11.0 Spoken Content Understanding, User-content Interaction and Beyond

2. “Multi-layered Summarization of Spoken Document Archives by Information Extraction and Semantic Structuring”, Interspeech 2006, Pittsburg, USA
User-Content Interaction for Spoken Content Retrieval

- **Problems**
  - Unlike text content, spoken content not easily summarized on screen, thus retrieved results difficult to scan and select
  - User-content interaction always important even for text content

- **Possible Approaches**
  - Automatic summary/title generation and key term extraction for spoken content
  - Semantic structuring for spoken content
  - Multi-modal dialogue with improved interaction
Multi-media/Spoken Document Understanding and Organization

- Key Term/Named Entity Extraction from Multi-media/Spoken Documents
  - personal names, organization names, location names, event names
  - key phrase/keywords in the documents
  - very often out-of-vocabulary (OOV) words, difficult for recognition

- Multi-media/Spoken Document Segmentation
  - automatically segmenting a multi-media/spoken document into short paragraphs, each with a central topic

- Information Extraction for Multi-media/Spoken Documents
  - extraction of key information such as who, when, where, what and how for the information described by multi-media/spoken documents.
  - very often the relationships among the key terms/named entities

- Summarization for Multi-media/Spoken Documents
  - automatically generating a summary (in text or speech form) for each short paragraph

- Title Generation for Multi-media/Spoken Documents
  - automatically generating a title (in text or speech form) for each short paragraph
  - very concise summary indicating the topic area

- Topic Analysis and Organization for Multi-media/Spoken Documents
  - analyzing the subject topics for the short paragraphs
  - clustering and organizing the subject topics of the short paragraphs, giving the relationships among them for easier access
Integration Relationships among the Involved Technology Areas

Key term extraction from spoken documents

Semantic analysis

Information indexing, retrieval and browsing
Key Term Extraction from Spoken Content (1/2)

- Key Terms: key phrases and keywords
- Key Phrase Boundary Detection
- An Example

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

- Left/right boundary of a key phrase detected by context statistics
• **Prosodic Features**
  – key terms probably produced with longer duration, wider pitch range and higher energy

• **Semantic Features (e.g. PLSA)**
  – key terms usually focused on smaller number of topics

• **Lexical Features**
  – TF/IDF, POS tag, etc.
Extractive Summarization of Spoken Documents

- Scoring sentences based on prosodic, semantic, lexical features and confidence measures, etc.
- Based on a given summarization ratio

• Selecting most representative utterances in the original document but avoiding redundancy

Document d:

Summary of document d:

- Correctly recognized word
- Wrongly recognized word
Title Generation for Spoken Documents

- Titles for retrieved documents/segments helpful in browsing and selection of retrieved results
- Short, readable, telling what the document/segment is about
- One example: Scored Viterbi Search
Multi-modal Dialogue

- An example: user-system interaction modeled as a Markov Decision Process (MDP)

- Example goals
  - small average number of dialogue turns (average number of user actions taken) for successful tasks (success: user’s information need satisfied)
  - less effort for user, better retrieval quality
Spoken Document Summarization

- Why summarization?
  - Huge quantities of information
  - Spoken content difficult to be shown on the screen and difficult to browse

- Broadcast News
- Meeting
- Books
- Lecture
- Websites
- Mails
- News articles
- Social Media
Spoken Document Summarization

- **More difficult than text summarization**
  - Recognition errors, Disfluency, etc.

- **Extra information not in text**
  - Prosody, speaker identity, emotion, etc.
Unsupervised Approach: Maximum Margin Relevance (MMR)

• Select **relevant** and **non-redundant** sentences

\[ MMR(x_i) = Rel(x_i) - \lambda Red(x_i, S) \]

Relevance : \( Rel(x_i) = Sim(x_i, d) \)

Redundancy : \( Red(x_i, S) = Sim(x_i, S) \)

Sim \((x_i, \bullet)\) : Similarity measure
Supervised Approach: SVM or Similar

- Trained with documents with human labeled summaries

Training data
- \( d_N: \) document
- \( d_2: \) document
- \( d_1: \) document
  - \( x_1, x_2, \ldots \)
- \( x_m: \) utterance

\( S_N: \) Summary

\( S_2: \) Summary
- \( S_1: \) Summary
  - \( s_1, s_2, \ldots \)
- \( s_i: \) selected utterance

Human labeled

Testing data
- \( \hat{d}_N: \) document
  - \( \hat{x}_1, \hat{x}_2, \ldots \)
- \( \hat{x}_m: \) utterance

\( v(\hat{x}_i): \) Feature vector of \( \hat{x}_i \)

Training phase

Testing phase

Binary classification problem:
\( x_i \in S, \) or \( x_i \notin S \)

Feature Extraction

Binary Classification model

Ranked utterances
Domain Adaptation of Supervised Approach

• Problem
  – Hard to get high quality training data
  – In most cases, we have labeled out-of-domain references but not labeled target domain references

• Goal
  – Taking advantage of out-of-domain data
Domain Adaptation of Supervised Approach

- *Model* \(_0\) trained by out-of-domain data, used to obtain *summary* \(_0\) for target domain

Out-of-domain data with labeled document/summary

Target domain data without labeled document/summary

Spoken Document

\(d_N\): document

\(d_2\): document

\(d_1\): document

\(x_1, x_2, \ldots\)

\(x_m\): utterance

Summary

Human labeled

\(S_N\): Summary

\(S_2\): Summary

\(S_1\): Summary

Summary

\(\tilde{S}_M\): Summary

\(\tilde{S}_2\): Summary

\(\tilde{S}_1\): Summary

Summary

\(\tilde{d}_M\): document

\(\tilde{d}_2\): document

\(\tilde{d}_1\): document

\(\tilde{x}_1, \tilde{x}_2, \ldots\)

\(\tilde{x}_m\): utterance

\(\tilde{S}_1\): Summary

\(\tilde{S}_2\): Summary

\(\tilde{S}_M\): Summary
Domain Adaptation of Supervised Approach

- Model$_0$ trained by out-of-domain data, used to obtain summary$_0$ for target domain
- summary$_0$ together with out-of-domain data jointly used to train Model$_1$

Out-of-domain data with labeled document/summary

Target domain data without labeled document/summary
Document Summarization

• **Extractive Summarization**  
  – select *sentences* in the document

• **Abstractive Summarization**  
  – Generate sentences describing the content of the document

**Example:**

- **Extractive:** 彰化 檢方 偵辦 芳苑 鄉公所 道路 排水 改善 工程 弊案 拘提 芳苑 鄉長 陳 聰明 檢方 認為 陳 聰明 等 人和 包商 勾結 涉嫌 貪污 和 圖利 罪嫌 凌晨 向 法院 聲請 羈押 以及 公所 秘書 楊 騰 煌 獲准

- **Abstractive:** 彰化 鄉公所 陳聰明 涉嫌 貪污
Document Summarization

• Extractive Summarization
  – select **sentences** in the document

• Abstractive Summarization
  – Generate sentences describing the content of the document

e.g.
Abstractive Summarization (1/4)

- An Example Approach
  1) Generating candidate sentences by a graph
  2) Selecting sentences by topic models, language models of words, parts-of-speech(POS), length constraint, etc.
Abstractive Summarization (2/4)

1) Generating Candidate sentences

- **Graph construction** + search on graph
  - Node: “word” in the sentence
  - Edge: word ordering in the sentence

- X1: 這個 飯店 房間 算 舒適.
- X2: 這個 飯店 的 房間 很 舒適 但 離 市中心 太遠 不方便
- X3: 飯店 挺 漂亮 但 房間 很 舊
- X4: 離 市中心 遠
1) Generating Candidate sentences

- **X1**: 這個 飯店 房間 算 舒適
- **X2**: 這個 飯店 的 房間 很 舒適 但 離 市中心 太遠 不方便
- **X3**: 飯店 挺 漂亮 但 房間 很 舊
- **X4**: 離 市中心 遠

**Graph construction** + search on graph
1) Generating Candidate sentences

Graph construction + search on graph

- X1: 這個 飯店 房間 算 舒適
- X2: 這個 飯店 的 房間 很 舒適 但 離 市中心 太遠 不方便
- X3: 飯店 挺 漂亮 但 房間 很 舊
- X4: 離 市中心 遠

Start node
Abstractive Summarization (3/4)

- 1) Generating Candidate sentences

□ X1: 這個 飯店 房間 算 舒適
□ X2: 這個 飯店 的 房間 很 舒適 但 離 市中心 太遠 不方便
□ X3: 飯店 挺 漂亮 但 房間 很 舊
□ X4: 離 市中心 遠

• Generating Candidate sentences

- Graph construction + search on graph

Diagram showing the graph construction process with start and end nodes.
Abstractive Summarization (4/4)

1) Generate Candidate sentences  Graph construction + search on graph

- Search: find Valid path on graph
- Valid path: path from start node to end node

Example sentences:

- X1: 這個 飯店 房間 算 舒適
- X2: 這個 飯店 的 房間 很 舒適 但 離 市中心 太遠 不方便
- X3: 飯店 挺 漂亮 但 房間 很 舊
- X4: 離 市中心 遠

Graph:

- Start node
- End node
Abstractive Summarization (4/4)

1) Generating Candidate sentences

Graph construction + search on graph

- Search: find Valid path on graph
- Valid path: path from start node to end node

Example sentences:
- X1: 這個 飯店 房間 算 舒適
- X2: 這個 飯店 的 房間 很 舒適 但 離 市中心 太遠 不方便
- X3: 飯店 挺 漂亮 但 房間 很 舊
- X4: 離 市中心 遠

Diagram:
- Start node
- End node
Sequence-to-Sequence Learning (1/3)

- Both input and output are sequences with different lengths.
  - machine translation (machine learning → 機器學習)
  - summarization, title generation
  - spoken dialogues
  - speech recognition

Containing all information about input sequence
sequence-to-sequence learning (2/3)

- Both input and output are sequences with different lengths.
  - machine translation (machine learning → 機器學習)
  - summarization, title generation
  - spoken dialogues
  - speech recognition

Don’t know when to stop
Sequence-to-Sequence Learning (3/3)

- Both input and output are sequences with different lengths.
  - machine translation (machine learning → 機器學習)
  - summarization, title generation
  - spoken dialogues
  - speech recognition

[Add a symbol “===“ (斷)]

[Ilya Sutskever, NIPS’14][Dzmitry Bahdanau, arXiv’15]
End-to-end Deep Learning for Speech Recognition

• Jointly Learn the Sound (Acoustic Models), Vocabulary (Lexicon) and Sentence Structure (Language Model)
  – rather than trained separately with different criteria

• One example

• A 70-year-old person has heard roughly no more than 0.6 million of hrs of voice in his life
  – machines can be trained with more than this quantity of data in very short time
Self-Attention

\[ a_1, a_2, a_3, a_4 \]

\[ b_1, b_2, b_3, b_4 \]

\[ b^i \text{ is obtained based on the whole input sequence.} \]

\[ b^1, b^2, b^3, b^4 \text{ can be parallelly computed.} \]

You can try to replace any thing that has been done by RNN with self-attention (e.g. easier for parallelization).
Self-attention

https://arxiv.org/abs/1706.03762

Attention is all you need.

\(q\): query (to match others)

\[ q^i = W^q a^i \]

\(k\): key (to be matched)

\[ k^i = W^k a^i \]

\(v\): information to be extracted

\[ v^i = W^v a^i \]
**Self-attention**

拿每個 query q 去對每個 key k 做 attention

Scaled Dot-Product Attention: \( \alpha_{1,i} = \frac{q^1 \cdot k^i}{\sqrt{d}} \)

d is the dim of q and k

dot product

拿每個 query q 去對每個 key k 做 attention
Self-attention

\[
\hat{\alpha}_{1,i} = \frac{\exp(\alpha_{1,i})}{\sum_j \exp(\alpha_{1,j})}
\]
Self-attention

Considering the whole sequence

\[ b^1 = \sum_i \hat{\alpha}_{1,i} v^i \]
Self-attention

拿每個 query q 去對每個 key k 做 attention

\[ b^2 = \sum_i \hat{\alpha}_{2,i} v^i \]
Self-attention

\( b^1, b^2, b^3, b^4 \) can be parallelly computed.
**Multi-head Self-attention**

(2 heads as example)

\[ q_{i,1} = W^{q,1} q^i \]
\[ q_{i,2} = W^{q,2} q^i \]

\[ q^i = W^q a^i \]
**Multi-head Self-attention**

(2 heads as example)

\[
q^{i,1} = W^{q,1} q^i \\
q^{i,2} = W^{q,2} q^i
\]

\[
q^i = W^q x^i \\
a^i
\]

\[
b^{i,1} \\
b^{i,2}
\]

\[
k^{i,1} \quad k^{i,2} \quad v^{i,1} \quad v^{i,2}
\]

\[
q^{j,1} \quad q^{j,2} \quad k^{j,1} \quad k^{j,2} \quad v^{j,1} \quad v^{j,2}
\]
Multi-head Self-attention

\[ b^i = W^O b^{i,1} b^{i,2} \]

(2 heads as example)

\[ b^i = W^O b^{i,1} b^{i,2} \]

\[ b^i = W^O b^{i,1} b^{i,2} \]

\[ b^i = W^O b^{i,1} b^{i,2} \]

\[ b^i = W^O b^{i,1} b^{i,2} \]

\[ b^i = W^O b^{i,1} b^{i,2} \]

\[ b^i = W^O b^{i,1} b^{i,2} \]

\[ b^i = W^O b^{i,1} b^{i,2} \]

\[ b^i = W^O b^{i,1} b^{i,2} \]

\[ b^i = W^O b^{i,1} b^{i,2} \]

\[ b^i = W^O b^{i,1} b^{i,2} \]

\[ b^i = W^O b^{i,1} b^{i,2} \]

\[ b^i = W^O b^{i,1} b^{i,2} \]

\[ b^i = W^O b^{i,1} b^{i,2} \]

\[ b^i = W^O b^{i,1} b^{i,2} \]

\[ b^i = W^O b^{i,1} b^{i,2} \]

\[ b^i = W^O b^{i,1} b^{i,2} \]

\[ b^i = W^O b^{i,1} b^{i,2} \]

\[ b^i = W^O b^{i,1} b^{i,2} \]

\[ b^i = W^O b^{i,1} b^{i,2} \]

\[ b^i = W^O b^{i,1} b^{i,2} \]

\[ b^i = W^O b^{i,1} b^{i,2} \]

\[ b^i = W^O b^{i,1} b^{i,2} \]

\[ b^i = W^O b^{i,1} b^{i,2} \]

\[ b^i = W^O b^{i,1} b^{i,2} \]

\[ b^i = W^O b^{i,1} b^{i,2} \]

\[ b^i = W^O b^{i,1} b^{i,2} \]

\[ b^i = W^O b^{i,1} b^{i,2} \]

\[ b^i = W^O b^{i,1} b^{i,2} \]

\[ b^i = W^O b^{i,1} b^{i,2} \]

\[ b^i = W^O b^{i,1} b^{i,2} \]

\[ b^i = W^O b^{i,1} b^{i,2} \]

\[ b^i = W^O b^{i,1} b^{i,2} \]

\[ b^i = W^O b^{i,1} b^{i,2} \]

\[ b^i = W^O b^{i,1} b^{i,2} \]

\[ b^i = W^O b^{i,1} b^{i,2} \]

\[ b^i = W^O b^{i,1} b^{i,2} \]

\[ b^i = W^O b^{i,1} b^{i,2} \]

\[ b^i = W^O b^{i,1} b^{i,2} \]

\[ b^i = W^O b^{i,1} b^{i,2} \]

\[ b^i = W^O b^{i,1} b^{i,2} \]
Positional Encoding

• No position information in self-attention.
• Original paper: each position has a unique positional vector $e^i$ (not learned from data)
• In other words: each $x^i$ appends a one-hot vector $p^i$

$$p^i = \begin{bmatrix} \vdots & \vdots & \vdots \\ 0 & 1 & 0 \\ \vdots & \vdots & \vdots \end{bmatrix}$$

$$x^i = \begin{bmatrix} W^I \\ W^P \end{bmatrix} + \begin{bmatrix} p^i \\ e^i \end{bmatrix}$$
Seq2seq with Attention

Review: https://www.youtube.com/watch?v=ZjfjPzXw6og&feature=youtu.be
Using Chinese to English translation as example
Transformer

Layer Norm: https://arxiv.org/abs/1607.06450

Batch Norm: https://www.youtube.com/watch?v=BZh1litr5Rkg

Batch Size

\[ \mu = 0, \sigma = 1 \]

Layer

\[ \mu = 0, \sigma = 1 \]

Batch

Masked: attend on the generated sequence

attend on the input sequence
Word Vector Representations (Word Embedding)

1-of-N encoding of the word \( w_{i-1} \)

- Use the input of the neurons in the first layer to represent a word \( w \)
- Word vector, word embedding feature: \( V(w) \)
- Word analogy task: \( \text{(king)} - \text{(man)} + \text{(woman)} \rightarrow \text{(queen)} \)

The probability for each word as the next word \( w_i \)

(P.55 of 9.0)
Contextualized Word Embedding

• Each word token has its own embedding (even though it has the same word type)
• The embeddings of word tokens also depend on its context.

...money in the bank...

...the river bank...

...own blood bank...
Embeddings from Language Model (ELMO)  


- RNN-based language models (trained from lots of sentences)

E.g. given “潮水 退了 就 知道 誰 沒穿 褲子”
Embeddings from Language Model (ELMO)  

• RNN-based language models (trained from lots of sentences)
  
e.g. given “潮水 退了 就 知道 誰 沒穿 褲子”
Each layer in deep LSTM can generate a latent representation.

Which one should we use???
ELMO

Learned with the downstream tasks

潮水 退了 就 知道 ……
Bidirectional Encoder Representations from Transformers (BERT)

- BERT = Encoder of Transformer

Learned from a large amount of text without annotation

Although I use “word” as unit here, “character” may be a better choice for Chinese.
Training of BERT

• Approach 1: Masked LM

如果兩個詞彙填在同一个地方沒有違和感，那它們就有類似的 embedding

Predicting the masked word

Linear Multi-class Classifier

vocabulary size
Training of BERT

Approach 2: Next Sentence Prediction

[SEP]: the boundary of two sentences

[CLS]: the position that outputs classification results

Linear Binary Classifier

yes
How to use BERT – Case 2

Input: single sentence, output: class of each word

Example: Slot filling

sentence

BERT

[CLS]

w₁ w₂ w₃

Linear Cls

Linear Cls

Linear Cls

class

class

class
Generative Pre-Training (GPT)

Transformer
Decoder

BERT
(340M)

ELMO
(94M)

GPT-2
(1542M)

Source of image: https://huaban.com/pins/1714071707/
Generative Pre-Training (GPT)

Many Layers...

\[
\begin{align*}
\alpha_{2,1} & \quad q^1 & k^1 & v^1 \\
\alpha_{2,2} & \quad q^2 & k^2 & v^2 \\
\end{align*}
\]
Generative Pre-Training (GPT)

Many Layers...

<<BOS>潮水退了就\alpha_{3,1}, \alpha_{3,2}, \alpha_{3,3}\>

\begin{align*}
q^1 & \rightarrow k^1 \rightarrow v^1 \\
a^1 & \rightarrow q^2 \rightarrow k^2 \rightarrow v^2 \\
a^2 & \rightarrow k^3 \\
q^3 & \rightarrow k^3 \rightarrow v^3 \\
a^3 & \rightarrow q^4 \rightarrow k^4 \rightarrow v^4 \\
a^4 & \rightarrow \end{align*}
Interactive dialogue: retrieval engine interacts with the user to find out more precisely his information need

- User entering the query
- When the retrieved results are divergent, the system may ask for more information rather than offering the results
Interactive dialogue: retrieval engine interacts with the user to find out more precisely his information need

- User entering the second query
- when the retrieved results are still divergent, but seem to have a major trend, the system may use a key word representing the major trend asking for confirmation
- User may reply: “Yes” or “No, Asia”
Markov Decision Process (MDP)

- **A mathematical framework for decision making, defined by (S,A,T,R,π)**
  - **S**: Set of states, current system status
    \[ \{s_1, s_2, s_3, \ldots\} \]
  - **A**: Set of actions the system can take at each state
    \[ \{A_1, A_2, A_3, \ldots\} \]
  - **T**: transition probabilities between states when a certain action is taken
  - **R**: reward received when taking an action
    \[ \{R_1, R_2, R_3, \ldots\} \]
  - **π**: policy, choice of action given the state
    \[ \{\pi: s_i \rightarrow A_j\} \]

- **Objective**: Find a policy that maximizes the expected total reward
Model as **Markov Decision Process (MDP)**

- After a query entered, the system starts at a certain state
- States: retrieval result quality estimated as a continuous variable (e.g. MAP) plus the present dialogue turn
- Action: at each state, there is a set of actions which can be taken: asking for more information, returning a keyword or a document, or a list of keywords or documents asking for selecting one, or showing results….
- User response corresponds to a certain negative reward (extra work for user)
- when the system decides to show to the user the retrieved results, it earns some positive reward (e.g. MAP improvement)
- Learn a policy maximizing rewards from historical user interactions (\( \pi: S_i \rightarrow A_j \))
Reinforcement Learning

• **Example approach: Value Iteration**
  
  - Define value function: \( Q^\pi : S \times A \rightarrow \mathbb{R} \)
    \[
    Q^\pi(s, a) = E[\sum_{k=0}^{\infty} \gamma^k r_k | s_0 = s, a_0 = a]
    \]
    The expected discounted sum of rewards given \( \pi \) started from \((s, a)\)

  - The real value of \( Q \) can be estimated iteratively from a training set:
    \[
    Q^*(s, a) = E_{s' | s, a}[R(s, a, s') + \gamma^\max_{b \in A} Q^*(s', b)]
    \]
    \( Q^*(s, a) \): estimated value function based on the training set

  - Optimal policy is learned by choosing the best action given each state such that the value function is maximized
Question-Answering (QA) in Speech

- **Question, Answer, Knowledge Source** can all be in text form or in Speech
- **Spoken Question Answering** becomes important
  - spoken questions and answers are attractive
  - the availability of large number of on-line courses and shared videos today makes spoken answers by distinguished instructors or speakers more feasible, etc.
- **Text Knowledge Source** is always important
Three Types of QA

- **Factoid QA:**
  - What is the name of the largest city of Taiwan? Ans: Taipei.

- **Definitional QA:**
  - What is QA?

- **Complex Question:**
  - How to construct a QA system?
Factoid QA

• **Question Processing**
  – Query Formulation: transform the question into a query for retrieval
  – Answer Type Detection (city name, number, time, etc.)

• **Passage Retrieval**
  – Document Retrieval, Passage Retrieval

• **Answer Processing**
  – Find and rank candidate answers
Factoid QA – Question Processing

• **Query Formulation:** Choose key terms from the question
  – Ex: What is the name of the largest city of Taiwan?
  – “Taiwan”, “largest city” are key terms and used as query

• **Answer Type Detection**
  – “city name” for example
  – Large number of hierarchical classes hand-crafted or automatically learned
An Example Factoid QA

- Watson: a QA system developed by IBM (text-based, no speech), who won “Jeopardy!”
More about QA

• **Definitional QA ≈ Query-focused summarization**
  – Use similar framework as Factoid QA: Question Processing, Passage Retrieval, Answer Processing is replaced by Summarization

• **QA based on Spoken content**
  – Spoken QA

• **QA based on Deep Learning**
  – e.g. BERT
What can Spoken Content Retrieval and the Related Technologies do for us?

- Google reads all text over the Internet
  - can find any text over the Internet for the user

- All Roles of Text can be realized by Voice

- Machines can listen to all voices over the Internet
  - can find any utterances over the Internet for the user

- A Spoken Version of Google
What can we do with a Spoken Version of Google?

• Multimedia Content exponentially increasing over the Internet
  
  300hrs of videos uploaded per min (2015.01)

Roughly 2000 online courses on Coursera (2016.04)

  – best archive of global human knowledge is here
  – desired information deeply buried under huge quantities of unrelated information

• Nobody can go through so much multimedia information, but Machines can

• Machines may be able to listen to and understand the entire multimedia knowledge archive over the Internet
  – extracting desired information for each individual user
A Target Application Example : Personalized Education Environment

- For each individual user
  - I wish to learn about Wolfgang Amadeus Mozart and his music
  - I can spend 3 hrs to learn

This is the 3-hr personalized course for you. I’ll be your personalized teaching assistant. Ask me when you have questions.

- Understanding, Summarization and Question Answering for Spoken Content
  - something we could Never do (even today)
  - semantic analysis for spoken content
Constructing the Semantic Structures of the Spoken Content

Example Approach 1: Spoken Content categorized by Topics and organized in a Two-dimensional Tree Structure (2005)

- each category labeled by a set of key terms (topic) located on a map
- categories nearby on the map are more related semantically
- each category expanded into another map in the next layer

[Eurospeech 2005]
### An Example of Two-dimensional Trees

- **Broadcast News Browser (2006)**

![Diagram of a two-dimensional tree browser for broadcast news](image)

<table>
<thead>
<tr>
<th>Top-Down Browsing</th>
<th>Powder Brazil</th>
<th>Aerobus On the plane</th>
<th>Fisherboat Crewman</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Suspect</td>
<td>Airplane Crash</td>
<td>Waters</td>
</tr>
<tr>
<td></td>
<td>Murder</td>
<td>This plane Landing</td>
<td>Submersible</td>
</tr>
<tr>
<td></td>
<td>Woman</td>
<td></td>
<td>Somalia</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>North Korea</td>
</tr>
<tr>
<td></td>
<td>Killed by accident</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Suicide</th>
<th>Volcano</th>
<th>Teenager</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jordanian</td>
<td>Tsunami</td>
<td>Paris</td>
</tr>
<tr>
<td>Israel</td>
<td>Earthquake center</td>
<td>France</td>
</tr>
<tr>
<td>Baghdad</td>
<td>Scale</td>
<td>Suburb</td>
</tr>
<tr>
<td>Iraq</td>
<td>Miyagi</td>
<td>Curfew</td>
</tr>
<tr>
<td>Bomb</td>
<td>Earthquake</td>
<td>Rebellion</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Strike</th>
<th>Deluge</th>
<th>Conflagration</th>
</tr>
</thead>
<tbody>
<tr>
<td>New York City</td>
<td>Hurricane</td>
<td>Blaze</td>
</tr>
<tr>
<td>New York</td>
<td>Blizzard</td>
<td>Fire condition</td>
</tr>
<tr>
<td>Transport</td>
<td>Blast</td>
<td>Miner</td>
</tr>
<tr>
<td>Industrial union</td>
<td>Typhoon</td>
<td>Traffic accident</td>
</tr>
<tr>
<td>Community</td>
<td>Kyushu</td>
<td>Coal mine</td>
</tr>
</tbody>
</table>

[Interspeech 2006]
Online Courses

• Sequential knowledge transfer lecture by lecture
• When a lecture in an online course is retrieved for a user
  – difficult for the user to understand this lecture without
    listening to previous related lectures
  – not easy to find out background or related knowledge
Example Approach 2: Key Term Graph (2009)
- each spoken slide labeled by a set of key terms (topics)
- relationships between key terms represented by a graph

Very Similar to Knowledge Graph

[ICASSP 2009][IEEE Trans ASL 2014]
An Example of Retrieving with an Online Course Browser (1/2)

- **Course**: Digital Speech Processing (2009)
  - Query: “triphone”
  - retrieved utterances shown with the spoken slides they belong to specified by the titles and key terms

---

**About 163 Results For Term "triphone"**

1. 5.01 sec. in 0:10:23.01
   in 5-7 Classification And Regression Trees (CART)
   *(Transcription: ... 那底下我們是要用它來做 triphone 的 train triphone ...)*
   Key Terms Related To This Slide: cart, classification and regression trees, entropy, machine learning, pattern recognition, triphone

2. 8.45 sec. in 0:24:26.33
   in 5-8 Splitting Criteria For The Decision Tree
   *(Transcription: ... 那我們現在可以來看我們現在怎麼做 TRI PHONE 那麼要做 TRI PHONE 的時候呢 ...)*
   Key Terms Related To This Slide: cross entropy, delta, entropy, k l distance, triphone

3. 8.69 sec. in 0:21:01.48
   in 5-10 Decision Tree Approach Extended To Different Context Dependent Unit
   *(Transcription: ... 那麼做 tri phone 最大的問題就是有一堆 unseen event 我們說過就是因為有很多個 unseen 的 tri phone ...)*
   Key Terms Related To This Slide: backward algorithm, co articulation, entropy, forward backward algorithm, gaussian, gaussian mixture, hmm, h t k, hidden markov model, information theory, k means, markov model, phoneme, segmental k means, silence, triphone

---

[ICASSP 2009][IEEE Trans ASL 2014]
An Example of Retrieving with an Online Course Browser (2/2)

- User clicks to view the spoken slide (2009)
  - including a summary, key terms and related key terms from the graph
  - recommended learning path for a specific key term

5-7 Classification And Regression Trees (CART)

This key term (entropy) first appears in 5-4
Also appears in slide(s): 5-5 5-6 5-7 5-8 5-9 5-10 6-1 6-2 6-5 6-10 9-5 12-1 12-8 13-6

[ICASSP 2009][IEEE Trans ASL 2014]
A Huge Number of Online Courses

- A user enters a keyword or a key phrase to coursera.

You searched for: **Machine Learning**. 752 matches

### Availability
- This Month: 360
- Pre-Enroll: 102
- Self Paced: 88

### All Topics
- Business: 252
- Computer Science: 143
- Social Sciences: 143

### Course Languages
- English: 739
- Chinese (Simplified): 6
- French: 2

---

National Taiwan University

### 機器學習基石 (Machine Learning Foundations)

Machine Learning Capstone: An Intelligent Application with Deep Learning

University of Washington

### 機器學習技法 (Machine Learning Techniques)

Robotics: Estimation and Learning
Having Machines Listen to all the Online Courses

three courses on some similar topic

Lectures with very similar content

[Interspeech 2015]
Having Machines Listen to all the Online Courses

three courses on some similar topic

sequential order for learning (prerequisite conditions)
Question Answering in the Era of Deep Learning

- Machine answering questions from the user

Knowledge source → Question Answering → Answer

- Question
- Knowledge source
- Search engine
- Unstructured documents
- Spoken content
Text v.s. Spoken QA (Cascading v.s. End-to-end)

- **Text QA**
  - Text retrieved
  - Question Answering
  - Answer

- **Spoken QA**
  - Spoken content retrieved
  - Speech Recognition (ASR)
  - Question Answering
  - Answer
  - Errors
  - Cascading
  - End-to-end Spoken Question Answering

[Interspeech 2020]
Audio-and-Text Jointly Learned SpeechBERT

- **Pre-training**
  - Reconstruction

- **Fine-tuning**
  - Start/End Position

- **End-to-end Globally Optimized for Overall QA Performance**
  - not limited by ASR errors (no ASR here)
  - extracting semantics directly from speech, not from words via ASR

[Interspeech 2020]
References

• Key terms

• Title Generation
  – “Abstractive Headline Generation for Spoken Content by Attentive Recurrent Neural Networks with ASR Error Modeling” IEEE Workshop on Spoken Language Technology (SLT), San Diego, California, USA, Dec 2016, pp. 151-157.
• Summarization


  – "Spoken Lecture Summarization by Random Walk over a Graph Constructed with Automatically Extracted Key Terms," InterSpeech 2011
References

• Summarization
  – “The Use of MMR, diversity-based reranking for reordering document and producing summaries” SIGIR, 1998
  – “Using Corpus and Knowledge-based Similarity Measure in Maximum Marginal Relevance for Meeting Summarization” ICASSP, 2008
References

- **Interactive Retrieval**
  - Partially observable Markov decision processes for spoken dialog systems, Jason D. Williams and Steve Young, Computer Speech and Language, 2007.
Reference

• Question Answering
  – “Hierarchical Attention Model for Improved Comprehension of Spoken Content”, IEEE Workshop on Spoken Language Technology (SLT), San Diego, California, USA, Dec 2016, pp. 234-238.
Reference

• Sequence-to-sequence Learning and End-to-end Speech Recognition
  – “Sequence to Sequence Learning with Neural Networks”, NIPS, 2014
  – “Listen, Attend and Spell: A Neural Network for Large Vocabulary Conversational Speech Recognition”, ICASSP 2016
Reference

• **Sequence-to-sequence Learning and End-to-end Speech Recognition**

• **Self-attention, Transformer, BERT, GPT**
  – [https://www.youtube.com/watch?v=UYPa347-DdE](https://www.youtube.com/watch?v=UYPa347-DdE)
Reference

• On-line Course Organization
  – “Structuring Lectures in Massive Open Online Courses (MOOCs) for Efficient Learning by Linking Similar Sections and Predicting Prerequisites”, Interspeech, Dresden, Germany, Sept 2015, pp. 1363-1367.