

Conditional Random Field Tutorial

Francis Yang, Shiang Chi Hsu, Yi Ting Ding, Han Chen

Data Introduction

- AI CUP 2018 生醫論文自動分析熱身賽
- 競賽所提供的資料集來源如下：
 - PubMed之生醫文獻摘要
 - PubMed Central生醫文獻全文中每個章節的第一個段落 (僅使用全文中之 Introduction、Discussion、Result與Conclusion)
 - PubMed Central之figure captions, 一篇生醫文獻只選一個 figure caption做為標註。
 - 生醫專利文件之摘要

Visualize the data

Task Dataset	Articles	Original Source			
		生醫文獻摘要	生醫文獻 First Paragraph	生醫專利摘要	全文段落
Train	2000	2000	-	-	-
Development	1000	500	150	150	200
Test	1000	500	150	150	200

Data Introduction

- Task:
 - **Named Entity Recognition (NER)**
 - Extract the correct named entity from the articles.

- Target for our course:
 - NE Type
 - Text - **Entity**(命名實體)

Article ID	NE Type	Position	Length	Text
25693640	Chemical	32	10	asparagine
25693640	Chemical	51	8	arginine
25693640	Chemical	43	7	glycine
25693640	Chemical	832	10	asparagine
25693640	Chemical	851	8	arginine
25693640	Chemical	843	7	glycine
25693640	Chemical	460	15	lysine residues

What is Named Entity Recognition (NER)?

To recognition the entity that the corpus need.


But, what is “entity”?

For instance, in AI Cup 2018, we need to extract the disease, gene and chemical in the articles. So, the “disease”, “gene” or the “chemical” are regarded as the entity.

PMID: 26385350

AID-associated DNA repair pathways regulate malignant transformation in a murine model of BCL6-driven diffuse large B-cell lymphoma.

Somatic hypermutation and class-switch recombination of the immunoglobulin (Ig) genes occur in germinal center (GC) B cells and are initiated through deamination of cytidine to uracil by activation-induced cytidine deaminase (AID). Resulting uracil-guanine mismatches are processed by uracil DNA glycosylase (UNG)-mediated base-excision repair and MSH2-mediated mismatch repair (MMR) to yield mutations and DNA

 Disease Named Entity

 Gene Named Entity

 Chemical Named Entity

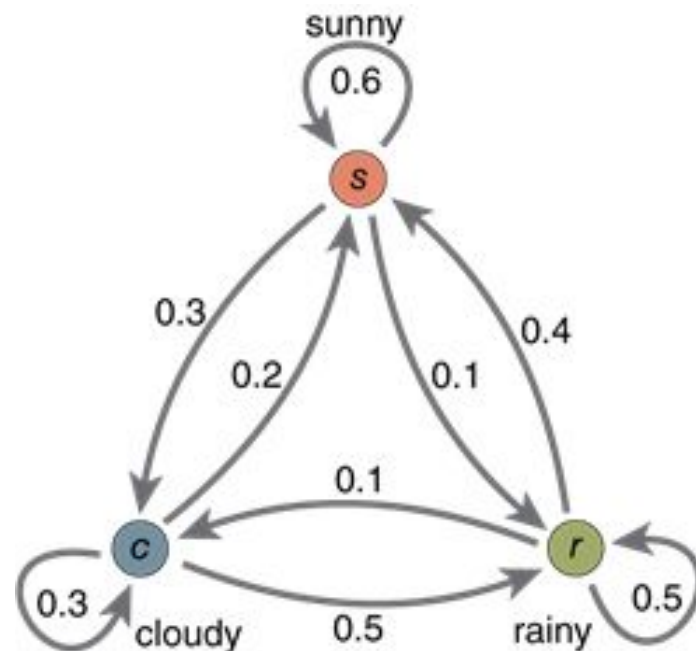
How to do NER?

- Why choose to use CRF?
 - CRF can find the best label path for the whole sentence.
 - Hidden Markov Model (HMM)
 - Freely to design the features for the NER model.

→ And what is Hidden Markov Model?

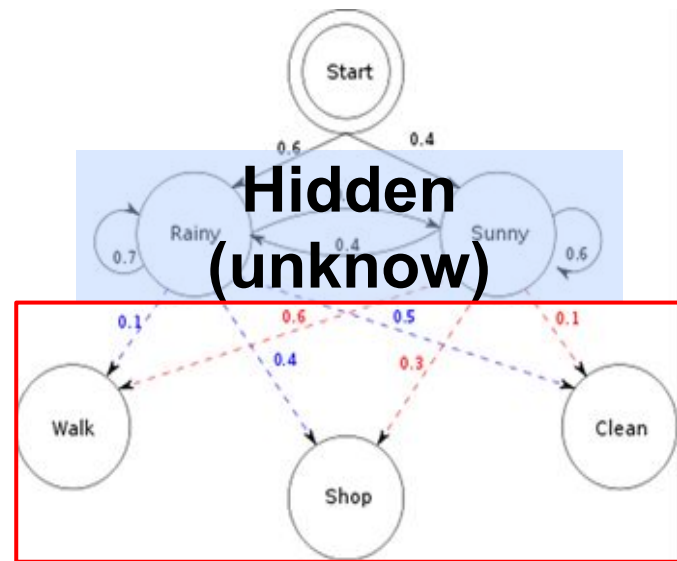
Hidden Markov Model (HMM)

- We use a weather model to illustrate.
 - Three circles represent three different statuses, sunny, cloudy and rainy.
 - The numbers beside the lines mean the transition probability.
 - The total probability of each status transform to itself or others statuses is 1.

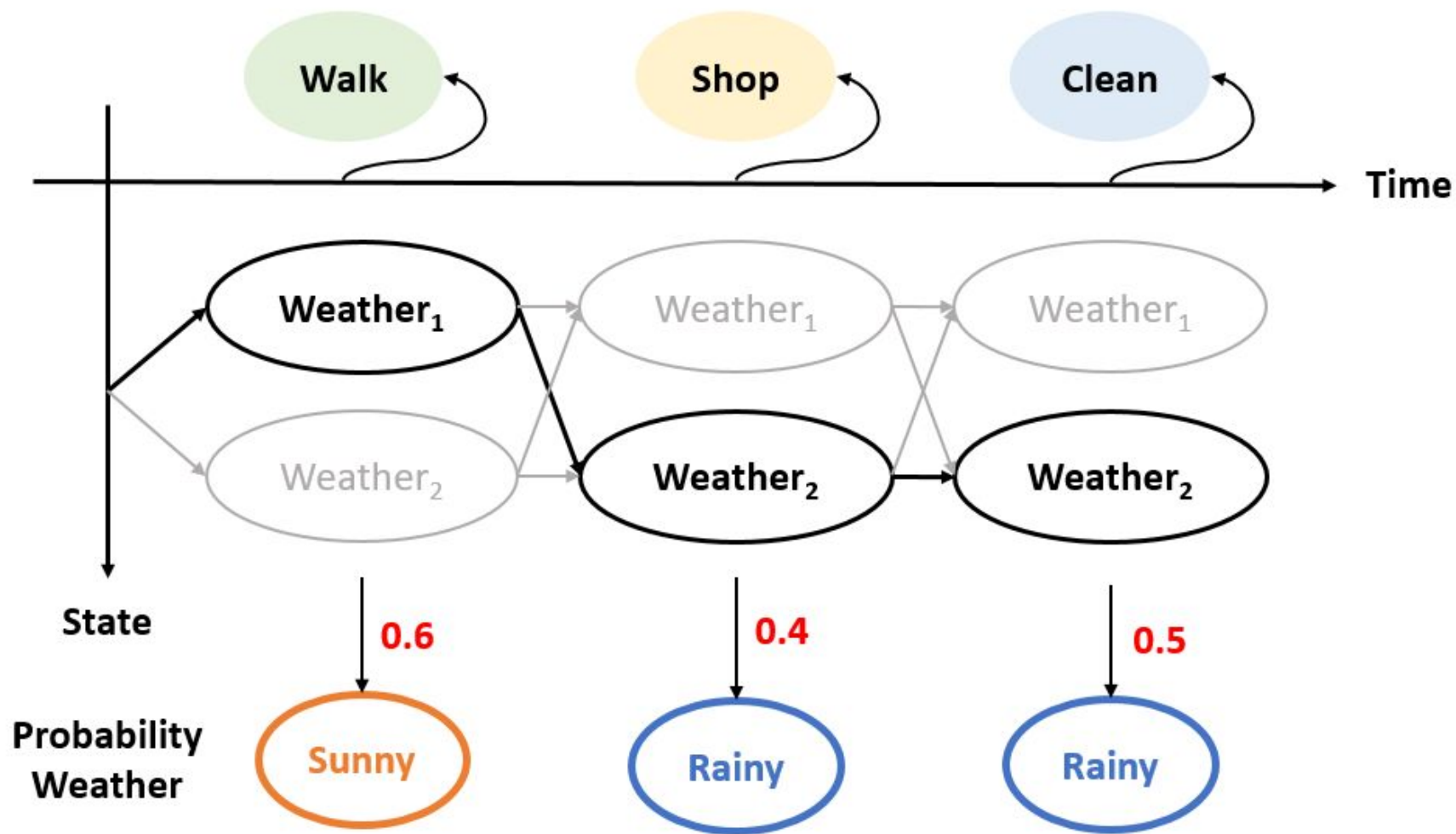


Hidden Markov Model (HMM)

- Hidden means we cannot observe these statuses.
- We can observe the status by some other ways.
 - Observation probability (blue and red number)
 - We can infer the states(weather) based on the observable status(Walk, Shop or Clean).
- Observation probability can help us to find **the most likely** status.
 - CRF can find the **best label** path.



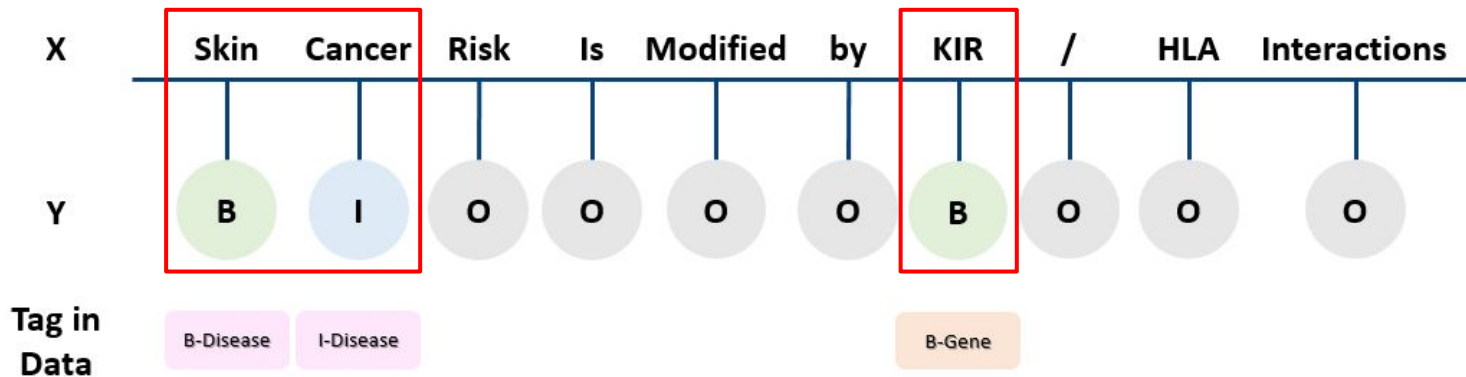
Observation



Why CRF can solve NER task?

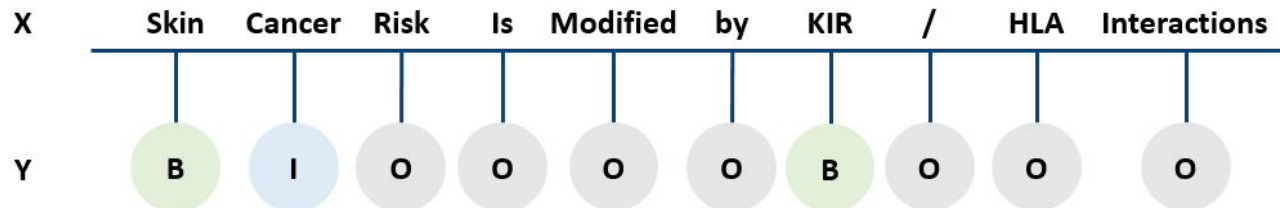
Get the label(B: begin, I: inside, O: other) of each token.

According to some rules like 'BI,' 'B,' and then to combine these word.



For this example, we can get two entities, "Skin Cancer" and "KIR".

CRF



- Obtaining the scores of each possible Y (label sequence):

$$\text{score}('000000', X), \text{score}('S00000', X), \dots, \text{score}('EEEEEE', X)$$

- Get probability of each possible Y for X (token sequence):

$$p(Y|X) = \frac{\exp(\text{score}(Y, X))}{\sum_{Y'} \exp(\text{score}(Y', X))}$$

- Take the highest probability as the best Y of the X.

CRF

- Score function: $score(\mathbf{Y}, \mathbf{X}) = \sum_{j=1}^m \sum_{i=1}^n \lambda_j f_j(\mathbf{X}, i, y_i, y_{i-1})$

λ : weight of feature function

f : feature function

j : number of feature function

i : position of the token in the token sequence

y_i : label of current token

y_{i-1} : label of last token

- Feature function: $f_1(\mathbf{X}, i, y_i, y_{i-1}) = \begin{cases} 1 & \text{if } y_i \text{ is } S \text{ and the } i^{th} \text{ word is a } NOUN; \\ 0 & \text{otherwise} \end{cases}$

- For instance, if the current label is 'S' and the current word is a noun, the label of the current token has a high probability being 'S'.
- You can design any feature function you want to find out what feature can help the recognition.

Design for feature function (template)

- Template has two categories: Unigram & Bigram
- Unigram/Bigram : output the unigram/bigram of the token, not feature.
 - Unigram: automatically generate $L*N$ feature functions.
 - (L: number of labels; N: unique string in the expand feature from the template)

- U01:%x[0,1]

```
func1 = if (output = B and feature = "NNP") return 1 else return 0  
func2 = if (output = I and feature = "NNP") return 1 else return 0  
func3 = if (output = O and feature = "NNP") return 1 else return 0  
...
```

- Bigram: combine with the last token and generate $L*L*N$ feature functions.

- B01:%x[0,1]

```
func1 = if (output = B and output = B and feature = "NNP") return 1 else return 0  
func2 = if (output = I and output = B and feature = "NNP") return 1 else return 0  
func3 = if (output = O and output = B and feature = "NNP") return 1 else return 0  
...
```

How to design the template?

- Template is used for selecting the features you designed as the input.
 - Assumed current token is “Risk” :

Feature Template	Explanation	Expanded Feature
%x[0,0]	R:0, C:0	Risk
%x[0,1]	R:0, C:1	4
%x[-1,0]	R:-1, C:0	Cancer
%x[-1,1]	R:1, C:1	2
%x[0,0]/%x[0,1]	R:0, C:0/R:0, C:1	Cancer/4

Token	Feature (length)	Label
Skin	4	B
Cancer	6	I
Risk	4	O
Is	2	O
Modified	8	O

- ❖ R: Row
- ❖ C: Column

How to design the template?

- Template is used for selecting the features you designed as the input.
 - Assumed current token is “Risk” :

Feature Template	Explanation	Expanded Feature
%x[0,0]	R:0, C:0	Risk
%x[0,1]	R:0, C:1	4
%x[-1,0]	R:-1, C:0	Cancer
%x[-1,1]	R:1, C:1	2
%x[0,0]/%x[0,1]	R:0, C:0/R:0, C:1	Cancer/4

❖ R: Row
❖ C: Column

Token	Feature (length)	Label
Skin	4	B
Cancer	6	I
Risk	4	O
Is	2	O
Modified	8	O

➡➡ Current token : Row & Column = 0

How to design the template?

- Template is used for selecting the features you designed as the input.
 - Assumed current token is “Risk” :

Feature Template	Explanation	Expanded Feature
%x[0,0]	R:0, C:0	Risk
%x[0,1]	R:0, C:1	4
%x[-1,0]	R:-1, C:0	Cancer
%x[-1,1]	R:1, C:1	2
%x[0,0]/%x[0,1]	R:0, C:0/R:0, C:1	Cancer/4

Token	Feature (length)	Label
Skin	4	B
Cancer	6	I
Risk	4	O
Is	2	O
Modified	8	O

❖ R: Row
❖ C: Column

➡➡ Right : Column plus one

How to design the template?

- Template is used for selecting the features you designed as the input.
 - Assumed current token is “Risk” :

Feature Template	Explanation	Expanded Feature
%x[0,0]	R:0, C:0	Risk
%x[0,1]	R:0, C:1	4
%x[-1,0]	R:-1, C:0	Cancer
%x[-1,1]	R:1, C:1	2
%x[0,0]/%x[0,1]	R:0, C:0/R:0, C:1	Cancer/4

Token	Feature (length)	Label
Skin	4	B
Cancer	6	I
Risk	4	O
Is	2	O
Modified	8	O

❖ R: Row
❖ C: Column

➤ Up : Row minus one

How to design the template?

- Template is used for selecting the features you designed as the input.
 - Assumed current token is “Risk” :

Feature Template	Explanation	Expanded Feature
%x[0,0]	R:0, C:0	Risk
%x[0,1]	R:0, C:1	4
%x[-1,0]	R:-1, C:0	Cancer
%x[-1,1]	R:1, C:1	2
%x[0,0]/%x[0,1]	R:0, C:0/R:0, C:1	Cancer/4

Token	Feature (length)	Label
Skin	4	B
Cancer	6	I
Risk	4	O
Is	2	O
Modified	8	O

❖ R: Row
❖ C: Column

➤ Down : Row pulls one
➤ Right : Column pulls one

What features can be designed?

- Feature that we used
 - Length of word
- Other features
 - POS (Part-Of-Speech)
 - Obtained from Stanford POS Tagger
 - noun, verb, adjective, etc.

Feature that we use

Sequence ↓	Length of Word		State
	Skin	4	B
	Cancer	6	I
	Risk	4	O
	Is	2	O
	Modified	8	O
	by	2	O
	KIR	3	B
	/	1	I

How to use CRF++ ?

- Prepare :
 - Download “CRF++-0.58.tar.gz” : <https://taku910.github.io/crfpp/>
 - Template
 - Training_data
 - Testing_data
- Train command line
 - parameters :
 - -a CRF-L2/CRF-L1 (default CRF-L2)
 - -c float
 - Adjust the value if it is overfitting.(default 1.0)
 - -f int
 - Filter the words if they are under the value.(default 1)
 - -p int
 - Number of CPU.(default 1)

How to use CRF++ ?

- Train command line
 - command line format :
 - `crf_learn.exe Template Training_data [output model]`
 - e.g. `> crf_learn -c 1.5 -f 3 -p 4 ../D_template ../D_training.data ../D_training.model`
- Test command line
 - command line format :
 - `crf_test.exe [output model] Testing_data >> [out_predict result]`
 - e.g. `> crf_test -m ../D_training.model ../D_testing.data >> pred_res.txt`

CRF++-0.58 Installation on linux

- Requirements

- C++ compiler (gcc 3.0 or higher)

- How to make

```
% tar zxvf CRF++-0.58.tar.gz
% cd CRF++-0.58
% ./configure
% make
% su
% make install
```

- If you get the error:

```
error while loading shared libraries: libcrfpp.so.0: cannot open shared object file:
No such file or directory
```

- run `sudo ldconfig` and then try again in a new terminal window.