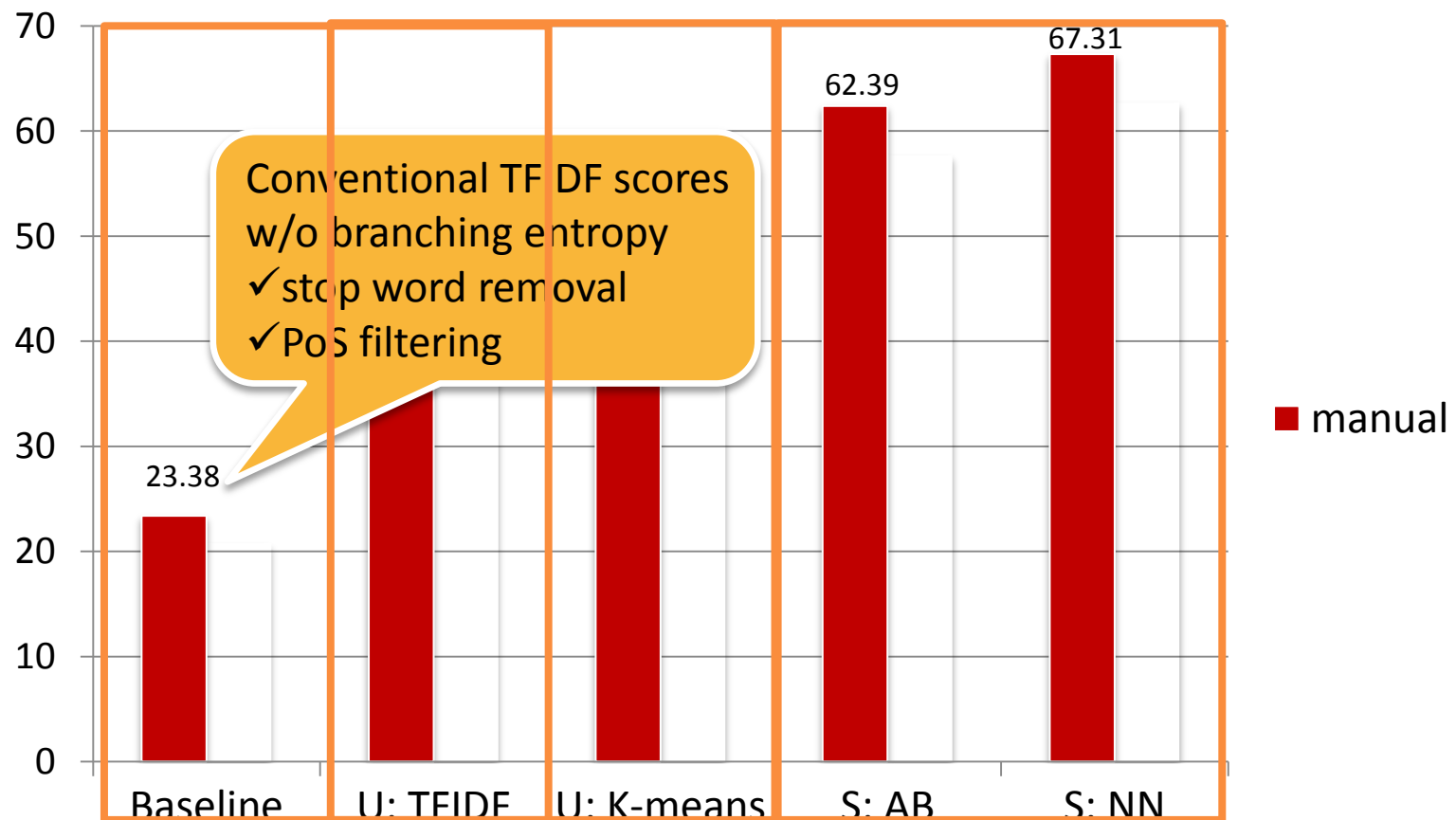
Experiments

Overall Performance

AB: AdaBoost

NN: Neural Network

F-measure



Supervised approaches are better than unsupervised approaches



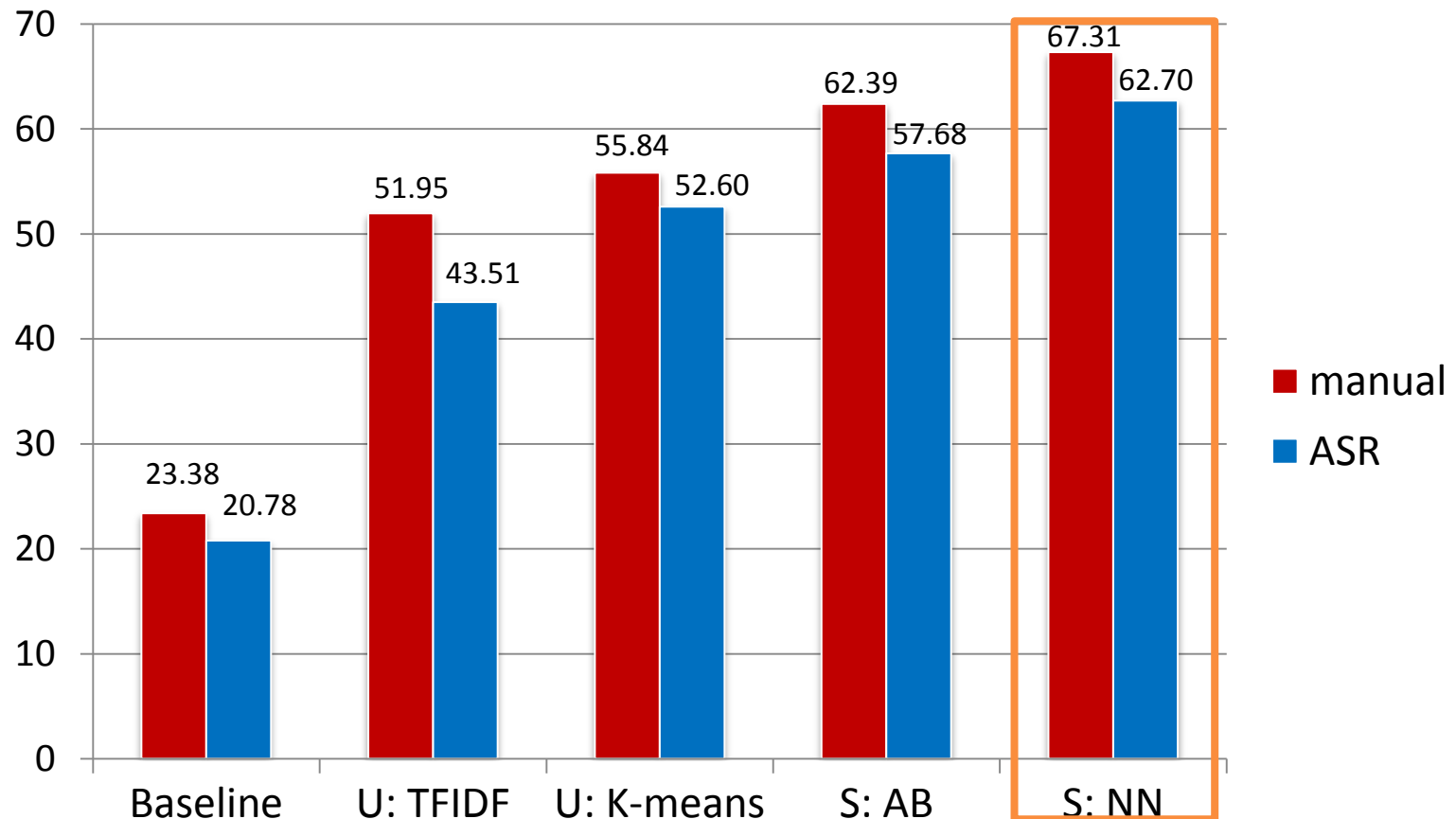
Experiments

Overall Performance

AB: AdaBoost

NN: Neural Network

F-measure



Supervised learning using neural network gives the best results



Conclusion



Conclusion

- We propose the new approach to extract key terms
- The performance can be improved by
 - Identifying phrases by branching entropy
 - Prosodic, lexical, and semantic features together
- The results are encouraging



Thanks for your attention! 😊
Q & A

Thank reviewers for valuable comments

NTU Virtual Instructor: <http://speech.ee.ntu.edu.tw/~RA/lecture>