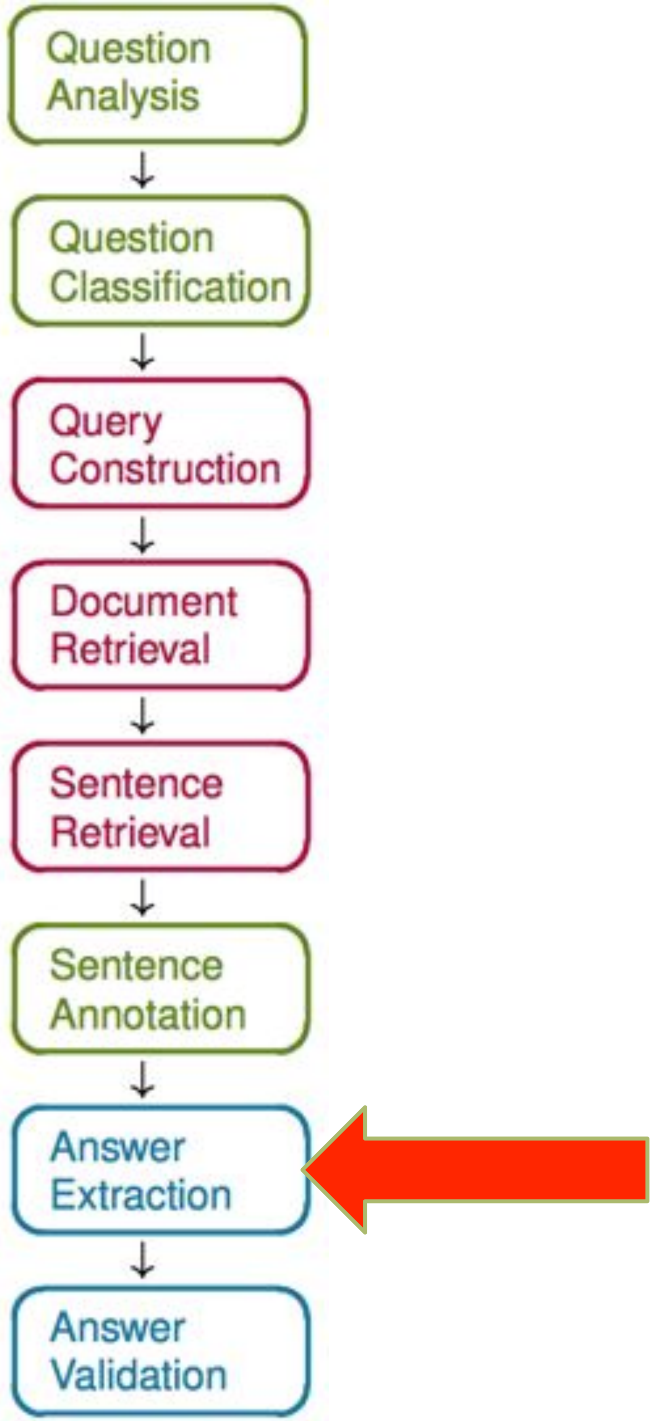
Answer Extraction

based on maximum entropy model

Minh Tuan Nguyen


Question Analysis
Question Classification
Query Construction
Document Retrieval
Sentence Retrieval
Sentence Annotation
Answer Extraction
Answer Validation

Answer Extraction

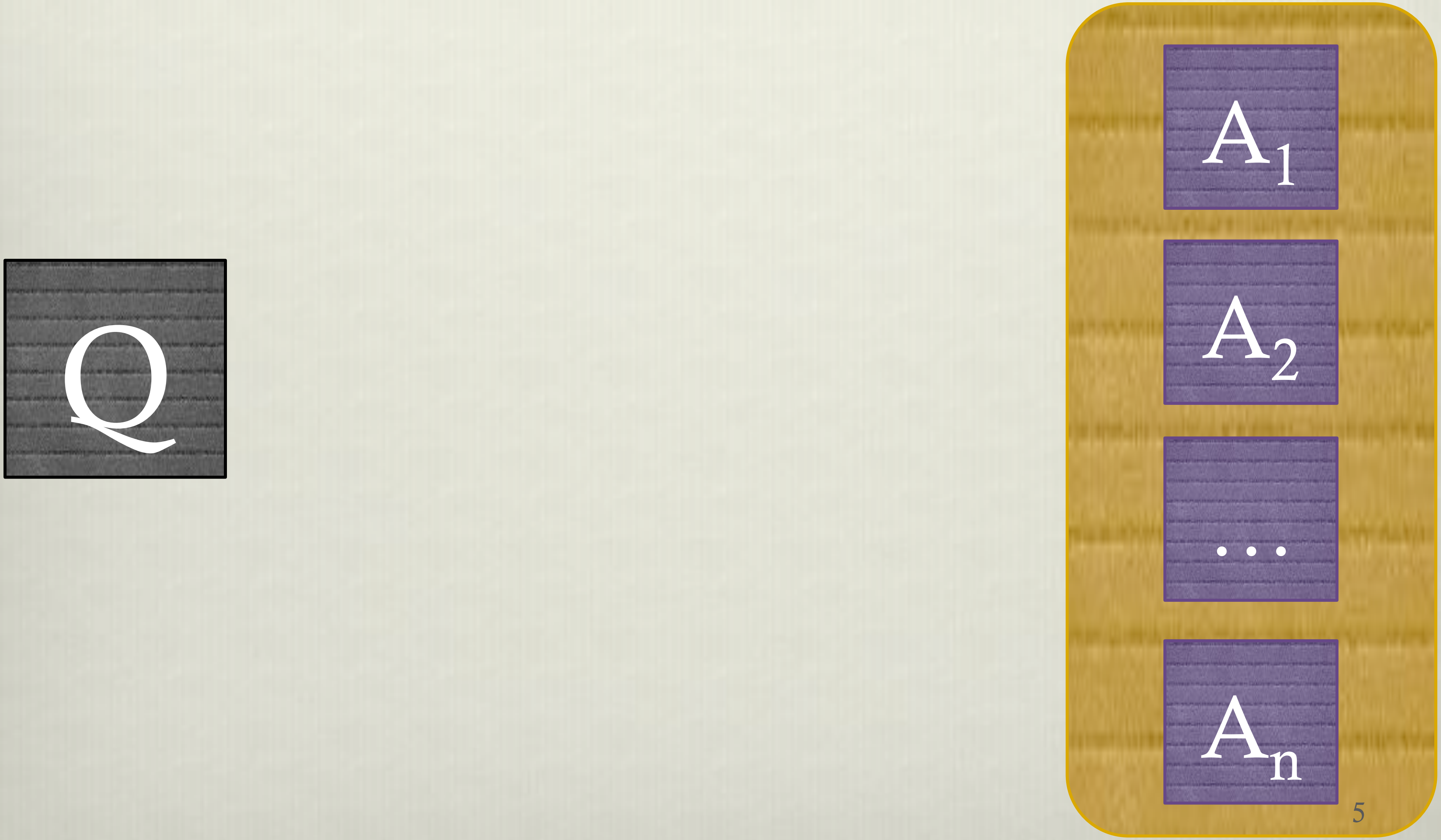
- Given
  - A question
  - A set of answer candidates

Answer Extraction

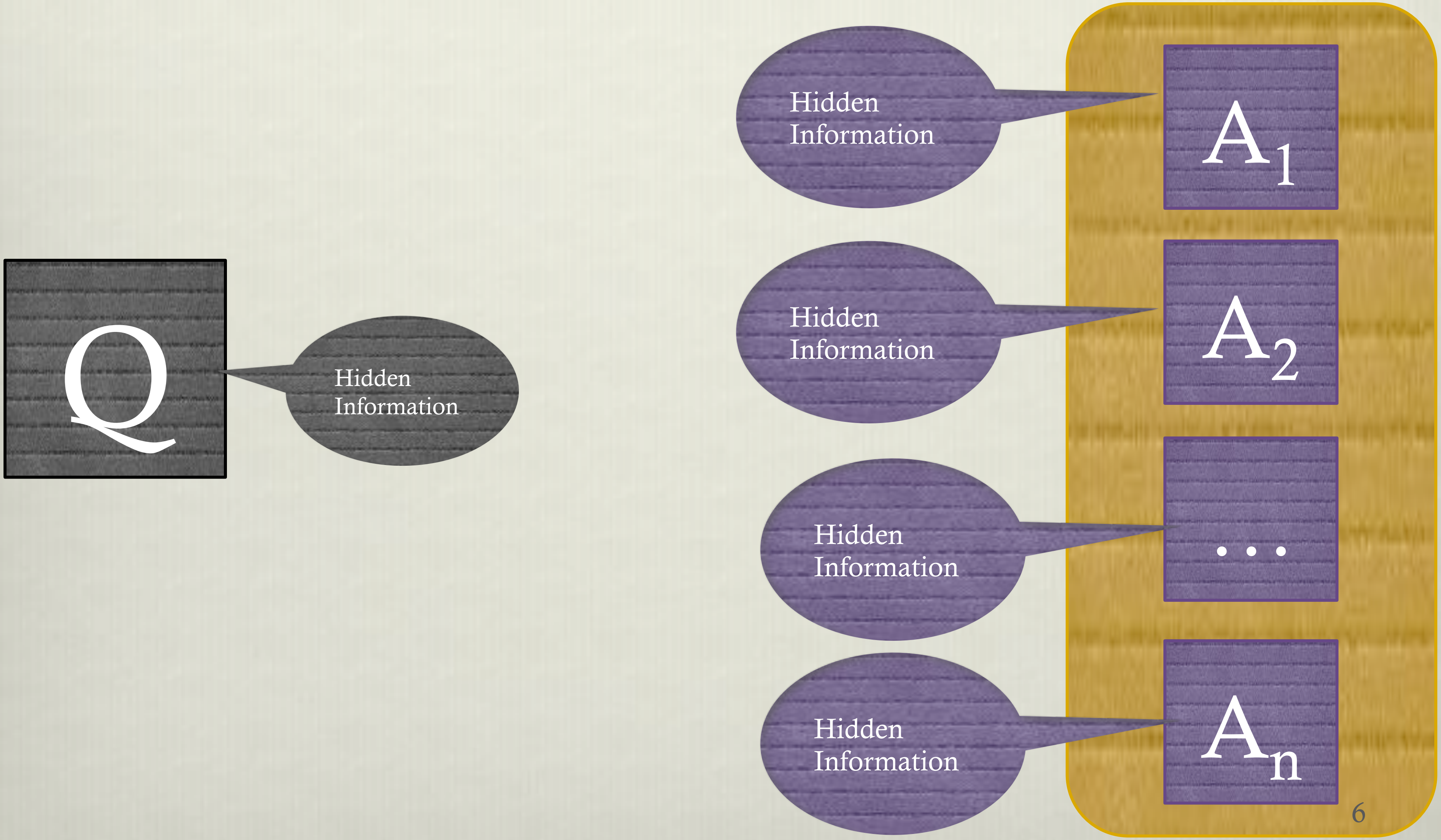
- Given
  - A question
  - A set of answer candidates

- Task
  - Select the correct answer from the set of answer candidates

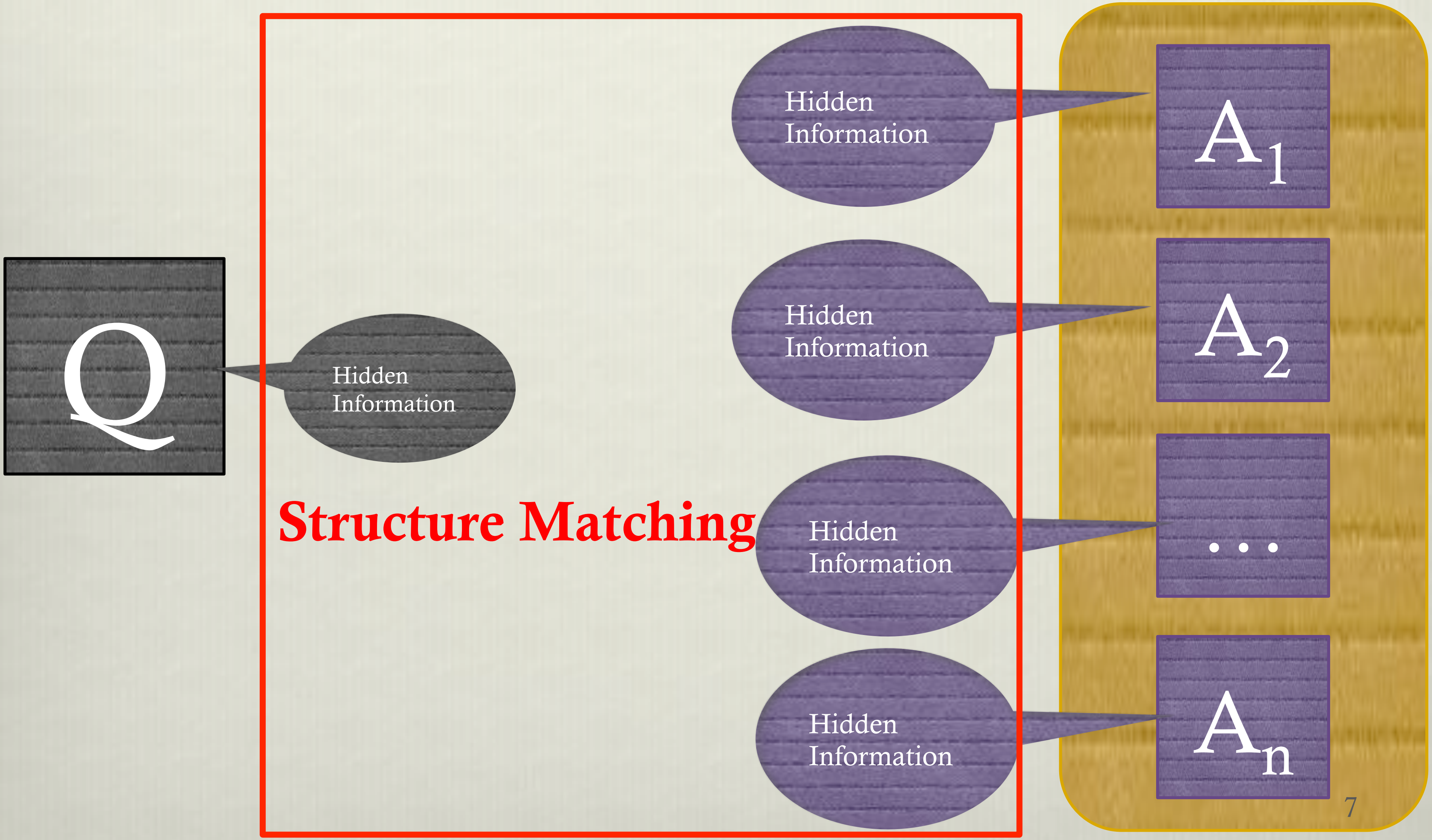
Answer Extraction



Answer Extraction



Answer Extraction



Example

- Question: *In what country was Albert Einstein born?*

Example

- Question: *In what country was Albert Einstein born?*
- Answer candidates :

  A: *Albert Einstein was born on 14 March 1879.*

  B: *Albert Einstein was born in Germany.*

  C: *Albert Einstein was born in a Jewish family.*

Example

- Question: *In what country was Albert Einstein born?*
- Answer candidates :

  A: *Albert Einstein was born on 14 March 1879.*

  B: *Albert Einstein was born in Germany.*

  C*: Albert Einstein was born in a Jewish family.*

→ Pattern 1: **X born in Y**

→ Pattern 2: **Location = Country**

Example

Question: *In what country was Albert Einstein born*

A: *Albert Einstein was born on 14 March 1879.*

B: *Albert Einstein was born in Germany.*

C: *Albert Einstein was born in a Jewish family.*

	X born in Y	Location = Country
Question		

	X born in Y	Location = Country
Answer A		
Answer B		
Answer C		

Example

Question: *In what country was Albert Einstein born*

A: *Albert Einstein was born on 14 March 1879.*

B: *Albert Einstein was born in Germany.*

C: *Albert Einstein was born in a Jewish family.*

	X born in Y	Location = Country
Question	YES	YES

	X born in Y	Location = Country
Answer A	NO	NO
Answer B	YES	YES
Answer C	YES	NO

Example

Question: *In what country was Albert Einstein born*

A: *Albert Einstein was born on 14 March 1879.*

B: *Albert Einstein was born in Germany.*

C: *Albert Einstein was born in a Jewish family.*

	X born in Y	Location = Country
Question	**YES**	**YES**

	X born in Y	Location = Country
Answer A	NO	NO
Answer B	**YES**	**YES**
Answer C	YES	NO

Answer Extraction

	Pattern 1	Pattern 2	…	Pattern n
Answer 1				
Answer 2				
Answer 3				
…				
Answer N				

How to model the pattern?

Feature function

- Feature is a binary-valued function:

$$f_j : X \times Y \rightarrow \{0, 1\}$$

X: space of contexts

Y: the set of classifier

Feature function

- Feature is a binary-valued function:

$$f_j : X \times Y \rightarrow \{0, 1\}$$

X: space of contexts

Y: the set of classifier

- Like a regular expression
  - 1 : match the specific pattern
  - 0 : don't match

Feature function example

- Question: Which name is female?

  A: Thomas　　B: Kevin

  C: Franz　　D: Bella

  →Pattern: **Name ends with vowel**

Feature function example

- Question: Which name is female?

  A: Thomas B: Kevin

  C: Franz D: Bella

  →Pattern: **Name ends with vowel**

$$f_{male}(x,y)=\begin{cases} 1 & \text{if } y \; is \; "male", \text{ last letter of } x \text{ is a vowel} \\ 0 & \text{otherwise} \end{cases}$$

$$f_{female}(x,y)=\begin{cases} 1 & \text{if } y \; is \; "female", \text{ last letter of } x \text{ is a vowel} \\ 0 & \text{otherwise} \end{cases}$$

Feature function example

- Question: Which name is female?

    A: Thomas   B: Kevin

    C: Franz   D: Bella

    →Pattern: **Name ends with vowel**

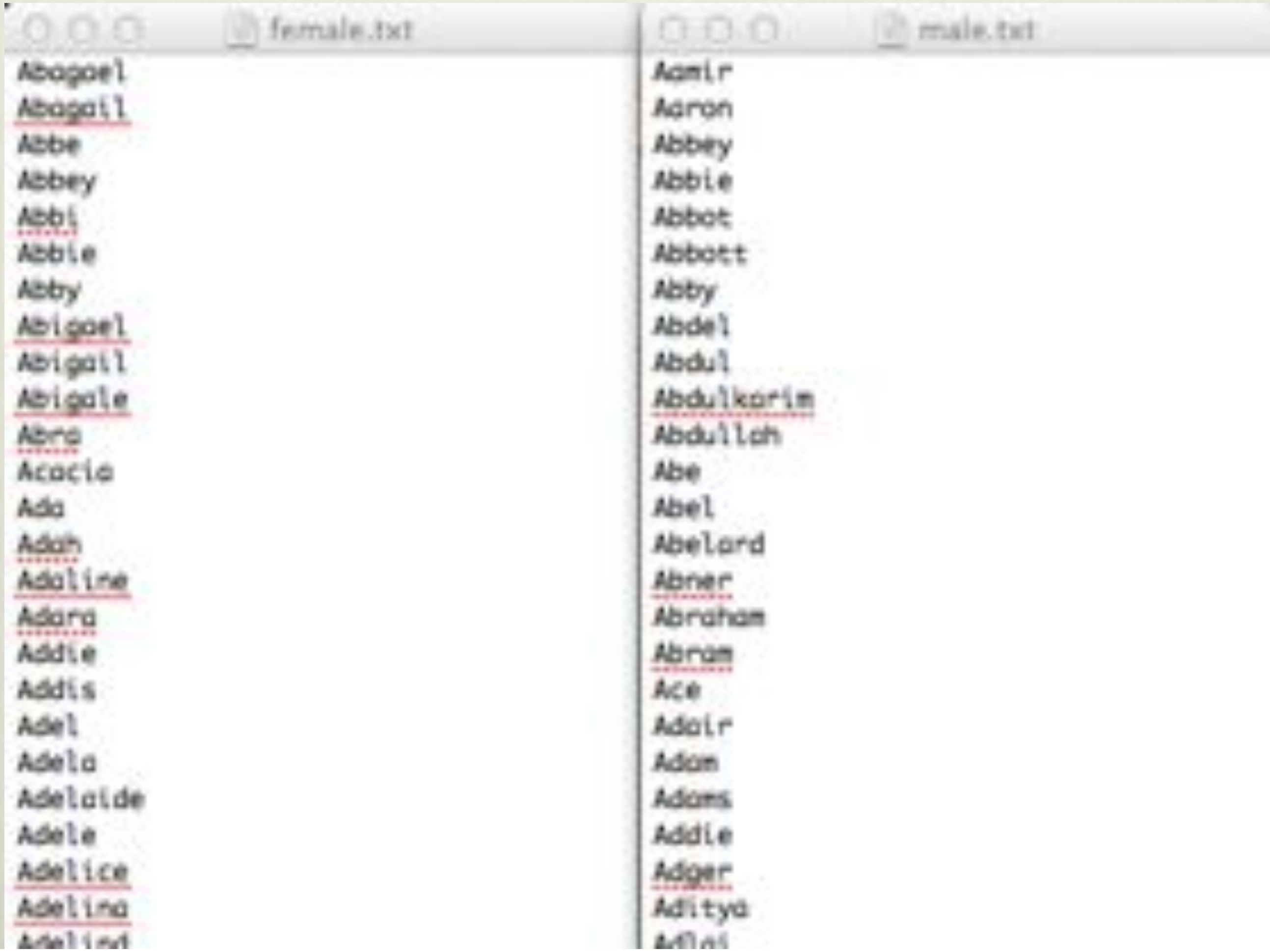
$$f_{male}(x,y)=\begin{cases}1 & \text{if } y \textit{ is } "\textit{male}", \text{ last letter of } x \text{ is a vowel}\\ 0 & \text{otherwise}\end{cases}$$

$$f_{female}(x,y)=\begin{cases}1 & \text{if } y \textit{ is } "\textit{female}", \text{ last letter of } x \text{ is a vowel}\\ 0 & \text{otherwise}\end{cases}$$

f_{male}(Bella, male) = 1 f_{female}(Bella, female) = 1

Expectation value of feature function

❖ Observation from training data set



Expectation value of feature function

- Observation from training data set

Training data size = 50. f_{female} feature found 10 names and f_{male} 5 names

$$E_p f_{male} = \frac{\sum_{i=1}^{N} f_{male}(x,y)}{N} = \frac{5}{50} = 0.1$$

$$E_p f_{female} = \frac{\sum_{i=1}^{N} f_{female}(x,y)}{N} = \frac{10}{50} = 0.2$$

Extraction Features

- Surface Features
  - Expected Answer Type Matching Features
  - Surface Pattern Matching
- Dependency Relation Features
- Semantic Structure Matching Features

Until now …

- We have defined N feature functions: $f_1, f_2, \ldots f_n$
- From the training data set we got the constraints:

$$\{E_p f_j = d_j, j = 1, \ldots, n\}$$

→ **How to combine all features to make an unified decision?**

Answer Extraction Modeling

- Naïve-Bayes

$$P(x \mid features) = \frac{P(x) * P(f_1 \mid x) * P(f_2 \mid x) * \ldots P(f_n \mid x)}{P(features)}$$

Answer Extraction Modeling

- Naïve-Bayes

$$P(x \mid features) = \frac{P(x) * P(f_1 \mid x) * P(f_2 \mid x) * \ldots P(f_n \mid x)}{P(features)}$$

- Problem
  - Make "naive" assumption that all features f_i are **independent**

Answer Extraction Modeling

- Maximum Entropy Model

Answer Extraction Modeling

- Maximum Entropy Model

  Maximize entropy = No assumption about feature dependency

Answer Extraction Modeling

- Maximum Entropy Model

  Maximize entropy = No assumption about feature dependency

$$p^*\left(y \mid x\right) = \frac{\exp\left(\sum_i \lambda_i f_i\left(x, y\right)\right)}{\sum_y \exp\left(\sum_i \lambda_i f_i\left(x, y\right)\right)}$$

y: what we predict

x: context

λ_i: weight of a feature function f_i

Parameter Estimation (**GIS**)

Initialize $\lambda_j = 0$

Do until convergerve

For each ***i***

calculate $E_{\lambda_j} f_j$

update $\lambda_j^{(n+1)} = \lambda_j^{(n)} + \frac{1}{C}(\log \frac{E\ f_j}{E_{\lambda_j} f_j})$

Parameter Estimation (**GIS**)

Initialize $\lambda_j = 0$

Do until convergerve

For each ***i***

calculate $E_{\lambda_j} f_j = \sum_{x \in \varepsilon} p^{(\lambda_j)}(x) f_j(x)$

where $p^{(\lambda_j)}(x) = \dfrac{e^{\sum_{j=1}^{k+1} \lambda_j^{(n)} f_j(x)}}{Z}$

update $\lambda_j^{(n+1)} = \lambda_j^{(n)} + \dfrac{1}{C}(\log \dfrac{E\, f_j}{E_{\lambda_j} f_j})$

Overview

1. Define a set of feature functions $\boldsymbol{f_i}$

2. Observe training data to find expectation value $\boldsymbol{d_i}$ of $\boldsymbol{f_i}$

3. Estimation parameters λ_i of each $\boldsymbol{f_i}$ based on $\boldsymbol{d_i}$

4. Compute probability of each output $\boldsymbol{y}$

$$p^*(y \mid x) = \frac{\exp\left(\sum_i \lambda_i f_i(x, y)\right)}{\sum_y \exp\left(\sum_i \lambda_i f_i(x, y)\right)}$$

A simple demo

Question: Which name is female?

A: Thomas	B: Kevin
C: Franz	D: Bella

A simple demo

Question: Which name is female?

A: Thomas　　B: Kevin
C: Franz　　D: Bella

→ ***p(female | x)***

A simple demo

1.Define a set of feature functions f_i

```
def names_features(name):
  features = {}
  features['startswith(vowel)'] = name[0].lower() in 'aeiouy'
  features['endswith(vowel)'] = name[-1].lower() in 'aeiouy'

  for letter in 'abcdefghijklmnopqrstuvwxyz':
     features['count(%s)' % letter] = name.lower().count(letter)
     features['has(%s)' % letter] = letter in name.lower()
     features['startswith(%s)' % letter] = (letter==name[0].lower())
     features['endswith(%s)' % letter] = (letter==name[-1].lower())
  return features
```

A simple demo

2.Observe training data to find expectation value $\boldsymbol{d_i}$ of $\boldsymbol{f_i}$

A simple demo

```
Unseen Names       P(Male|x)  P(Female|x)
----------------------------------------
  Thomas            0.871909  0.128091
  Kevin             0.551180  0.448820
  Franz             0.527687  0.472313
  Bella             0.165552  0.834448
----------------------------------------


  -1.227 endswith(d)==True and label is 'female'
   0.986 endswith(o)==True and label is 'male'
   0.891 endswith(m)==True and label is 'male'
  -0.885 endswith(s)==True and label is 'female'
  -0.842 count(s)==2 and label is 'male'
  -0.810 has(w)==True and label is 'female'
  -0.785 startswith(z)==True and label is 'female'
  -0.785 endswith(h)==True and label is 'male'
   0.676 endswith(g)==True and label is 'male'
  -0.663 startswith(w)==True and label is 'female'
```

A simple demo

3.Estimation parameters of each f_i based on $\boldsymbol{d_i}$

```
Unseen Names        P(Male|x)  P(Female|x)
------------------------------------------
  Thomas             0.871909  0.128091
  Kevin              0.551180  0.448820
  Franz              0.527687  0.472313
  Bella              0.165552  0.834448
------------------------------------------

 -1.227 endswith(d)==True and label is 'female'
  0.986 endswith(o)==True and label is 'male'
  0.891 endswith(m)==True and label is 'male'
 -0.885 endswith(s)==True and label is 'female'
 -0.842 count(s)==2 and label is 'male'
 -0.810 has(w)==True and label is 'female'
 -0.785 startswith(z)==True and label is 'female'
 -0.785 endswith(h)==True and label is 'male'
  0.676 endswith(g)==True and label is 'male'
 -0.663 startswith(w)==True and label is 'female'
```

λ_i

A simple demo

4. Compute probability of each output $\boldsymbol{y}$

```
Unseen Names         P(Male|x)  P(Female|x)
---------------------------------------------
  Thomas              0.871909   0.128091
  Kevin               0.551180   0.448820
  Franz               0.527687   0.472313
  Bella               0.165552   0.834448
---------------------------------------------

  -1.227 endswith(d)==True and label is 'female'
   0.986 endswith(o)==True and label is 'male'
   0.891 endswith(m)==True and label is 'male'
  -0.885 endswith(s)==True and label is 'female'
  -0.842 count(s)==2 and label is 'male'
  -0.810 has(w)==True and label is 'female'
  -0.785 startswith(z)==True and label is 'female'
  -0.785 endswith(h)==True and label is 'male'
   0.676 endswith(g)==True and label is 'male'
  -0.663 startswith(w)==True and label is 'female'
```

$p(female \mid x)$

λ_i

Overview

1. Define a set of feature functions $\boldsymbol{f_i}$

2. Observe training data to find expectation value $\boldsymbol{d_i}$ of $\boldsymbol{f_i}$

3. Estimation parameters λ_i of each $\boldsymbol{f_i}$ based on $\boldsymbol{d_i}$

4. Compute probability of each output $\boldsymbol{y}$

$$p^*(y \mid x) = \frac{\exp\left(\sum_i \lambda_i f_i(x, y)\right)}{\sum_y \exp\left(\sum_i \lambda_i f_i(x, y)\right)}$$

Answer extraction models
proposed by Dan Shen, 2008

- Answer Candidate Ranking

- Answer Candidate Classification

Answer Candidate Ranking

$$p^*\left(ac \mid q, \{ac_1, \ldots ac_N\}\right) = \frac{\exp\left(\sum_{m=1}^{M} \lambda_m f_m\left(q, ac\right)\right)}{\sum_{ac' \in \{ac_1, \ldots ac_N\}} \exp\left(\sum_{m=1}^{M} \lambda_m f_m\left(q, ac'\right)\right)}$$

q : question
ac : answer candidate
$\boldsymbol{f_i}$: feature function

$$ac^* = \underset{ac \in \{ac_1, ac_2, \ldots ac_n\}}{\arg\max}\; p(ac \mid \{ac_1, ac_2, \ldots ac_n\})$$

Answer Candidate Classification

$$p^*\left(c \mid q, ac\right) = \frac{\exp\left(\sum_{m=1}^{M} \lambda_{m,c} f_m\left(q, ac\right)\right)}{\sum_{c' \in \{true, false\}} \exp\left(\sum_{m=1}^{M} \lambda_{m,c'} f_m\left(q, ac\right)\right)}$$

q : question
ac : answer candidate
$\boldsymbol{f_i}$: feature function

$$ac^* = \underset{ac \in \{ac_1, ac_2, \ldots ac_n\}}{\arg\max}\; p(true \mid q, ac)$$

Comparison

	# Events	# Classes	# Parameters
Classification	$Q \times N$	2	$2M$
Ranking	Q	N	M

$$p^*\left(c \mid q, ac\right) = \frac{\exp\left(\sum_{m=1}^{M} \lambda_{m,c} f_m\left(q, ac\right)\right)}{\sum_{c' \in \{true, false\}} \exp\left(\sum_{m=1}^{M} \lambda_{m,c'} f_m\left(q, ac\right)\right)}$$

$$p^*\left(ac \mid q, \{ac_1, \ldots ac_N\}\right) = \frac{\exp\left(\sum_{m=1}^{M} \lambda_{m,c} f_m\left(q, ac\right)\right)}{\sum_{ac' \in \{ac_1, \ldots ac_N\}} \exp\left(\sum_{m=1}^{M} \lambda_m f_m\left(q, ac'\right)\right)}$$

Question

?

Reference

Dan Shen, Exploring Rich Evidence for Maximum Entropy-based Question Answering, PhD Thesis, 2008

A brief Maxent tutorial, Adam Berger, 2004

Maximum entropy models for natural language ambiguity resolution, Adwait ratnarpakhi, 1998

Maxent Models Conditional Estimation and Optimization, Dan Klein and Chris Manning, 2003

A Survey of Answer Extraction Techniques in Factoid Question Answering, Mengqiu Wang, 2006

Maximum entropy model, Fei Xia, 2006

A Simple Introduction to Maximum Entropy Models for Natural Language Processing, Adwait Ratnaparkhi, 1997