CS460/626 : Natural Language Processing/Speech, NLP and the Web
(Lecture 13, 14–Argmax Computation)

Pushpak Bhattacharyya
CSE Dept.,
IIT Bombay

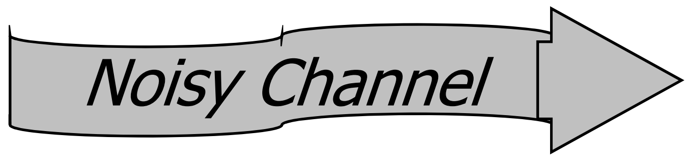
3rd and 7th Feb, 2011

Key difference between Statistical/ML-based NLP and Knowledge-based/linguistics-based NLP

- Stat NLP: speed and robustness are the main concerns
- KB NLP: Phenomena based
- Example:
  - *Boys, Toys, Toes*
  - To get the root *remove "s"*
  - How about *foxes, boxes, ladies*
  - Understand phenomena: go deeper
  - Slower processing

Noisy Channel Model

w



t

$(w_n, w_{n-1}, \ldots, w_1)$ $(t_m, t_{m-1}, \ldots, t_1)$

Sequence *w* is transformed into sequence *t*.

$$T^* = \underset{w}{\mathrm{argmax}}(P(T|W))$$

$$W^* = \underset{T}{\mathrm{argmax}}(P(W|T))$$

Bayesian Decision Theory

- Bayes Theorem : Given the random variables A and B,

$$P(A \mid B) = \frac{P(A)P(B \mid A)}{P(B)}$$

$P(A \mid B)$ Posterior probability

$P(A)$ Prior probability

$P(B \mid A)$ Likelihood

To understand when and why to apply Bayes Theorem

An example: *it is known that in a population, 1 in 50000 has meningitis and 1 in 20 has stiff neck. It is also observed that 50% of the meningitis patients have stiff neck.*

A doctor observes that a patient has stiff neck. What is the probability that the patient has meningitis?

(Mitchel, Machine Learning, 1997)

Ans: We need to find

P(m|s): probability of meningitis given the stiff neck

Apply Bayes Rule (why?)

$$P(m|s)$$

$$= \quad [P(m).\ P(s|m)]/P(s)$$

P(m)= prior probability of meningitis

P(s|m)=likelihod of stiff neck given meningitis

P(s)=Probability of stiff neck

Probabilities

$P(m)=$ $\frac{1}{50000}$ Prior

$P(s)=$ $\frac{1}{20}$

$P(s|m)= 0.5$ Likelihood

$$P(m|s)=\frac{P(m)P(s|m)}{P(s)}=\frac{\frac{1}{50000}*0.5}{\frac{1}{20}}=\frac{1}{5000}$$

posterior

$$P(m|s) \ll P(\sim m|s)$$

Hence meningitis is not likely

Some Issues

- *p(m/s)* could have been found as

$$\frac{\#(m \cap s)}{\# s}$$

Questions:

- Which is more reliable to compute, *p(s/m)* or *p(m/s)*?
- Which evidence is more sparse , *p(s/m)* or *p(m/s)*?
- Test of significance : The counts are always on a sample of population. Which probability count has sufficient statistics?

5 problems in NLP whose probabilistic formulation use Bayes theorem

The problems

- Statistical Spell Checking
- Automatic Speech Recognition
- Part of Speech Tagging: *discussed in detail in subsequent classes*
- Probabilistic Parsing
- Statistical Machine Translation

Some general observations

$$A^* = \underset{A}{\operatorname{argmax}}\,[P(A|B)]$$

$$= \underset{A}{\operatorname{argmax}}\,[P(A).P(B|A)]$$

Computing and using P(A) and P(B|A), both need

(i) *looking at the* ***internal structures*** *of A and B*

(ii) *making independence assumptions*

(iii) *putting together a computation from smaller parts*

Corpus

- A collection of text called *corpus*, is used for collecting various language data
- With annotation: more information, but manual labor intensive
- Practice: *label automatically; correct manually*
- The famous *Brown Corpus* contains 1 million tagged words.
- **Switchboard:** very famous corpora 2400 conversations, 543 speakers, many US dialects, annotated with orthography and phonetics

Example-1 of Application of Noisy Channel Model: Probabilistic Speech Recognition (Isolated Word)[8]

- **Problem Definition : Given a sequence of speech signals, identify the words.**
- **2 steps :**
  - **Segmentation (Word Boundary Detection)**
  - **Identify the word**
- **Isolated Word Recognition :**
  - **Identify W given SS (speech signal)**

$$\hat{W} = \arg\max_{W} P(W \mid SS)$$

Identifying the word

$$\hat{W} = \arg\max_{W} P(W \mid SS)$$
$$= \arg\max_{W} P(W)P(SS \mid W)$$

- *P(SS/W)* = likelihood called "phonological model " → intuitively more tractable!
- *P(W)* = prior probability called "language model"

$$P(W) = \frac{\text{\# W appears in the corpus}}{\text{\# words in the corpus}}$$

Ambiguities in the context of P(SS|W) or P(W|SS)

- Concerns
  - Sound → Text ambiguity
    - *whether* v/s *weather*
    - *right* v/s *write*
    - *bought* v/s *bot*
  - Text → Sound ambiguity
    - *read* (present tense) v/s *read* (past tense)
    - *lead* (verb) v/s *lead* (noun)

Primitives

- Phonemes (sound)
- Syllables
- ASCII bytes (machine representation)

Phonemes

- Standardized by the IPA (International Phonetic Alphabet) convention
- */t/* →sound of *t* in *tag*
- */d/* →sound of *d* in *dog*
- */D/* →sound of *the*

Syllables

Advise (verb)

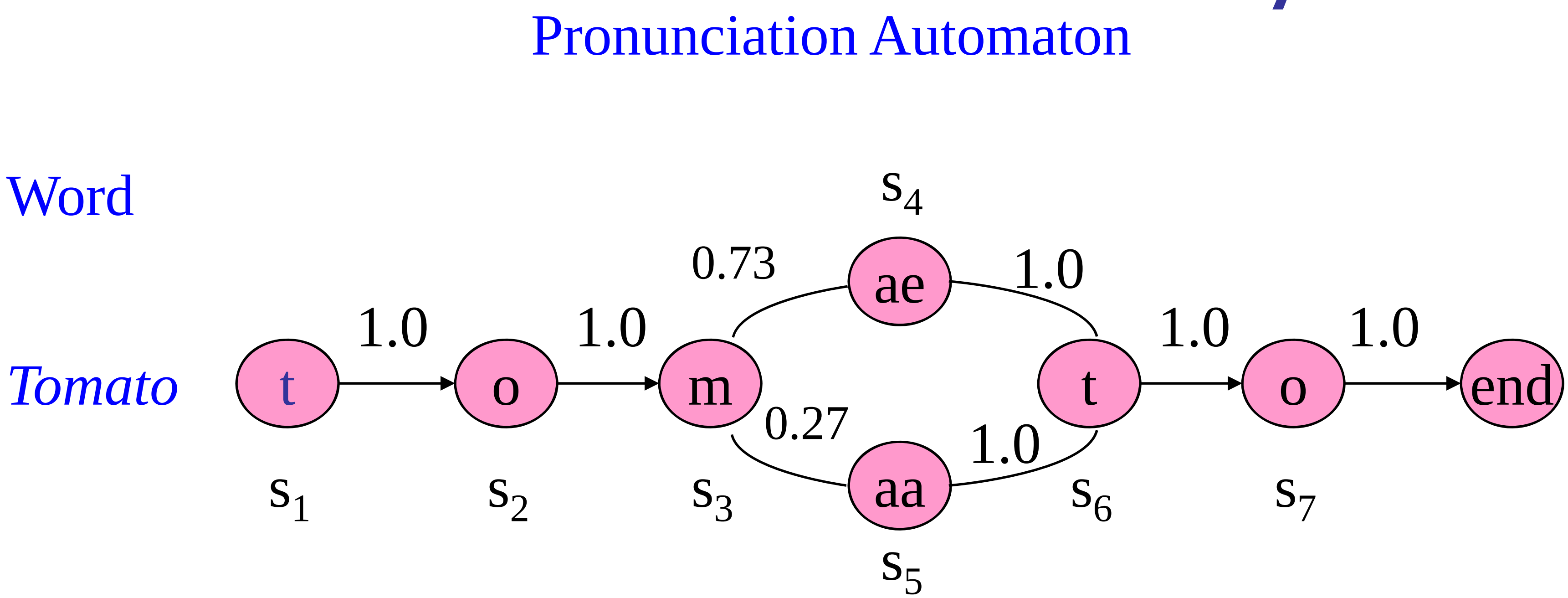
ad vise

Advice (noun)

ad vice

- Consists of
  1. Rhyme
     1. Nucleus
     2. Onset
     3. Coda

Pronunciation Dictionary



- *P(SS|W)* is maintained in this way.
- *P(t o m ae t o |Word is "tomato")* = Product of arc probabilities

Problem 2: Spell checker: apply Bayes Rule

$$W^* = \arg\max [P(W|T)]$$
$$= \arg\max [P(W).P(T|W)]$$

W=correct word, T=misspelt word

- Why apply Bayes rule?
  - Finding $p(w/t)$ vs. $p(t/w)$?
- Assumptions :
  - t is obtained from w by a single error.
  - The words consist of only alphabets

(Jurafsky and Martin, Speech and NLP, 2000)

4 Confusion Matrices: *sub, ins, del* and *trans*

- If x and y are alphabets,
  - sub(x,y) = # times y is written for x (substitution)
  - ins(x,y) = # times x is written as xy
  - del(x,y) = # times xy is written as x
  - trans(x,y) = # times xy is written as yx

Probabilities from confusion matrix

- $P(t|w) = P(t|w)_S + P(t|w)_I + P(t|w)_D + P(t|w)_X$

where

$P(t|w)_S$ = sub(x,y) / count of x

$P(t|w)_I$ = ins(x,y) / count of x

$P(t|w)_D$ = del(x,y) / count of x

$P(t|w)_X$ = trans(x,y) / count of x

- These are considered to be mutually exclusive events

URLs for database of misspelt words

- http://www.wsu.edu/~brians/errors/errors.html
- http://en.wikipedia.org/wiki/Wikipedia:Lists_of_common_misspellings/For_machines

A sample

- abandonned->abandoned
- aberation->aberration
- abilties->abilities
- abilty->ability
- abondon->abandon
- abondoned->abandoned
- abondoning->abandoning
- abondons->abandons
- aborigene->aborigine

fi yuo cna raed tihs, yuo hvae a sgtrane mnid too.
Cna yuo raed tihs? Olny 55 plepoe can.

i cdnuolt blveiee taht I cluod aulaclty uesdnatnrd waht I was rdanieg.
The phaonmneal pweor of the hmuan mnid, aoccdrnig to a rscheearch at
Cmabrigde Uinervtisy, it dseno't mtaetr in waht oerdr the ltteres in a wrod are, the olny iproamtnt tihng is taht the frsit and lsat ltteer be in the rghit pclae. The rset can be a taotl mses and you can sitll raed
it whotuit a pboerlm. Tihs is bcuseae the huamn mnid deos not raed
ervey lteter by istlef, but the wrod as a wlohe. Azanmig huh? yaeh and
I

awlyas tghuhot slpeling was ipmorantt! if you can raed tihs forwrad it.

Spell checking: Example

- Given *aple*, find and rank
  - P(maple|aple), P(apple|aple), P(able|aple), P(pale|aple) etc.

- *Exercise*: *Give an intuitive feel for which of these will rank higher*

Example 3: Part of Speech Tagging

- POS Tagging is a process that attaches each word in a sentence with a suitable tag from a given set of tags.
- The set of tags is called the Tag-set.
- Standard Tag-set : Penn Treebank (for English).

Penn Treebank Tagset: sample

- 1. CC Coordinating conjunction; *Jack and_CC Jill*
- 2. CD Cardinal number; *Four_CD children*
- 3. DT Determiner; *The_DT sky*
- 4. EX Existential *there ; There_EX was a king*
- 5. FW Foreign word; *शब्द_FW means 'word'*
- 6. IN Preposition or subordinating conjunction; *play with_IN ball*
- 7. JJ Adjective; *fast_JJ car*
- 8. JJR Adjective, comparative; *faster_JJR car*
- 9. JJS Adjective, superlative; *fastest_JJS car*
- 10. LS List item marker; *1._LS bread 2._LS butter 3._LS Jam*
- 11. MD Modal; *You may_MD go*
- 12. NN Noun, singular or mass; *water_NN*
- 13. NNS Noun, plural; *boys_NNS*
- 4. NNP Proper noun, singular; *John_NNP*

POS Tags

- NN – Noun; e.g. *Dog_NN*
- VM – Main Verb; e.g. *Run_VM*
- VAUX – Auxiliary Verb; e.g. *Is_VAUX*
- JJ – Adjective; e.g. *Red_JJ*
- PRP – Pronoun; e.g. *You_PRP*
- NNP – Proper Noun; e.g. *John_NNP*
- etc.

POS Tag Ambiguity

- In English : I $bank_1$ on the $bank_2$ on the river $bank_3$ for my transactions.
  - $Bank_1$ is verb, the other two banks are noun
- In Hindi :
  - ”Khaanaa” : can be noun (food) or verb (to eat)
  - Mujhe khaanaa khaanaa hai. (first khaanaa is noun and second is verb)

For Hindi

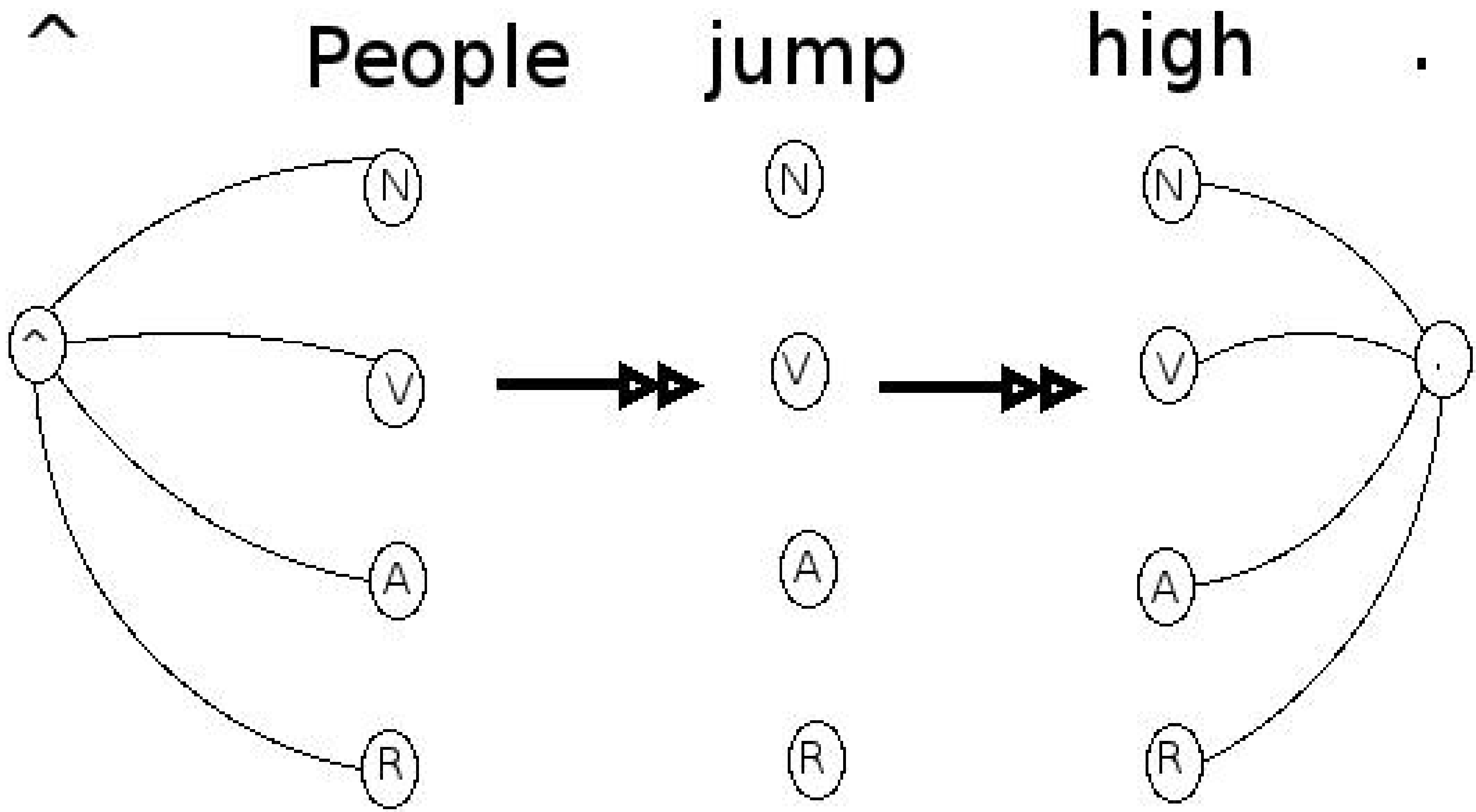
- *Rama achhaa gaata hai*. (hai is VAUX : Auxiliary verb); *Ram sings well*
- *Rama achha ladakaa hai*. (hai is VCOP : Copula verb); *Ram is a good boy*

Process

- List all possible tag for each word in sentence.
- Choose best suitable tag sequence.

Example

- ”People jump high”.
- People : Noun/Verb
- jump : Noun/Verb
- high : Noun/Verb/Adjective
- We can start with probabilities.



^
People
jump
high
.
N
V
A
R

Derivation of POS tagging formula

Best tag sequence

= T*

= argmax P(T|W)

= argmax P(T)P(W|T)   (by Baye's Theorem)

$P(T) = P(t_0{=}\hat{}\ t_1 t_2 \ldots t_{n+1}{=}.)$

$= P(t_0)P(t_1|t_0)P(t_2|t_1 t_0)P(t_3|t_2 t_1 t_0) \ldots$

$P(t_n|t_{n-1} t_{n-2} \ldots t_0)P(t_{n+1}|t_n t_{n-1} \ldots t_0)$

$= P(t_0)P(t_1|t_0)P(t_2|t_1) \ldots P(t_n|t_{n-1})P(t_{n+1}|t_n)$

$= \prod_{i=1}^{N+1} P(t_i|t_{i-1})$ Bigram Assumption

Lexical Probability Assumption

$P(W|T) = P(w_0|t_0\text{-}t_{n+1})P(w_1|w_0t_0\text{-}t_{n+1})P(w_2|w_1w_0t_0\text{-}t_{n+1}) \ldots$
$P(w_n|w_0\text{-}w_{n-1}t_0\text{-}t_{n+1})P(w_{n+1}|w_0\text{-}w_nt_0\text{-}t_{n+1})$

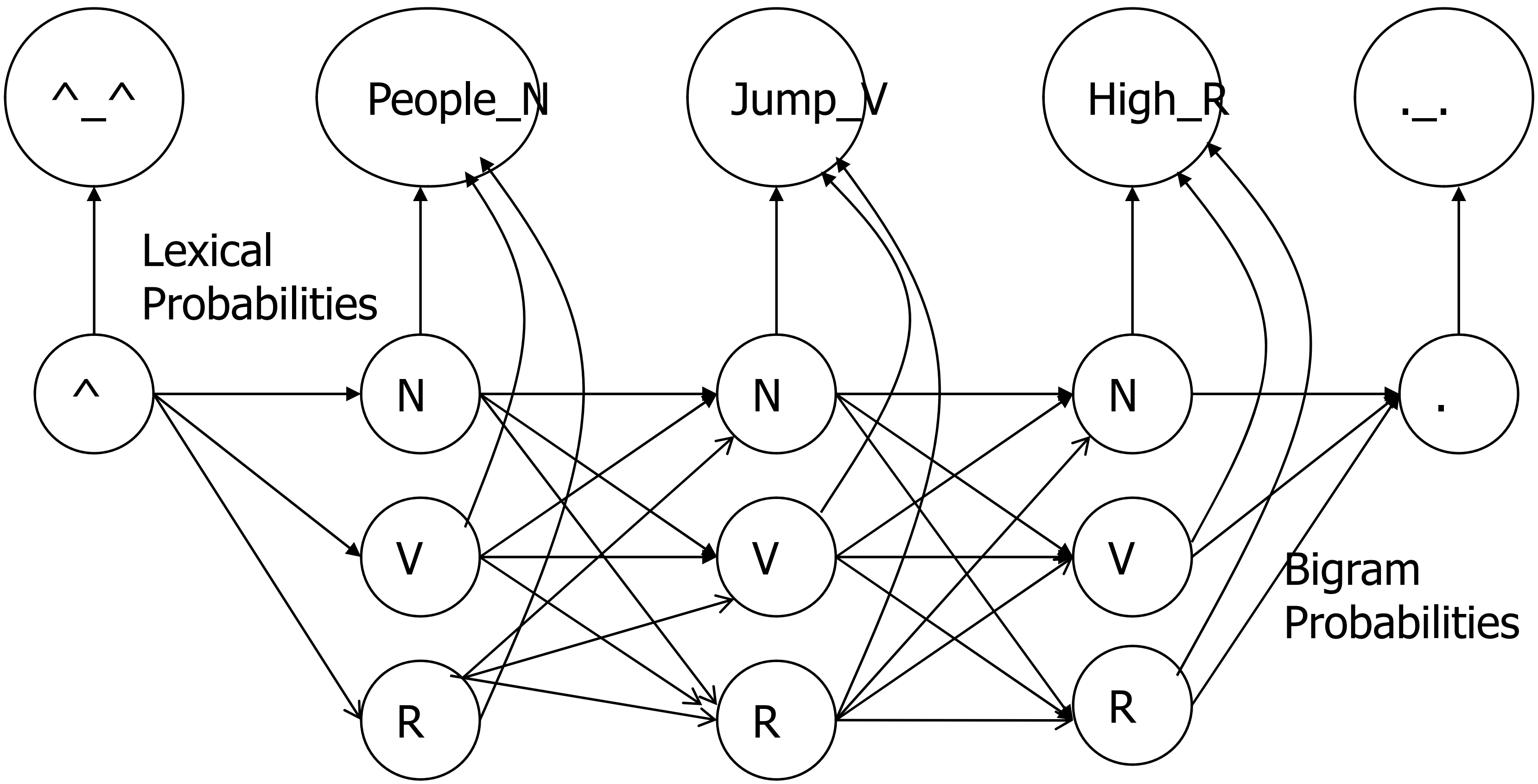
Assumption: A word is determined completely by its tag. This is inspired by speech recognition

$$= P(w_o|t_o)P(w_1|t_1) \ldots P(w_{n+1}|t_{n+1})$$

$$= \prod_{i=0}^{n+1} P(w_i|t_i)$$

$$= \prod_{i=0}^{n+1} P(w_i|t_i) \qquad \text{(Lexical Probability Assumption)}$$

Generative Model



This model is called Generative model.
Here words are observed from tags as states.
This is similar to HMM.

Bigram probabilities

-

	N	*V*	*A*
N	0.2	0.7	0.1
V	0.6	0.2	0.2
A	0.5	0.2	0.3

Lexical Probability

	People	jump	high
N	10^{-5}	$0.4x10^{-3}$	10^{-7}
V	10^{-7}	10^{-2}	10^{-7}
A	0	0	10^{-1}

  values in cell are P(col-heading/row-heading)

Calculation from actual data

- Corpus
  - *^ Ram got many NLP books. He found them all very interesting.*
- Pos Tagged
  - *^ N V A N N . N V N A R A .*

Recording numbers

	^	N	V	A	R	.
^	0	2	0	0	0	0
N	0	1	2	1	0	1
V	0	1	0	1	0	0
A	0	1	0	0	1	1
R	0	0	0	1	0	0
.	1	0	0	0	0	0

Probabilities

	^	N	V	A	R	.
^	0	1	0	0	0	0
N	0	1/5	2/5	1/5	0	1/5
V	0	1/2	0	1/2	0	0
A	0	1/3	0	0	1/3	1/3
R	0	0	0	1	0	0
.	1	0	0	0	0	0

To find

- $T^* = \text{argmax}\,(P(T)\,P(W/T))$
- $P(T).P(W/T) = \prod_{i=1 \rightarrow n+1} P(\,t_i\,/\,t_{i-1}\,).P(w_i\,/t_i)$
- $P(\,t_i\,/\,t_{i-1}\,)$: *Bigram probability*
- $P(w_i\,/t_i)$: *Lexical probability*
- Qrwh: $P(w_i/t_i)=1$ iru $i=0$ +a / vhqwhqfh ehjlqqhu,, dqg $i=(n+1)$ *(., fullstop)*

Bigram probabilities

-

	N	***V***	***A***	***R***
N	0.15	0.7	0.05	0.1
V	0.6	0.2	0.1	0.1
A	0.5	0.2	0.3	0
R	0.1	0.3	0.5	0.1

Lexical Probability

	People	jump	high
N	10^{-5}	0.4×10^{-3}	10^{-7}
V	10^{-7}	10^{-2}	10^{-7}
A	0	0	10^{-1}
R	0	0	0

values in cell are P(col-heading/row-heading)

Some notable text corpora of English

- American National Corpus
- Bank of English
- British National Corpus
- Corpus Juris Secundum
- Corpus of Contemporary American English (COCA) 400+ million words, 1990-present. Freely searchable online.
- Brown Corpus, forming part of the "Brown Family" of corpora, together with LOB, Frown and F-LOB.
- International Corpus of English
- Oxford English Corpus
- Scottish Corpus of Texts & Speech