# Maximum Entropy Model (MEM)

# Disadvantages of HMM

1. At run-time, u get unknown words, we do not have the required probabilities.

   Possible solutions:

   - use morphological ceus (Captilization, suffix) to assign a more calculated guess

2. Limited context:

   is clearly marked -> verb, past participle

   he clearly marked -> verb, past tense

   Possible solution:

   Use higher order model

# Maximum Entropy Model

- Allows to identify heterogeneous set of features which contribute to the choice of POS tag of the current word.
- For eg:
  - Whether it is the first word in the article
  - Whether the next word is “to”
  - Whether one of the last 5 words is a preposition, etc
- Max Ent combines these features in a probabilistic framework to be able to come up with actual POS sequence for a sentence

# Maximum Entropy : The Model

P(y/x) = $$\frac{1}{Z(x)} \exp\left(\sum_i \lambda_i f_i(x,y)\right)$$

where , Z(x) is a normalizing constant given by :

Z(x) = $$\sum_y \exp\left(\sum_i \lambda_i f_i(x,y)\right)$$

ensuring P(y/x) will add up to 1 for all values of y.

- $\boldsymbol{\lambda_i}$ is the weight given to a feature $\mathbf{f_i}$
- **x** denotes an observed data and **y** denotes a class

**What is the form of features?**

# Features in Maximum Entropy Model

- Features encode elements of the context x for predicting tag y
- Context x is taken around the word w, for which a tag y is to be predicted

**What are features?**

- Features are binary valued functions, e.g.,

$$f(x,y) = \begin{cases} 1 & \text{, if isCapitalized(w)\&y=NNP} \\ 0 & \text{,otherwise} \end{cases}$$

# Example Features

***Example: Named Entities***

- LOCATION(in Pune)
- LOCATION(in Québéc)
- DRUG(taking Paracetamol)
- PERSON(saw Sumedh)

***Example features:***

- $f_1(x, y)$= [y=LOCATION Λ w_$_1$="in" Λ isCapitalized(w)]
- $f_2(x, y)$= [y=LOCATION Λ w_$_1$="in"Λ hasAccentedLatinChar(w)]
- $f_3(x, y)$= [y=DRUG Λ ends(w, "mol")]
- $f_4(x, y)$= [y=PERSON Λ w_$_1$="saw" Λ isCapitalized(w)]

# Tagging with Maximum Entropy Model

- W = $w_1 \ldots w_n$- words in the corpus (observed)
- T = $t_1 \ldots t_n$ – the corresponding tags(unknown)

  Tag sequence candidate $\{t_1,\ldots,t_n\}$ has conditional probability:

$$P(t_1,\ldots,t_n / w_1 \ldots w_n) = \prod_{i=1}^{n} p(t_i / x_i)$$

- $x_i$ is context around word $w_i$.
- The context $x_i$ also includes previously assigned tags for a fixed history.
- Beam search is used to find the most probable sequence

# Beam search algo

- At each position, keep the top k complete sequences
- Extend each sequence in each local way
- The extension compete for the k slots at the next position

# What is an Max Entropy Model

**Intuitive Principle:**

Model all that is known and assume nothing about that which is unknown.

{ Given observation, and certain hypothesis, Take the model that satisfies all the constrains and do not make any additional assumptions}

**That says,**

Given a collection of facts, choose a model which is consistent with all the facts, but otherwise it is not making further assumption i.e., as uniform as possible.

- For all the possible models that satisfies a given set of constrains , choose the one that is the most uniform of all.

- What do we mean by this?

# Maximum Entropy: Overview

- Suppose we wish to model an expert translator's decisions concerning the proper French rendering of the English word '**in**'
- Given : Data from actual translator who translated some sentences from English to French.
- Now, how to model this-translation of English to French?

# Maximum Entropy: overview

- Each french word or phrase f is assigned an estimate p(f), probability that the expert would choose f as a translation of ‘in’.
- Collect a large sample of instances of expert decisions
- From these samples, observe some facts.
- Goal: extract a set of facts from sample examples about the decision-making process(first task) and used these facts inside ur model (Second task).
  - Facts would be features.

# Maximum Entropy Model: overview

- **First Clue**: list of allowed translations
  - Suppose the translator always chooses among {dans, en, a, au, de}
- Find probability distribution over french phrases P(f)
- Suppose there are I million french phrases: f1,..,fm
- So what is the constrain: the translator always chooses among {dans, en, a, au, de}
- **First Constrain becomes:**

$$P(dans) + P(en) + P(a) + P(au) + P(de) = 1$$

Let us call it as:

$$P(f1) + P(f2) + P(f3) + P(f4) + P(f5) = 1$$

- **Given constrain, choose model ?**
- There are infinite number of models **p** that satisfies the constrain.
- In such situation, P(f6, .., fm)=0. If any of these features is non-zero then constrain not satisfied.
- As there are infinite models, like:

P(f1) - 0 to 1,

P(f2) - 0 to 1, and so on

# What MaxEntropy Does?

- Among all the models that satisfies constrain, it takes that model that is **most uniform**.
- **Among these models, which is most uniform.**
- The most uniform will be the one, in which P(fi) = 1/5 (where i =1 to 5). Allocate the total prob. evenly among the five possible phrases.
- **Is it the most uniform model overall?**
  - Suppose , the constrain is not there.
- Most uniform model is one which gives equal probability to each of the million french phrases.

# Maximum Entropy Model: overview

- Some more facts (clues) from expert's decision
- **Second clue:** Suppose the expert choose either 'dans' or 'en' 30% of the time.

  P(f1)+P(f2) = 0.3

**What will be the most uniform model?**

P(f1)+P(f2)=0.3 means P(f1)=0.15 , P(f2)=0.15

P(f3),P(f4),P(f5) = 0.7/3

- **Third clue:** the expert choose either 'dans' or 'au' 50% of the time.

  P(f1)+P(f3) = 0.5

# Maximum Entropy Model: overview

- The  situation gets complex with constrains.
- We need  to understand the concept of maximum entropy
- To find out the model that is most uniform given constrains is equivalent to finding a model that has maximum entropy among all.

**How do we measure uniformity of  a model?**

As we add complexity to the model, we face two difficulties:

- What exactly is meant by “uniform”?
- How can one measure the uniformity of a model?

# Maximum Entropy Modeling

**Entropy** : measure the uncertainty of a distribution.

**Quantifying uncertainty ("surprise")**

- Event x
- Probability $P_x$
- Surprise: $\log(1/P_x)$

**Entropy measures the amount of surprise in this event.**

- If it is very very surprising then entropy is more
- If it is very obvious then entropy is low
- Prob low then entropy is high
- Prob high then entropy is low
- Therefore, Entropy for event x = $\log(1/P_x)$

# Maximum Entropy Modeling

- **For distribution , how do we compute entropy?**
- Take expected value of surprise i.e., $-\sum_x p_x log_2 p_x$
- Standard formula for Entropy: expected surprise (over p)

$H(p) = E_p[\log_2(1/p_x)] = -\sum_x p_x log_2 p_x$

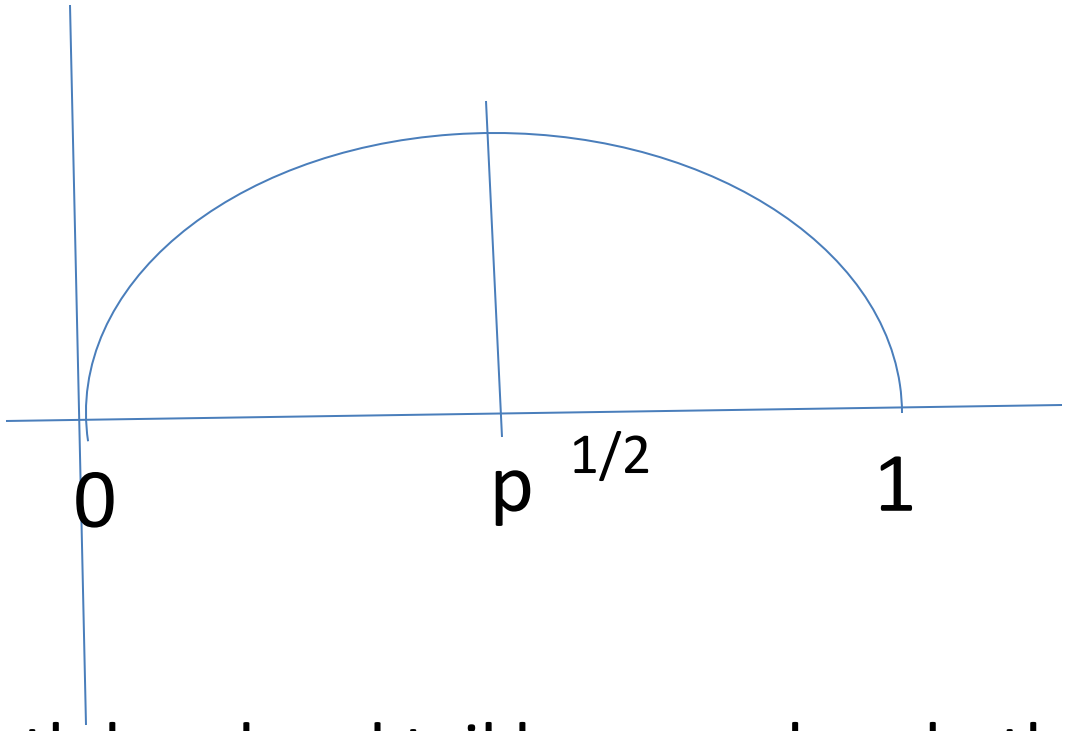
Now, in case of distribution when will the entropy maximum .

**Consider the example of coin tossing.**

Prob of tossing head :P , Entropy : $-p\log_2 p$

Prob of tossing tail : 1- P, Entropy: $-(1-p)\log_2(1-p)$

# Maximum Entropy Modeling




- When both head and tail has equal prob, then entropy of model is highest.
- When many vars, the distribution will have max entropy when it is  most uniform.

# Maximum Entropy Modeling

- Among all the possible distributions, want to choose a model which maximizes entropy, subject to feature-based constrains:

- Empirical probability we are finding from data is similar to the one the model is giving.
- Adding constraints:
  - Lowers maximum entropy
  - Brings the distribution further from uniform and closer to data.

# Note:

- Maximize entropy subject to certain constraints.
- So, we will have certain constraints from the data and these constraints are nothing, but the features.
- Try to obtain maximum entropy
- As we keep on adding constraints one by one, the entropy of model decreases.
- It comes close to data.
- Initially might start with the most uniform model that is giving the equal probability to everything, but as you keep on adding constraint one by one you come close to data and far away from the most uniform model.
- So, you decrease your entropy and you come closer to data. So, this is the idea.
- So, given some constraints , the model comes as close as possible to the data, but otherwise obtain the most uniform model in terms of maximum entropy.

# Maximum Entropy Markov Model

- **How to use MEM?**

  Consider the max entropy model for POS tagging, where u want to estimate P(tag/word). In a hypothetical setting, assume that tag can take the values D, N and V (Determiner, Noun and Verb resp). The var word could be any member of a set W of possible words, where **W** contains words a, boy, plays, as well as additional words. The distribution should give the following probabilities:

**1.** P(D/a) =0.9 **2.** P(N/boy)=0.9 **3.** P(V/plays)=0.9

**4.**P(D/word) = 0.6, for any word other than a, boy, or plays

**5.**P(N/word) = 0.3, for any word other than a, boy, or plays

**6.**P(V/word) = 0.1, for any word other than a, boy, or plays

# Maximum Entropy Markov Model

- It is assumed that all other probabilities, not defined above could take any value such that:

$\sum_{tag} P(tag/word) = 1$ is satisfied for any word in W

**Now, Questions to be addressed:**

1. Define the features of your maximum entropy model that can model this distribution. Mark your features as $f_1$, $f_2$, and so on. Each feature should have the same format (i.e., f(x,y) where x is context and y is tag) [Hint: 6 features should make the analysis easier]

# Maximum Entropy Markov Model

2. For each feature $f_i$, assume a weight $\lambda_i$. Now, write down expression for the following probabilities in terms of your model parameters (i.e., $\lambda_1$ to $\lambda_6$)
   - P(D/cat)
   - P(N/laughs)
   - P(D/plays)
3. What values do the parameters(i.e., $\lambda_1$ to $\lambda_6$) in your model take to give the distribution as described above (i.e., P(D/a)=0.9 and so on). You may leave the final answer in terms of equations.

# Solution:

**Answer Q1)**

- We have some facts or constrains from data. Consider first constrain, P(D/a) =0.9 and so on
- Find out some features, f(x, y), such that tries to match above constrain i.e., P(D/a)

**Now, what is observed, when word is "a" and tag is "D" with prob. =0.9**

**What features will model this?**

f(x, y) = word ="a" and tag = "D" . Let us call it $f_1$.

**$f_1$=f(x, y) = 1 , if word ="a" and tag = "D"**

**= 0 , otherwise**

# Solution:

**Answer Q1)**

Now, define other features . So take 2nd constrain:
P(N/boy) =0.9

f(x, y) = word =“boy” and tag = “N”. Let us call it $f_2$.

**$f_2$=f(x, y) = 1 , if word =“boy” and tag = “N”**

**= 0 , otherwise**

Similarly, for 3rd constrain: P(V/plays)

**$f_3$=f(x, y) = 1 , if word =“plays” and tag = “V”**

**= 0 , otherwise**

# Solution:

**Answer Q1)**

Now, for 4th constrain: P(D/word) = 0.6

word ϵ W – {a, boy, plays} = W′

**$f_4$= f(x, y) = 1 , if word ϵ W′ and tag = “D”**

**= 0 , otherwise**

**$f_5$= f(x, y) = 1 , if word ϵ W′ and tag = “N”**

**= 0 , otherwise**

**$f_6$= f(x, y) = 1 , if word ϵ W′ and tag = “V”**

**= 0 , otherwise**

Q2) . For each feature $f_i$, assume a weight $\lambda_i$. Now, write down expression for the following probabilities in terms of your model parameters(i.e., $\lambda_1$ to $\lambda_6$)

P(D/cat) P(N/laughs) P(D/boy)

**Solution:** Finding model parameters, $\lambda_1$ to $\lambda_6$ .

In terms of MEM,

**P(D/Cat) =** $\dfrac{e^{\sum \lambda_i f_i}}{Z}$

Now, $\sum \lambda_i f_i(x,y) = \lambda_1 f_1 + \lambda_2 f_2 + \lambda_3 f_3 + \lambda_4 f_4 + \lambda_5 f_5 + \lambda_6 f_6 = \lambda_4$

Where x=cat and y=D

For this, $f_1 = 0$, $f_2 = 0$, $f_3 = 0$, $f_4 = 1$, $f_5 = 0$, $f_6 = 0$

**P(D/Cat) =** $\dfrac{e^{\lambda 4}}{Z}$

To find Z, compute other 2 prob's i.e. P(N/Cat) , P(V/Cat)

Q2) . For each feature $f_i$, assume a weight $\lambda_i$. Now, write down expression for the following probabilities in terms of your model parameters(i.e., $\lambda_1$ to $\lambda_6$)

P(D/cat) P(N/laughs) P(D/boy)

**Solution:**

**P(N/Cat) =** $\dfrac{e^{\sum \lambda_i f_i}}{Z}$

Now, $\sum \lambda_i f_i(x,y) = \lambda_1 f_1 + \lambda_2 f_2 + \lambda_3 f_3 + \lambda_4 f_4 + \lambda_5 f_5 + \lambda_6 f_6 = \lambda_5$

Where x = cat and y = N

For this, $f_1 = 0$, $f_2 = 0$, $f_3 = 0$, $f_4 = 0$, $f_5 = 1$, $f_6 = 0$

**P(N/Cat) =** $\dfrac{e^{\lambda 5}}{Z}$

**Similarly, P(V/cat) =** $\dfrac{e^{\lambda 6}}{Z}$ **as** $f_6 = 1$

**Therefore,** $Z = e^{\lambda 4} + e^{\lambda 5} + e^{\lambda 6}$

**Substituting Z value for each.**

Q2) . For each feature $f_i$, assume a weight $\lambda_i$. Now, write down expression for the following probabilities in terms of your model parameters(i.e., $\lambda_1$ to $\lambda_6$)

P(D/cat) P(N/laughs) P(D/boy)

**Solution:**

$$\textbf{P(D/Cat)} = \frac{e^{\lambda 4}}{Z} = \frac{e^{\lambda 4}}{e^{\lambda 4} + e^{\lambda 5} + e^{\lambda 6}}$$

**Now, find P(N/laughs) and P(D/man)**

Q2) . For each feature $f_i$, assume a weight $\lambda_i$. Now, write down expression for the following probabilities in terms of your model parameters(i.e., $\lambda_1$ to $\lambda_6$)

P(D/cat) P(N/laughs) P(D/boy)

**Solution:**

**P(N/laughs) =** $\dfrac{e^{\sum \lambda_i f_i}}{Z}$

Now, $\sum \lambda_i f_i(x,y) = \lambda_1 f_1 + \lambda_2 f_2 + \lambda_3 f_3 + \lambda_4 f_4 + \lambda_5 f_5 + \lambda_6 f_6 = \lambda_5$

Where x = laughs and y = N

For this, $f_1 = 0$, $f_2 = 0$, $f_3 = 0$, $f_4 = 0$, $f_5 = 1$, $f_6 = 0$

P(N/laughs) = $\dfrac{e^{\lambda 5}}{Z}$, P(D/laughs) = $\dfrac{e^{\lambda 4}}{Z}$ as $f_4 = 1$

P(V/laughs) = $\dfrac{e^{\lambda 6}}{Z}$ as $f_6 = 1$

Therefore, $Z = e^{\lambda 4} + e^{\lambda 5} + e^{\lambda 6}$

Substituting Z value for each.

**P(N/laughs) =** $\dfrac{\mathbf{e^{\lambda 5}}}{\mathbf{e^{\lambda 4} + e^{\lambda 5} + e^{\lambda 6}}}$

Q2) . For each feature $f_i$, assume a weight $\lambda_i$. Now, write down expression for the following probabilities in terms of your model parameters(i.e., $\lambda_1$ to $\lambda_6$)

P(D/cat) P(N/laughs) P(D/boy)

**Solution:**

**P(D/boy) =** $\dfrac{e^{\sum \lambda_i f_i}}{Z}$

Now, $\sum \lambda_i f_i(x,y) = \lambda_1 f_1 + \lambda_2 f_2 + \lambda_3 f_3 + \lambda_4 f_4 + \lambda_5 f_5 + \lambda_6 f_6 = 0$

Where x = boy and y = D

For this, $f_1 = 0$, $f_2 = 1$, $f_3 = 0$, $f_4 = 0$, $f_5 = 0$, $f_6 = 0$

P(D/boy) = $\dfrac{e^0}{Z}$, P(N/boy) = $\dfrac{e^{\lambda 2}}{Z}$ as $f_2 = 1$

P(V/boy) = $\dfrac{e^0}{Z}$ as all features have false value

Therefore, $Z = e^{\lambda 2} + e^0 + e^0 = e^{\lambda 2} + 1 + 1 = e^{\lambda 2} + 2$

Substituting Z value for P(D/boy).

**P(D/boy) =** $\dfrac{1}{e^{\lambda 2} + 2}$

Q2) . For each feature $f_i$, assume a weight $\lambda_i$. Now, write down expression for the following probabilities in terms of your model parameters(i.e., $\lambda_1$ to $\lambda_6$)

P(D/cat) P(N/laughs) P(D/boy)

**Solution:**

$$\mathbf{P(D/Cat) = \frac{e^{\lambda 4}}{Z} = \frac{e^{\lambda 4}}{e^{\lambda 4} + e^{\lambda 5} + e^{\lambda 6}}}$$

$$\mathbf{P(N/laughs) = \frac{e^{\lambda 5}}{Z} = \frac{e^{\lambda 5}}{e^{\lambda 4} + e^{\lambda 5} + e^{\lambda 6}}}$$

$$\mathbf{P(D/boy) = \frac{1}{Z} = \frac{1}{e^{\lambda 2} + 2}}$$

Q3) . What values do the parameters(i.e., $\lambda_1$ to $\lambda_6$) in your model take to give the distribution as described above (i.e., P(D/a)=0.9 and so on). You may leave the final answer in terms of equations.

**Solution:**

$$\mathbf{P(D/a) = \frac{e^{\lambda 1}}{Z}}$$

**What is Z?**

Find : P(V/a) and P(N/a)

$$\mathbf{P(V/a) =} \frac{1}{Z}[e^{[0]}] = \text{all features are zero} = \frac{1}{Z}$$

$$\mathbf{P(N/a) =} \frac{1}{Z}[e^{[0]}] = \text{all features are zero} = \frac{1}{Z}$$

$$\mathbf{Z = e^{\lambda 1} + 1 + 1 = e^{\lambda 1} + 2}$$

Q3) . What values do the parameters(i.e., $\lambda_1$ to $\lambda_6$) in your model take to give the distribution as described above (i.e., P(D/a)=0.9 and so on). You may leave the final answer in terms of equations.

**Solution:**

$$P(D/a) = \frac{e^{\lambda 1}}{e^{\lambda 1} + 2} = 0.9$$

$$e^{\lambda 1} = 0.9 * ( e^{\lambda 1} + 2)$$

$$e^{\lambda 1} - 0.9e^{\lambda 1} = 1.8$$

$$0.1e^{\lambda 1} = 1.8$$

$$e^{\lambda 1} = 18$$

$$\text{Log}_e(18) = \lambda_1$$

$$\lambda_1 = 2.8903$$

$$P(D/a ) = \frac{e^{2.8903}}{e^{2.8903} + 2} = 0.9$$

# Practice Question

- Suppose want to use a MaxEnt tagger to tag the sentence " The Little Book". We know the top 2 POS tags for the words : the little and book are:{Det Noun}, {Verb Adj} and { Verb, Noun} respectively. Assume the MaxEnt model uses the following history $h_i$(context) for a word wi:

$$\mathbf{h_i=\{w_i, w_{i-1}, w_{i+1}, t_{i-1}\}}$$

where $w_{i-1}$ and $w_{i+1}$ corresponds to previous and next words and $t_{i-1}$ corresponds to tag of the previous word. Accordingly the following features are used by Max Ent Model:

- $f_1$:$t_{i-1}$ =Det and $t_i$ =Adj
- $f_2$:$t_{i-1}$ =Noun and $t_i$ =Verb
- $f_3$:$t_{i-1}$ =Adj and $t_i$ =Noun

# continued…

- Features are used by Max Ent Model: continued

- $f_4$: $w_{i-1}$ =the and $t_i$ =Adj
- $f_5$: $w_{i-1}$ =the, $w_{i+1}$ = book and $t_i$ =Adj
- $f_6$: $w_{i-1}$ = light and $t_i$ =Noun
- $f_7$: $w_{i+1}$ = light and $t_i$ =Det
- $f_8$: $w_{i-1}$ = NULL and $t_i$ =Noun

Assumed that each feature has a uniform weight of 1.0. use beam search algorithm with a beam size of 2 to identify the highest probability tag sequence for the sentence

# How to start

- Find out: $\mathbf{P(t_i/w_i)}$ where $w_i$ is $h_i$ (context: $h_i=\{w_i, w_{i-1}, w_{i+1}, t_{i-1}\}$ )
- Formulation of $\mathbf{P(t_i/h_i) = exp(\sum \lambda_i f_i)/Z}$

where, z is normalization constant so that all text probabilities sum to 1. Each $\boldsymbol{\lambda_i}=1$

| **Noun** | **Adj** | **Noun** |
|---|---|---|
| | | |
| **Det** | **Verb** | **Verb** |
| | | |
| The | Light | book |
| | | |

# Now consider the word "the"

| Noun | Adj | Noun |
|---|---|---|
| $P(Noun/h_i) = e^1/ 2e^1 = 0.5$ | | |
| Det | Verb | Verb |
| $P(Det/h_i) = e^1/ 2e^1 = 0.5$ | | |
| The | Light | book |

1. For **the** as **Det** : only feature **f7 is true**, this feature becomes 1 for **the as Det.** Thus, **λf =1.**

   Therefore, **$P(Det/ h_i) = e^1/ Z$** (Z will be normalization constant for all possible values of **the**)

2. For **the** as **Noun**: only feature **f8 is true**, this feature becomes 1 for **the as Noun.** Thus, **λf =1.**

   Therefore, **$P(Noun/ h_i) = e^1/ Z$**

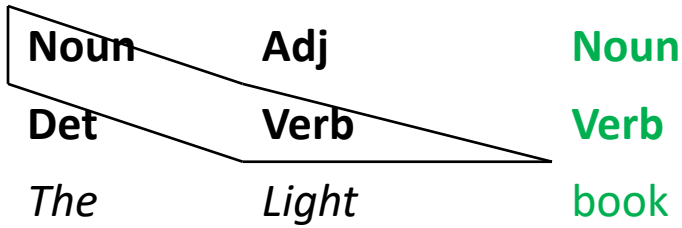
Now $Z = e^1 + e^1 = 2\ e^1$

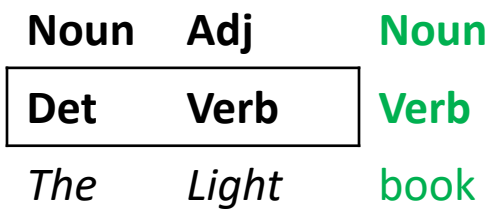
**$P(Det/h_i) = e^1/ 2e^1 = 0.5$ ; $P(Noun/h_i) = e^1/ 2e^1 = 0.5$**

# Now consider the word “light”

| Noun | Adj | Noun |
|---|---|---|
| P(Noun/$h_i$) = $e^1$/ $2e^1$ =0.5 | | |
| Det | Verb | Verb |
| P(Det/$h_i$) = $e^1$/ $2e^1$ = 0.5 | | |
| *The* | *Light* | book |

Now, P(Verb/ $h_i$). $h_i$={$w_i$, $w_{i-1}$, $w_{i+1}$, $t_{i-1}$} ). To compute P(Verb/ $h_i$) need to compute history (features like previous tag, previous word, etc), so need to talk in terms of sequences . **Consider the tag: verb for word:light**




The sequences are: **{Noun, verb} and { Det, Verb}**

1. **P(Noun, Verb)**= $\exp\sum\lambda_i f_i/Z$ * P(Noun)

For the feature f2 is true, so the λf =1.

P(N, V) = $e^1/Z$ * 0.5

2. **P(Det, Verb)** = $\exp\sum\lambda i fi/Z$ * P(Det)

P(D, V) = $e^0/Z$ * 0.5

3. **P(Noun, Adj)** = $\exp\sum\lambda_i f_i/Z$ * P(Noun)

For feature f4 & f5 is true, so λf =2.

P(N, Adj)=$e^2/Z$ *0.5

**For Noun Sequences, Z= $e^1$ + $e^2$**

4. **P(Det, Adj)= $\exp\sum\lambda ifi/Z$ * P(Det)**

For feature f1, f4 & f5 is true, so λf =3.

P(D, Adj) = $e^3/Z$ *P(Det) = $e^3/Z$ * 0.5

**For Det Sequences, Z= 1+ $e^3$**

- So we have four sequences:

1. **P(Noun, Verb) =** $e^1/Z * 0.5 = (e^1/ e^1 + e^2) * 0.5$
   $= (e^1/ e^3) * 0.5 = (1/ e^2) * 0.5 = (1/7.389)*0.5$
   **= 0.0677**
2. **P(Noun, Adj) =** $e^2/Z *0.5 = e^2/e^3 *0.5 = 1/e^1 *0.5 =$ $1/2.718 *0.5$ **= 0.1839**
3. **P(Det, Verb) =** $e^0/Z * 0.5 = 1/1+e^3 * 0.5$
   $=1/1+20.079 *0.5 =1/21.079$
   $*0.5=0.0474*0.5$**=0.0237**
4. **P(Det, Adj) =** $e^3/Z * 0.5 = (e^3/1+e^3) * 0.5 =$ $(20.079/ 21.079) *0.5 = 0.9525*0.5$ **= 0.4763**

**Top 2 highest probability sequences selected:**

P(Det, Adj) and P(N, Adj)

# Now consider the word **“book”** and tag **”Verb”**

| Noun | Adj | Noun |
|---|---|---|
| Det | Verb | Verb |
| *The* | *Light* | *book* |

Now, P(Verb/ $h_i$). $h_i$={$w_i$, $w_{i-1}$, $w_{i+1}$, $t_{i-1}$} ). To compute P(Verb/ $h_i$) need to compute history (features like previous tag, previous word, etc), so need to talk in terms of sequences . **Consider the tag: verb for word: book**

Earlier stage,

**Top 2 highest probability tag sequences selected:**

P(Det, Adj) = **0.4763** and P(Noun, Adj) = **0.1839**

**The sequences are:** { Det, Adj, Verb} and {Noun, Adj, Verb}

**We have four sequences :**

| | | |
|---|---|---|
| Det | Adj | Verb |
| Det | Adj | Noun |
| Noun | Adj | Verb |
| Noun | Adj | Noun |

P(Det, Adj) = **0.4763** and P(Noun, Adj) = **0.1839**

**1. P(Det, Adj, Verb)** = $\exp\sum\lambda_i f_i/Z$ * P(Det, Adj)

No feature is true, so the λf =0,

P(Det, Adj, V) = $e^0/Z$ * P(Det, Adj) = 1/Z * 0.4763

**2. P(Det, Adj, Noun)** = $\exp\sum\lambda_i f_i/Z$ * P(Det, Adj)

For the feature f3 and f6 is true, so the λf =2.

**P(Det, Adj, Noun) =** $e^2/Z$ * P(Det, Adj).

Now, Z= $1+e^2$

So,

**P(Det, Adj, Verb) = (**$1/1+e^2$ )* 0.4763 = 0.1192 *0.4763 =0.0568

**P(Det, Adj, Noun) = (** $e^2/1+e^2$ ) * 0.4763 = 0.8808 *0.4763 = 0.4195

P(Det, Adj) = **0.4763** and P(Noun, Adj) = **0.1839**

**3. P(Noun, Adj, Verb)** = $\exp\sum\lambda_i f_i/Z$ * P(Noun, Adj)

No feature is true, so the λf =0,

**P(Noun, Adj, Verb)** = $e^0/Z$ * 0.1839

4. **P(Noun, Adj, Noun) =** $\exp\sum\lambda i fi/Z$ * P(Noun, Adj)

For the feature f3 and f6 is true, so the λf =2.

**P(Noun, Adj, Noun) =** $e^2/Z$ * 0.1839

Now, Z= 1+ $e^2$

**So,**

**P(Noun, Adj, Verb) =** $1/1+e^2$ * 0.1839 = 0.1192 * 0.1839 = **0.0219**

**P(Noun, Adj, Noun) =** $e^2/1+e^2$ * 0.1839 = 0.8808 * 0.1839 = **0.1620**

# Probability values

| | | | |
|---|---|---|---|
| Det | Adj | Verb | 0.0568 |
| **Det** | **Adj** | **Noun** | **0.4195** |
| Noun | Adj | Verb | 0.0219 |
| Noun | Adj | Noun | 0.1620 |

**The best tag sequence due to MaxEnt tagger is with highest probability value :**

*The_**Det*** *light_**Adj*** book_**Noun** with probability **0.4195**