CS626: Speech, NLP and the Web

Shallow Parsing with Conditional Random Field, Morphology Brief

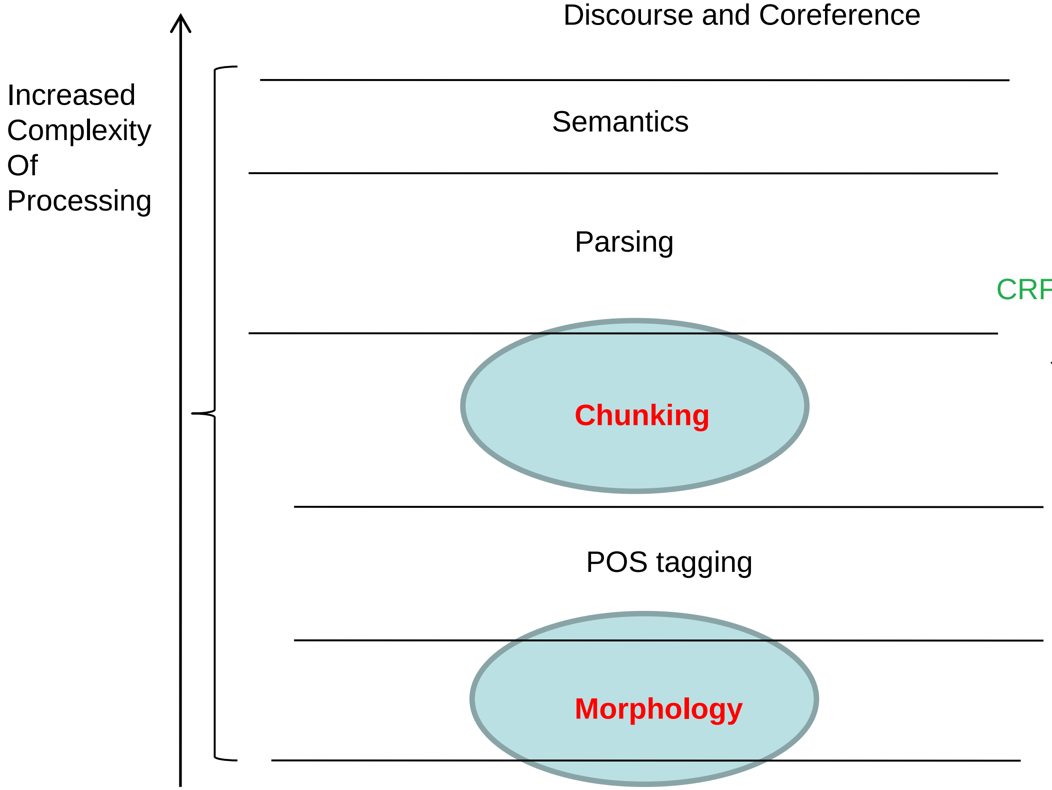
Pushpak Bhattacharyya

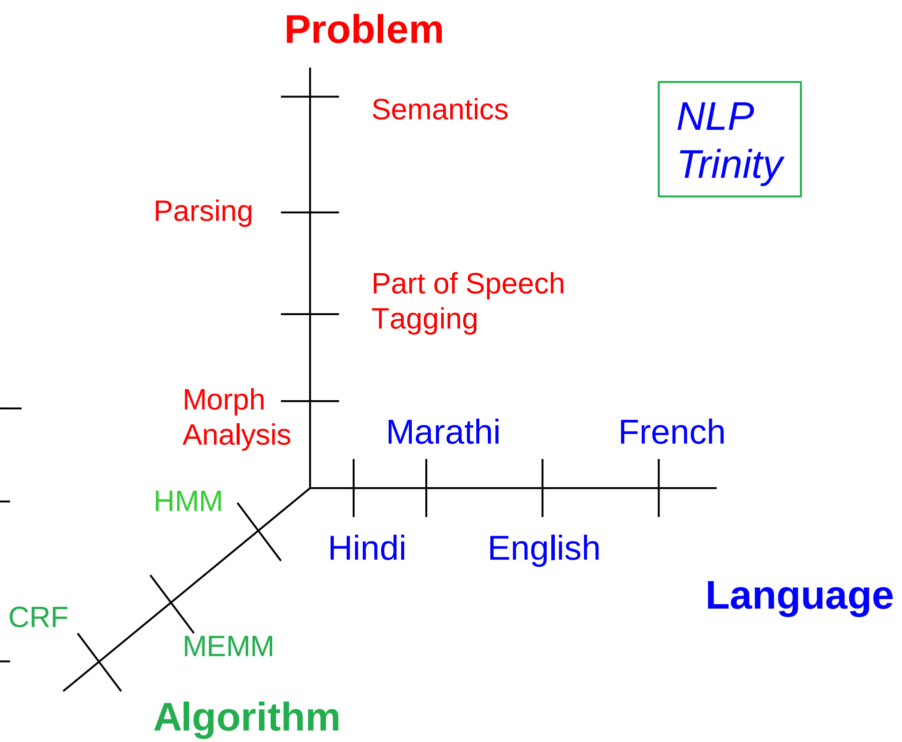
Computer Science and Engineering Department

IIT Bombay

Week of 17th August, 2020

NLP: multilayered, multidimensional





Agenda for the week (1/2)

- Define and solve detecting chunks/shallow_parses
  - Base Pharses/non-recursive phases
  - Using CRF (John Lafferty, Andrew McCallum, and Fernando C.N. Pereira, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data", ICML 2001. https://repository.upenn.edu/cgi/viewcontent.cgi?article=1162&context=cis_papers

Agenda for the week (2/2)

- Support from morphology
  - Data sparsity can be solved by looking inside words
  - NLP Stack backoff
  - “proposition”→ NN because of ‘tion’
  - “abruptly” → RB (adverb) because of ‘ly’
  - Should be weighed against evidence from other features (previous tag)
- Evaluation of POS tagging (and in general of any sequences)

Evaluation of sequence to sequence labelling

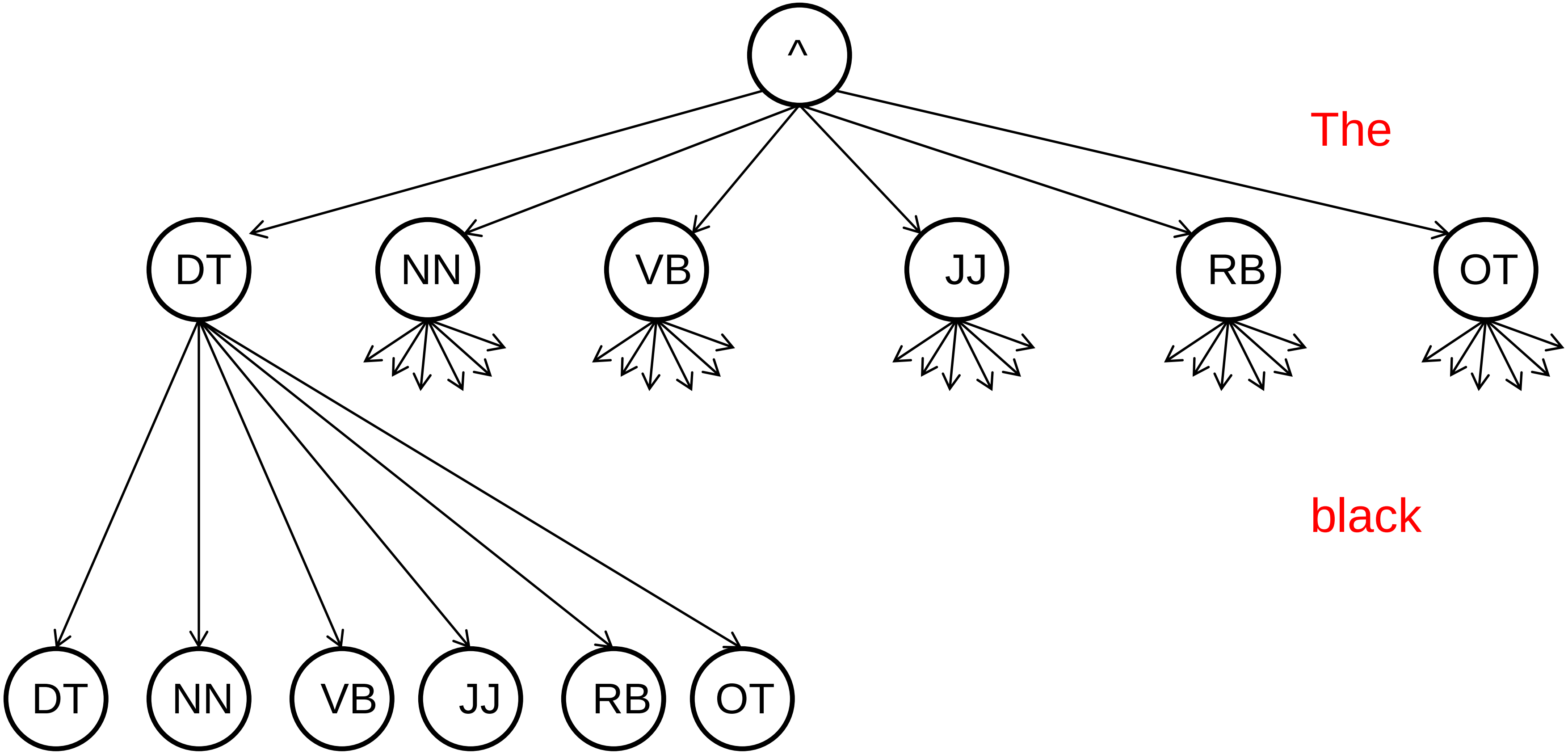
POS Tagging Example

- Suppose our tags are
  - DT, NN, VB, JJ, RB and OT
- E.g.

DT- determiner
NN- Noun
VB- Verb
JJ- Adjective
RB- Adverb
OT- others

^	The	black	dog	barks	.
^	DT	DT	DT	DT	.
	NN	NN	NN	NN	
	VB	VB	VB	VB	
	JJ	JJ	JJ	JJ	
	RB	RB	RB	RB	
	OT	OT	OT	OT	

Possible tags



dog

Correct: ^_^ The_DT black_JJ dog_NN barks_VB ._.

barks

Incorrect: ^_^ The_DT black_NN dog_VB barks_VB ._.

Precision

- **^_^ The_DT black_NN dog_VB barks_VB ._.**

- 4 out of 6 correct

- Precision= 66.67%

- True for population?

Question

- The POS tagger I built, will it for all time to come function with 66.67% precision
- That is, will it on ***an average*** tag 67% of the words correctly?
- That is, one an average, 20 out of every ***sample*** of 30 words sequences be correct?

Precision question similar to Coin Tossing Problem

- *X1_H X2_H X3_T X4_T X5_T…*
- Suppose is *H* is "correct" and *T* "incorrect"

- Then *"Precision"*= *K/N*, where *#H=K* and *#Tosses=N*

We are in the realm of Bernoulli Trial and Binomial Distribution

- Probability of *K* successes in *N* Bernoulli Trials with probability of success being *p* in each trial is given as

$$\Pr(K; N; p) = {}^{N}C_{K}\, p^{K} (1-p)^{N-K}$$

Normal Approximation to Binomial

- The normal distribution can be used as an approximation to the binomial distribution under certain circumstances
- Namely: If $\boldsymbol{X \sim B(n, p)}$ and if n is large and/or p is close to ½, then X is approximately $N(np, npq)$, *i.e.*, normal with mean np and standard deviation npq, where $q=1-p$

Now, we are in the realm of Normal!

- Use the machinery of normal distribution
- Can use 95% confidence interval as well as *np* and *npq* to estimate test data requirement

- Of course, *p* is a function of training efficacy

Morphology

Acknowlegement:

Mugdha Bapat, ex-M.Tech student, CFILT, CSE

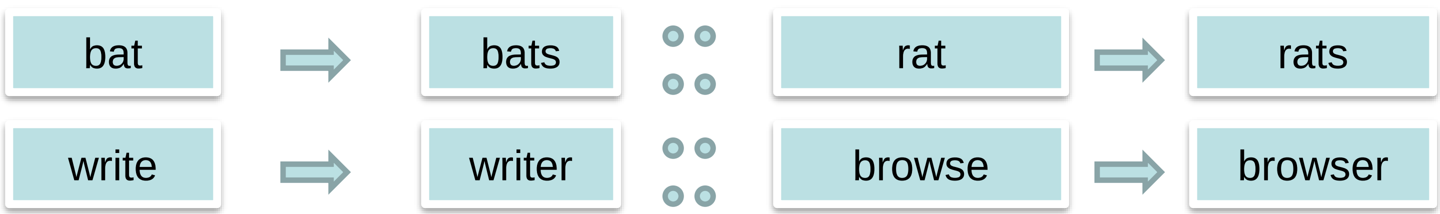
Based on:

Akmajian et al, *LINGUISTICS An Introduction to Language and Communication*, 7th edition, MIT Press, 2017

What is Morphology?

- Study of Words
- Their internal structure

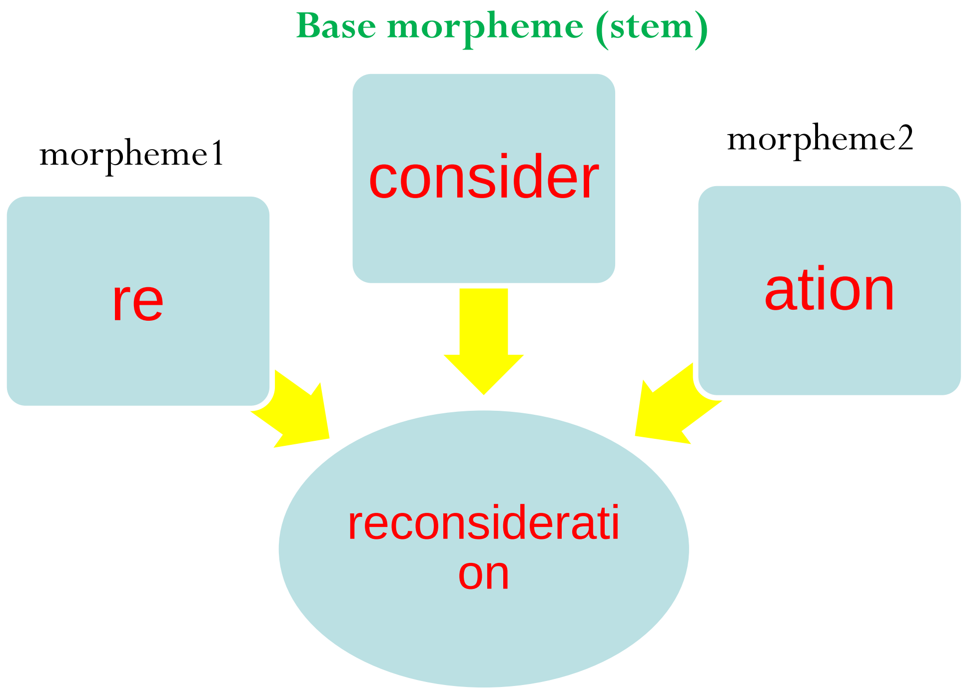
- How they are formed?



- Morphology tries to formulate rules that show the knowledge of the speakers of those languages

Morphemes

- Smallest linguistic pieces with a grammatical function



Accuracy vs. data size: general POS and Chunk

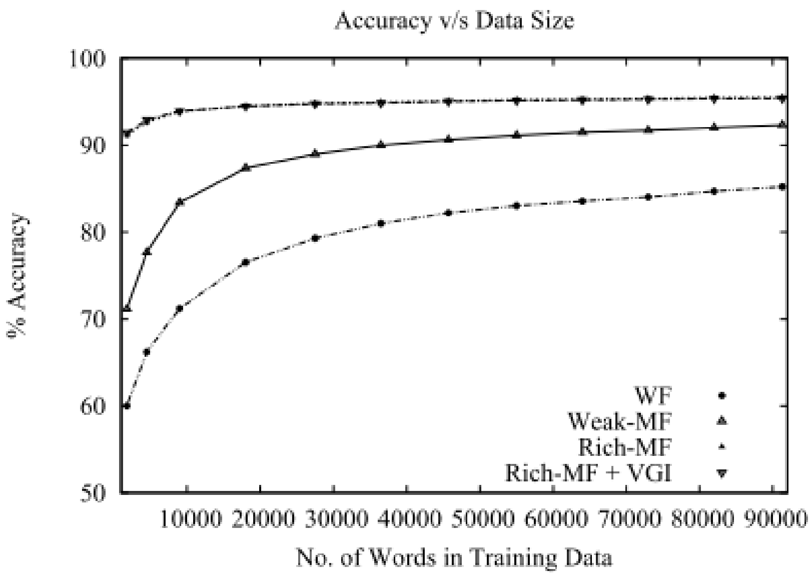


Figure 3: Average Accuracy of all POS Tags
(Note: The graphs for Rich-MF and Rich-MF+VGI coincide)

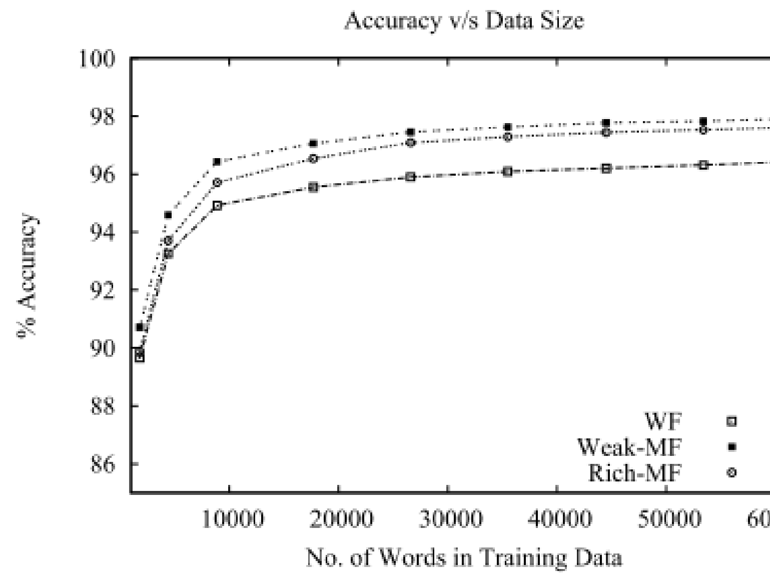


Figure 6: Average Accuracy of all Chunk Ta

Harshada Gune, Mugdha Bapat, Mitesh Khapra and Pushpak Bhattacharyya, Verbs are where all the Action Lies: Experiences of Shallow Parsing of a Morphologically Rich Language, Computational Linguistics Conference (COLING 2010), Beijing, China, August 2010.

Verb POS and Verb Chunk

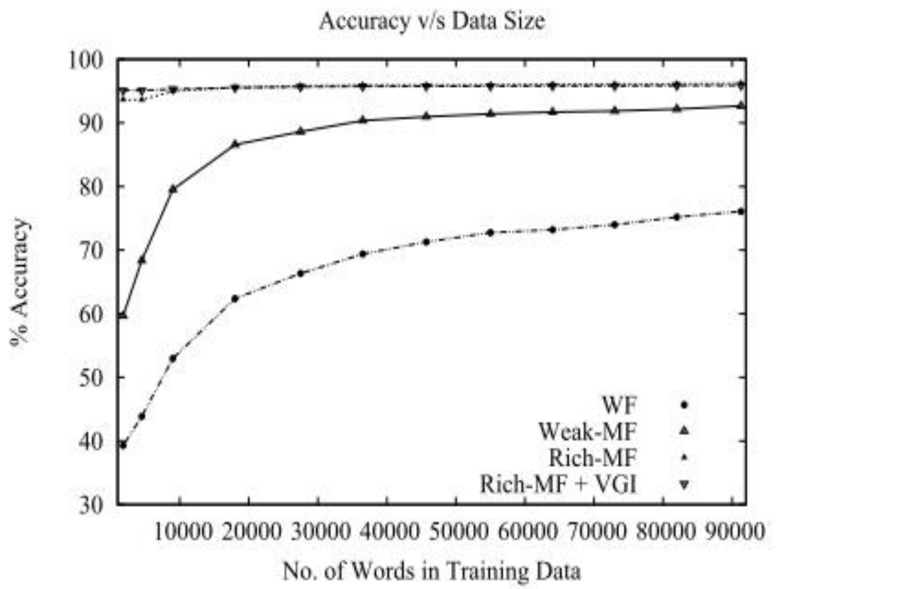


Figure 4: Average Accuracy of Verb POS Tags

(Note: The graphs for Rich-MF and Rich-MF+VGI almost coincide)

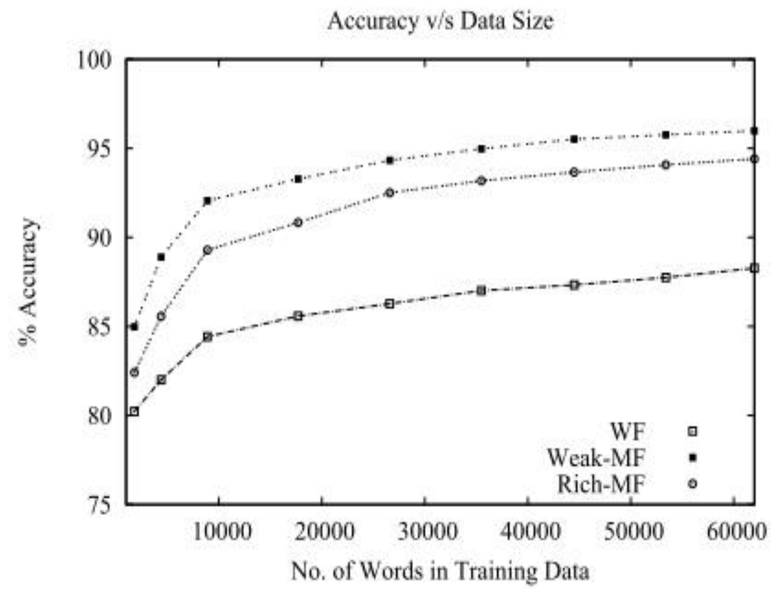


Figure 7: Average Accuracy of Verb Chunks

Harshada Gune, Mugdha Bapat, Mitesh Khapra and Pushpak Bhattacharyya, Verbs are where all the Action Lies: Experiences of Shallow Parsing of a Morphologically Rich Language, Computational Linguistics Conference (COLING 2010), Beijing, China, August 2010.

Non veb POS and Non Verb Chunk

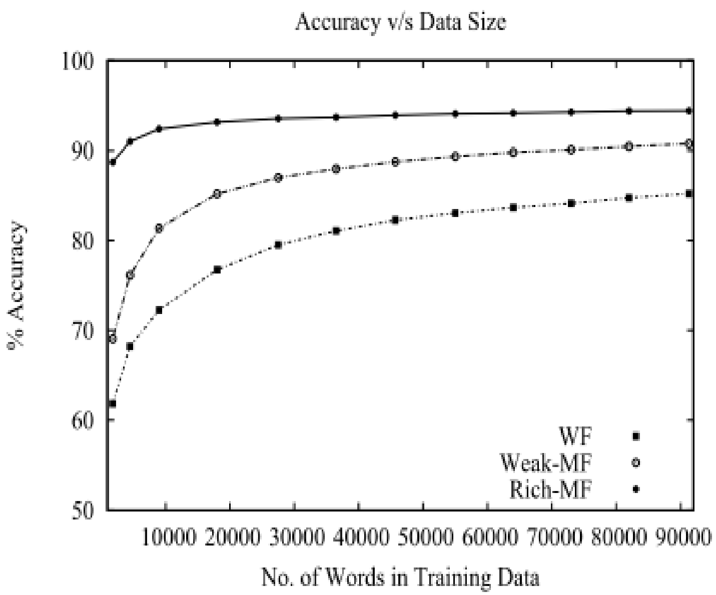


Figure 5: Average Accuracy of Non Verb POS Tags

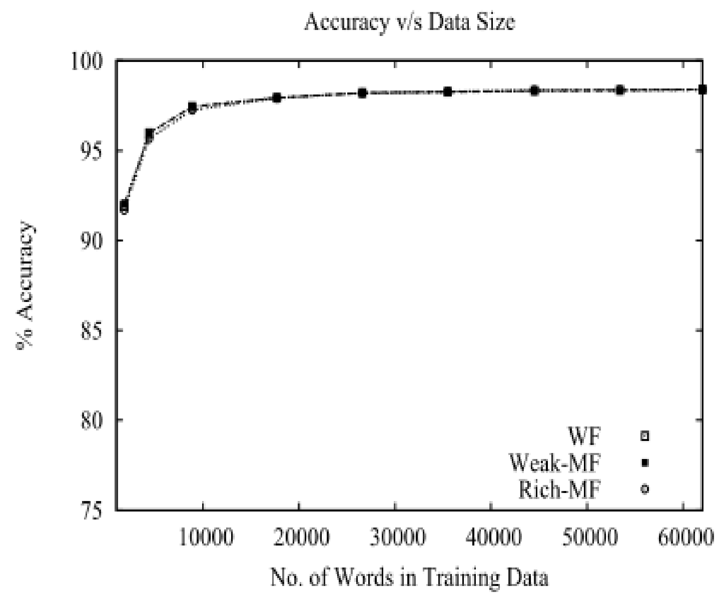
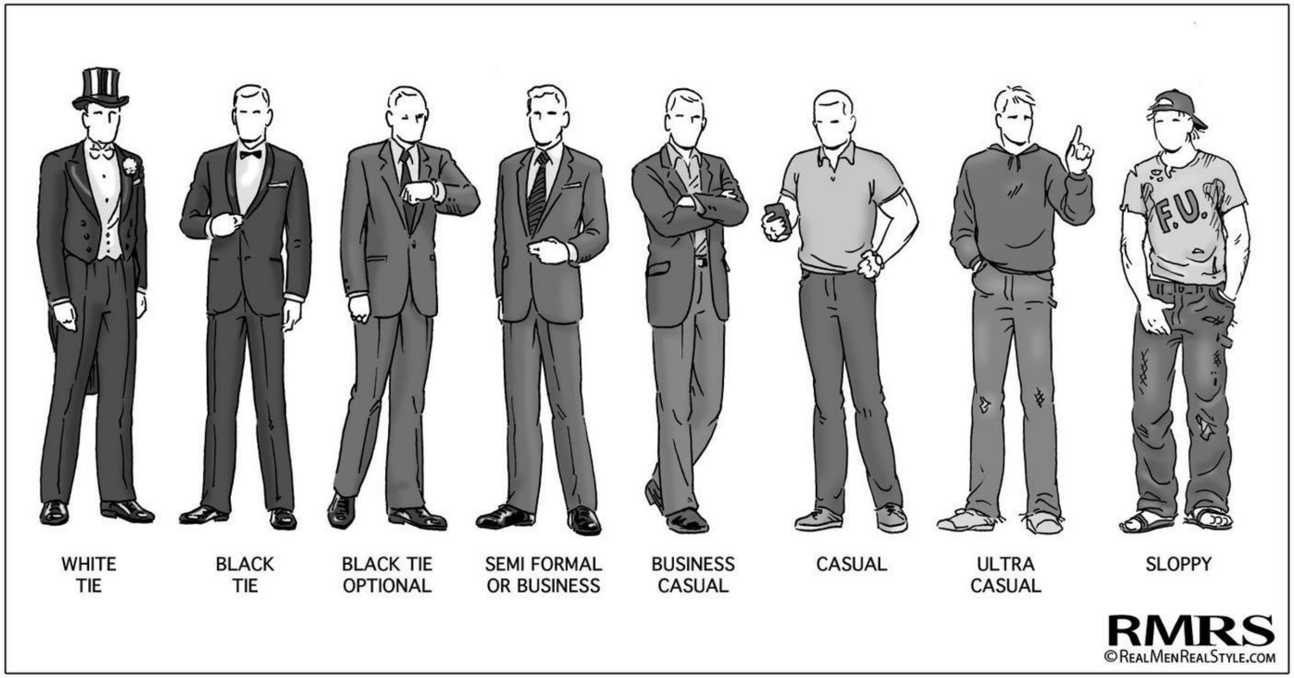


Figure 8: Average Accuracy of Non Verb Chunks
(Note: All the graphs coincide.)

Harshada Gune, Mugdha Bapat, Mitesh Khapra and Pushpak Bhattacharyya, Verbs are where all the Action Lies: Experiences of Shallow Parsing of a Morphologically Rich Language, Computational Linguistics Conference (COLING 2010), Beijing, China, August 2010.

Rich morphology vs. poor morphology: analogy



Verb conjugation: Gender Number Person Tense Aspect Modality: **GNPTAM**

jaanaa: jaauMgaa, jaaoge, jaayeMge …

English:

Will go

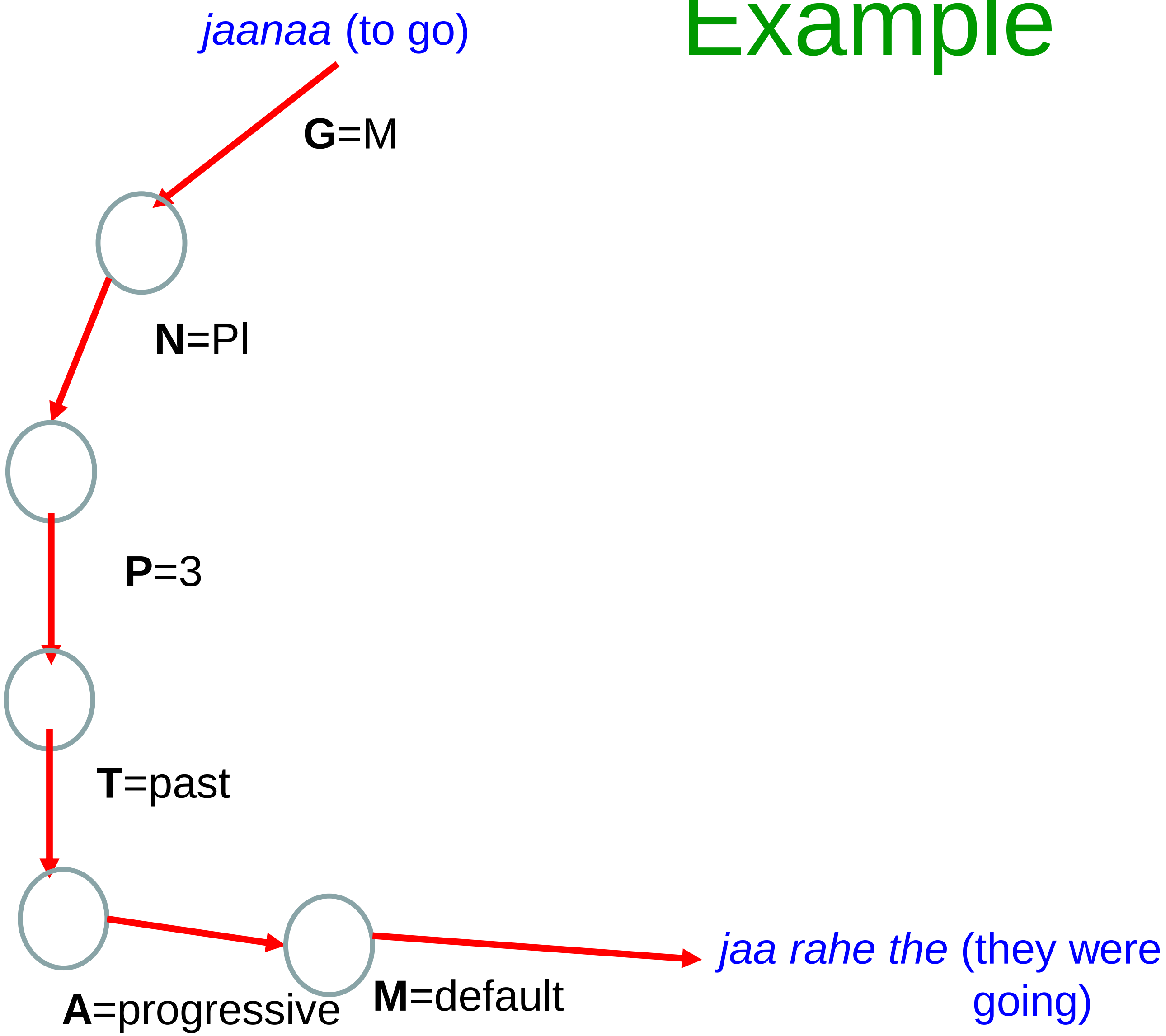
Combinatorics of Morphology: Verb Conjugation

- Gender (**G**)- 3 (M,F, N; 2 for Hindi)
- Number (**N**)- 2 (S, P; 3 for Sanskrit and other ancient languages: dual)
- Person (**P**)- 3 (1p, 2p, 3p)
- Tense (**T**)- 3 (past, present, future)
- Aspect (**A**)- 3 (progressive, perfect, Default)
- Modality (**M**)- 4 (declarative, Imperative, Interrogative, Exclamation)

Combinatorics

- #possibilities (GNPTAM)- 3 X 2 X 3 X 3 X 3 X 4= 648
- Given a verb root (also called stem), 648 forms

Example


jaanaa (to go)
G=M
N=Pl
P=3
T=past
A=progressive
M=default
jaa rahe the (they were going)

More combinatorics

- Typically about 30% of the lexical repository of any language is verbs
- Assuming the lexicon size to be 100,000
- There are 30,000 verbs
- If unambiguous morphology existed, then we would have 30000 X 648 verb forms=

  ~ 20 million or 2 crore verb forms

Reflections on morphology combinatorics

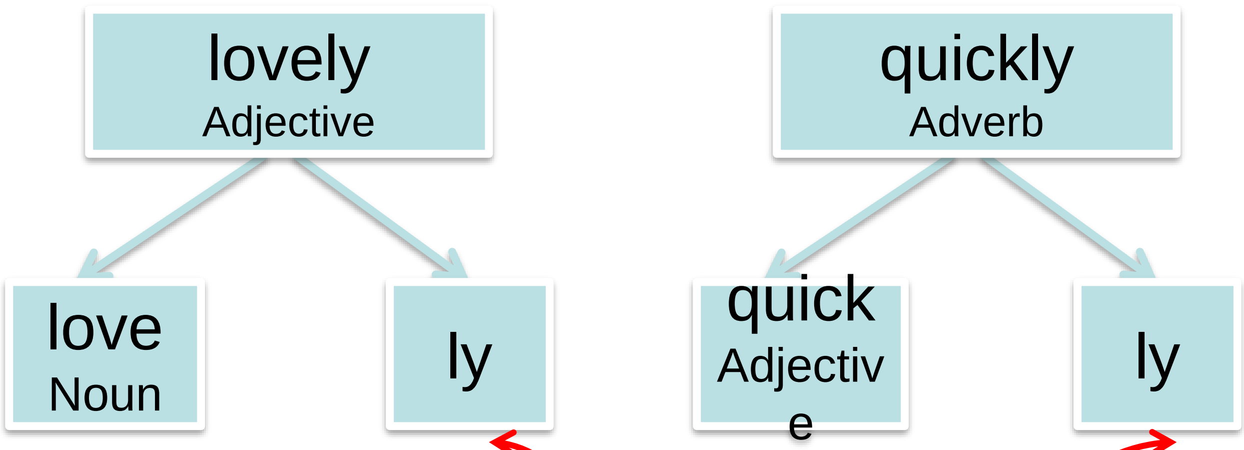
- Could have been a blow up of about 650 times
- Only verb forms occurring by themselves could give rise to a 20 million words corpora
- Combinatorial blow up does not happen
- Why?

Phenomena that control morphological combinatorial explosion

- Syncretism- overloading of forms
  - *Will go*
  - *G=M/F, N-S/Pl, P-1/2/3, T-Fut, A-Default, M-Declarative*
- Many verbs occur rarely, e.g., *perambulating (English), curvetting (English), batiyana, drumaayate (Sanskrit), kingkartavyabimur (bangla)*

More about Morphemes

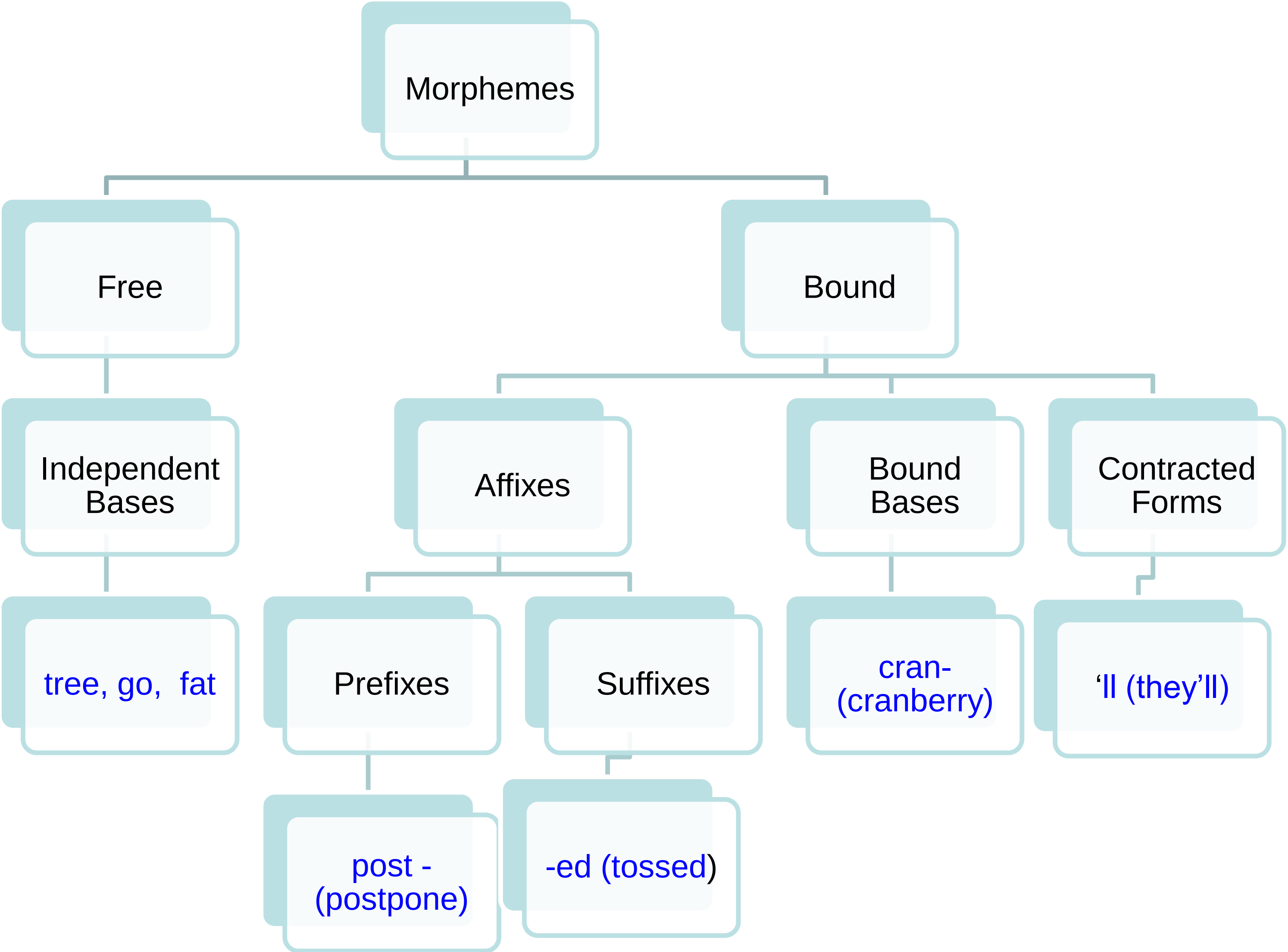
- Grammatical function of a morpheme must be *constant*



two different morphemes!

But same function- **manner-of/state-of**

Basic classification of English Morphemes



Infix: A type of affix- inside a word

In the language *Bonto Igorot*

- The infix ‘in’ is used to
- indicate a *completed product*

Sanskrit

raajaayate:

raajaa+ya+te

‘ya’ is infix (behaves like a king)

Original word: kayu	**Complex word: k*in*ayu**
Meaning: wood	Meaning: gathered wood

Morphology & Grammatical Categories

- Morphology as evidence for classification

English Nouns
- Inflect for number

English Adjectives
- Do not inflect for number

English Verbs
- Inflect for tense

English Nouns
- Do not inflect for tense

Classification of Free Morphemes

Open-class words, aka Content Words	**Closed-class words, aka function words**
Large in number	Small in number (include fixed elements)
Open-ended: Unlimited number of new words can be created and added	Addition of a new word to this class is very rare event
Grammatical categories that fall in this class: 1. Nouns 2. Verbs 3. Adjectives 4. Adverbs	Grammatical categories that fall in this class: 1. Conjunctions 2. Articles 3. Demonstratives 4. Prepositions 5. Comparatives 6. Quantifiers

Morphology

- Derivational Morphology
- Inflectional Morphology

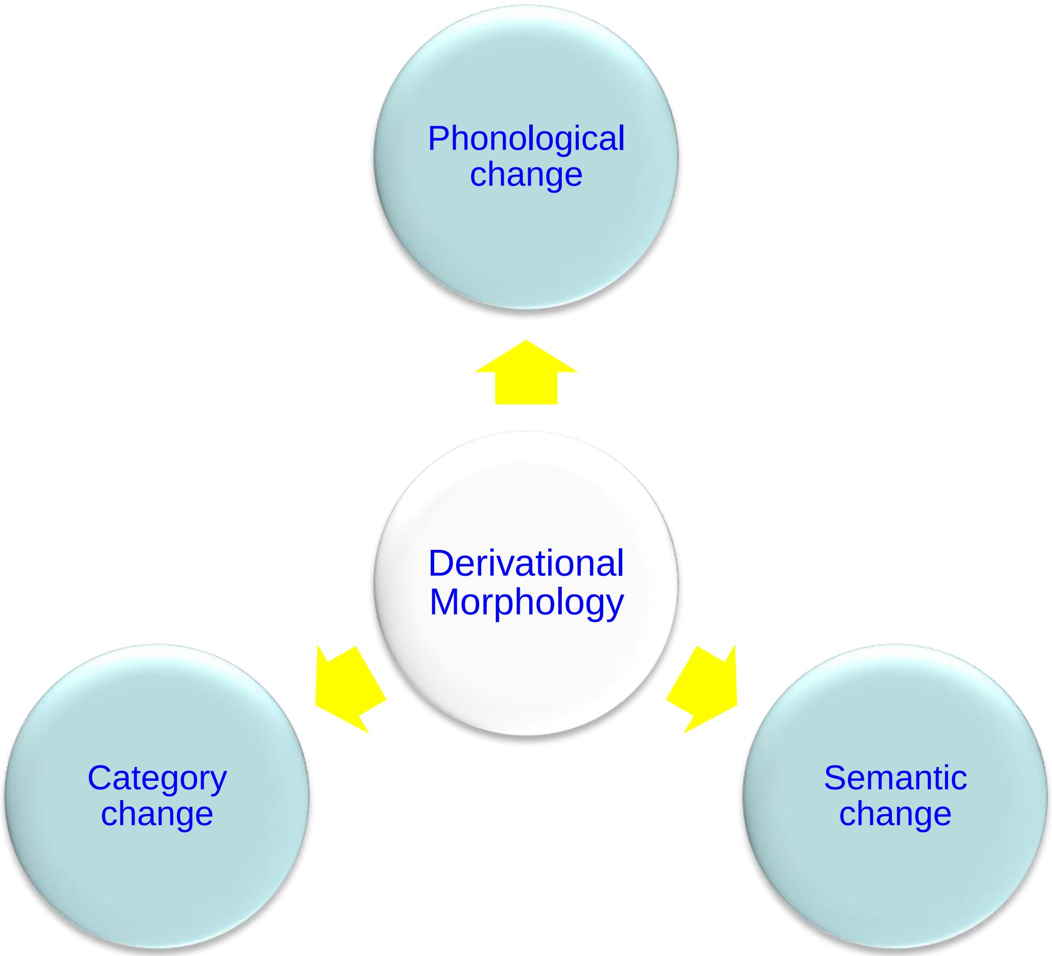
Derivational Morphology

- Derivation: Combination of a stem with a morpheme

Noun+Noun	Adjective+Noun	Preposition+Noun	Verb+Noun
hair dresser	black pepper	underground	pick pocket
water bottle	dry dinner	overdose	get goer
delivery boy	dead end	underarm	hit wicket

Adjective+Adjective	Noun+Adjective	Preposition+Verb
red hot	bottle green	underestimate
icy-cold	lion-hearted	uplift
bittersweet	earthbound	overstuff

Word Formation Rule



The –able suffix

X	Able to be X'd
read	readable
eat	eatable
break	breakable
perish	perishable

Word formation rule

Phonological change	• Pronunciation of the base is augmented by the phonetic sequence corresponding to 'able'
Category change	• -able is attached to transitive verbs and converts them into adjectives
Semantic change	• If X is the meaning of the verb, then formed word has the meaning "able to be X'd"

Backformation

- Creating a new word by removing actual or supposed affixes

Existed earlier	Formed later by backformation
resurrection	to resurrect
preemption	to preempt
television	to televise
donation	to donate

Inflectional Suffixes

- Do not cause change in the category of the base morpheme
- Indicate certain grammatical functions of the words
  - Plurality
  - Tense
- Do not cause any unpredictable changes in the meaning of the base word

Inflectional Morphology

Noun inflectional suffixes	•Plural marker -s •Possessive marker ‘s
Verb inflectional suffixes	•Third person present singular marker -s •Past tense marker -ed •Progressive marker -ing •Past participle markers -en or –ed
Adjective inflectional suffixes	•Comparative marker -er •Superlative marker -est

Problems in Morphological Analysis

- Productivity
- False Analysis
- Bound Base Morphemes

Complicate the isolation of the base of a complex word

Productivity

- Property of a morphological process to give rise to new formations on a systematic basis

Transitive Verb (read) + -able = Productive (readable)

Noun (game) + -able = Not Productive (gameable)

- Exceptions

Peaceable	Actionable	Companionable
Saleable	Marriageable	Reasonable
Impressionable	Fashionable	knowledgeable

False analysis

hospitable, sizeable

Do not have the meaning “to be able”

They can not take the suffix -ity to form a noun

Analyzing them as the words containing suffix *-able* leads to false analysis

Bound Base Morphemes

- Occur only in a particular complex word
- Do not have independent existence

base (nonexistent) + morpheme (known) = Compound

malleable

feasible (feas+ible)

- -able has the regular meaning “be able”
- -ity form is possible
- Base words do not exit independently

Classic Work (MDL Principle, Morfessor)

- John Goldsmith, *Unsupervised learning of the morphology of a natural language*, Computational Linguistics, Volume 27, Issue 2, 2001

- Mathias Creutz and Krista Lagus. *Unsupervised discovery of morphemes*, In Proceedings of the Workshop on Morphological and Phonological Learning of ACL-02, pages 21-30, Philadelphia, Pennsylvania, 11 July, 2002.

Classic Work (Porter Stemmer)

- M.F. Porter, *An algorithm for suffix stripping*, Program, **14**(3) pp 130−137, 1980.
- Uses rules like:
  - (m > 1) *EMENT -> null*
  - Here S1 is ‘EMENT’ and S2 is null. This would map REPLACEMENT to REPLAC, since REPLAC is a word part for which m = 2.

Recent Developments

FastText (embedding that respects multilinguality and morphology)

294 languages	
Developer(s)	Facebook's AI Research (FAIR) lab[1]
Initial release	November 9, 2015; 4 years ago
Stable release	0.2.0[2] / December 19, 2018; 20 months ago
Repository	github.com/facebookresearch/fastText
Written in	C++, Python
Platform	Linux, macOS, Windows
Type	Machine learning library
License	BSD License
Website	fasttext.cc

https://research.fb.com/downloads/fasttext/

Pre-trained Embeddings for Indian Languages (respects morphology)

- Kumar Saurav, Kumar Saunack, Diptesh Kanojia, and Pushpak Bhattacharyya, *"A Passage to India": Pre-trained Word Embeddings for Indian Languages,* Proceedings of the 1st Joint SLTU and CCURL Workshop (SLTU-CCURL 2020)

- Major languages from Indo-Aryan and Dravidian Family

Joint Model for Embeddings and Morphology

- Kris Cao, Marek Rei, *A Joint Model for Word Embedding and Word Morphology,* Proceedings of the 1st Workshop on Representation Learning for NLP, Berlin, 2016
- splits individual words into segments, and weights each segment according to its ability to predict context words
- Deals with unseen words which correlate better with human judgments.

Byte Pair Encoding (BPE)

- Sennrich R., Haddow B. and Birch A., *Neural machine translation of rare words with subword units*, arXiv preprint arXiv:1508.07909, 2015.
- Devlin J., Chang M. W., Lee K., and Toutanova K, *Bert: Pre-training of deep bidirectional transformers for language understanding*, arXiv preprint arXiv:1810.04805, 2018.

BPE example

Byte Pair Encoding is a compression technique (Gage, 1994)

Number of BPE merge operations=3
Vocab: A B C D E F

P_1=AD P_2=EE P_3=P_1D

Words to encode	Iterations							
BADD	1	BADD	2	BP$_1$D	3	BP$_1$D	4	BP$_3$
FAD		FAD		FP$_1$		FP$_1$		FP$_1$
FEEDE		FEEDE		FEEDE		FP$_2$DE		FP$_2$DE
ADDEEF		ADDEEF		P$_1$DEEF		P$_1$DP$_2$F		P$_3$P$_2$F

Data-dependent segmentation

- Inspired from compression theory
- MDL Principle *(Rissansen, 1978)* ⇒ Select segmentation which maximizes data likelihood

BPE construction

(1) Iteratively count character pairs in all tokens of the vocabulary.

(2) Merge every occurrence of the most frequent pair, add the new character n-gram to the vocabulary.

(3) Repeat 2, until the desired number of merge operations are completed or the desired vocabulary size is achieved (which is a hyperparameter).

BPE Application

- Quickly, slowly, abruptly, decidedly, justly, justifiably, arguably, humanly
- QuickP1, slowP1, abruptP1, decidedP1, justP1, justifiabP1, arguabP1, humanP1
- When we see a new word with P1, tag this as adverb (high probability)
- Pitfall *(not adverbs):* Lily, homely, homily, ugly

Subwords (for “jaauMgaa”, जाऊंगा)

- Characters: “j+aa+u+M+g+aa”
- Morphemes: “jaa”+”uMgaa”
- Syllables: “jaa”+”uM”+”gaa”
- Orthographic syllables: “jaau”+”Mgaa”
- BPE (depends on corpora, statistically frequent patterns): both “jaa” and “uMgaa” are likely

Chunking

Erik F. Tjong Kim Sang and Sabine Buchholz, Introduction to the CoNLL-2000 Shared Task: Chunking. In: *Proceedings of CoNLL-2000*, Lisbon, Portugal, 2000.

Data Example

[NP He] [VP reckons] [NP the current account deficit] [VP will narrow] [PP to] [NP only # 1.8 billion] [PP in] [NP September] .

He	PRP	B-NP
reckons	VBZ	B-VP
the	DT	B-NP
current	JJ	I-NP
account	NN	I-NP
deficit	NN	I-NP
will	MD	B-VP
narrow	VB	I-VP

to	TO	B-PP
only	RB	B-NP
#	#	I-NP
1.8	CD	I-NP
billion	CD	I-NP
in	IN	B-PP
September	NNP	B-NP
.	.	

https://www.aclweb.org/anthology/W00-0726.pdf

Indian Language Examples: Marathi

माणसाने	उडण्याचा	प्रयत्न	केला
NN	**VG**	**NN**	**VBD**
B	**B**	**B**	**I**

Man tried flying

त्याने	चालायला	सुरुवात	केली
PRP	**VINF**	**NN**	**VBD**
B	**B**	**B**	**I**

He started to walk

Harshada Gune, Mugdha Bapat, Mitesh Khapra and Pushpak Bhattacharyya, Verbs are where all the Action Lies: Experiences of Shallow Parsing of a Morphologically Rich Language, Computational Linguistics Conference (COLING 2010), Beijing, China, August 2010

NLP Layer

What a gripping movie was Three_Idiots!

What/WP a/DT gripping/JJ movie/NN was/VBD Three_Idiots/NNP !/!

Parse

```
(ROOT
  (FRAG
    (SBAR
      (WHNP
        (WP What))
        (S
          (NP
            (DT a)
            (JJ gripping)
            (NN movie)
          )
          (VP
            (VBD was)
            (NP
            (NNP Three_idiots)))))
          (. !)
  )
)
```

Universal dependencies

dobj(Three_Idiots-6, What-1)
det(movie-4, a-2)
amod(movie-4, gripping-3)
nsubj(Dangal-6, movie-4)
cop(Dangal-6, was-5)
root(ROOT-0, Three_idiots-6)

Algorithmics and Mathematics of Chunking

Noisy Channel Model

W *Noisy Channel* **T**

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

Sequence *W* is transformed into sequence *T*

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

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

Sequence to Sequence Labelling: Chunk w/o chunk type

माणसाने	उडण्याचा	प्रयत्न	केला
NN	**VG**	**NN**	**VBD**
B	**B**	**B**	**I**

Chunking vs. POS Tagging

- Much simpler task than POS tagging!
- Only 2 tags in the simplest form: '***B***' and '***I***'
- Makes use of POS and MORPH information
- Slightly more complex when the "TYPE" of chunk also is required

Chunk with chunk type

[NP He] [VP reckons] [NP the current account deficit] [VP will narrow] [PP to] [NP only # 1.8 billion] [PP in] [NP September] .

He	PRP	B-NP
reckons	VBZ	B-VP
the	DT	B-NP
current	JJ	I-NP
account	NN	I-NP
deficit	NN	I-NP
will	MD	B-VP
narrow	VB	I-VP

to	TO	B-PP
only	RB	B-NP
#	#	I-NP
1.8	CD	I-NP
billion	CD	I-NP
in	IN	B-PP
September	NNP	B-NP
.	.	

Decoding for the best chunk

$$\hat{\boldsymbol{y}} = \arg\max_{\boldsymbol{y}} p_{\boldsymbol{\lambda}}(\boldsymbol{y}|\boldsymbol{x}) = \arg\max_{\boldsymbol{y}} \boldsymbol{\lambda} \cdot \boldsymbol{F}(\boldsymbol{y}, \boldsymbol{x})$$

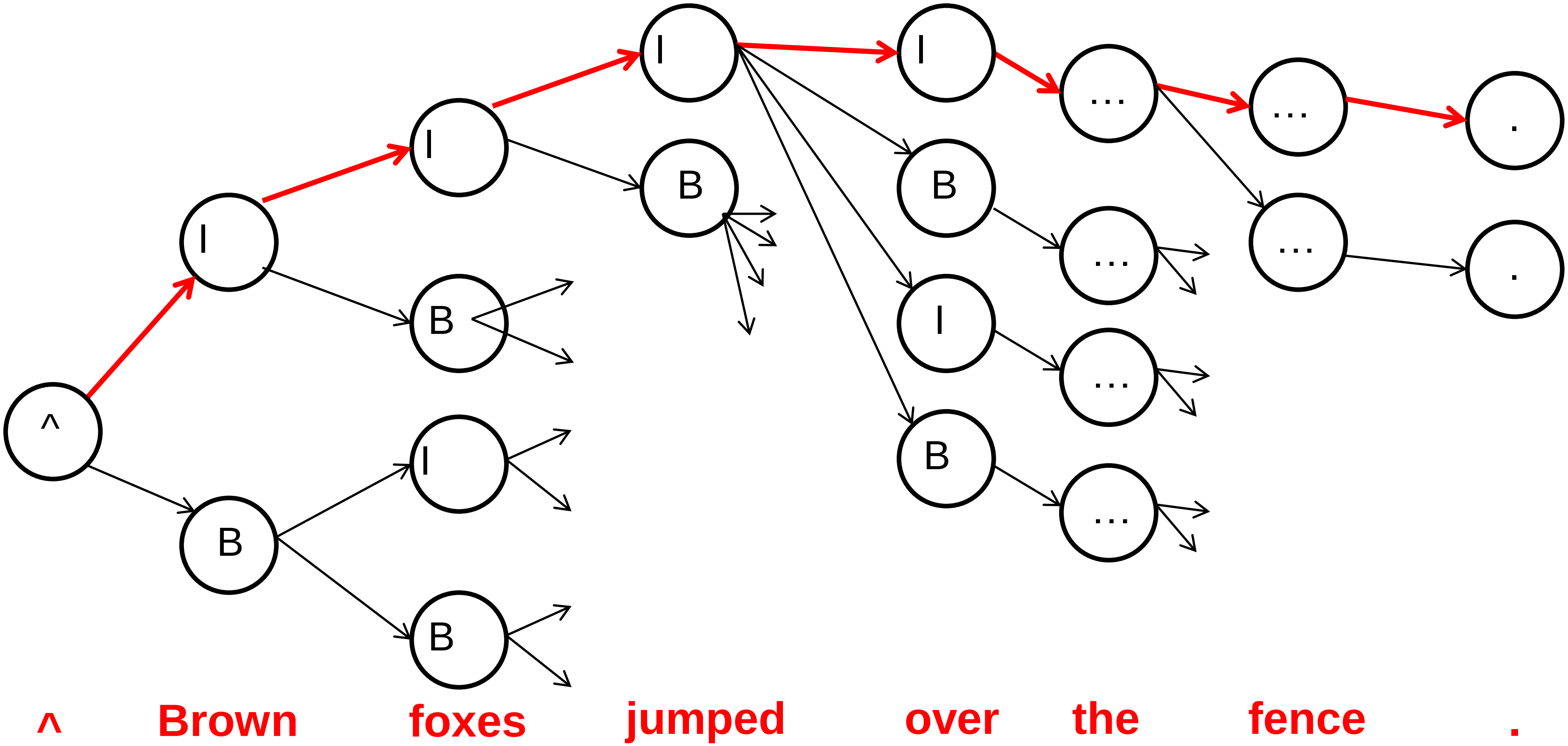
$$p_{\boldsymbol{\lambda}}(\boldsymbol{Y}|\boldsymbol{X}) = \frac{\exp \boldsymbol{\lambda} \cdot \boldsymbol{F}(\boldsymbol{Y}, \boldsymbol{X})}{Z_{\boldsymbol{\lambda}}(\boldsymbol{X})} \qquad (1)$$

where

$$Z_{\boldsymbol{\lambda}}(\boldsymbol{x}) = \sum_{\boldsymbol{y}} \exp \boldsymbol{\lambda} \cdot \boldsymbol{F}(\boldsymbol{y}, \boldsymbol{x})$$

$$\boldsymbol{F}(\boldsymbol{y}, \boldsymbol{x}) = \sum_{i} \boldsymbol{f}(\boldsymbol{y}, \boldsymbol{x}, i)$$

i ranges over the input positions

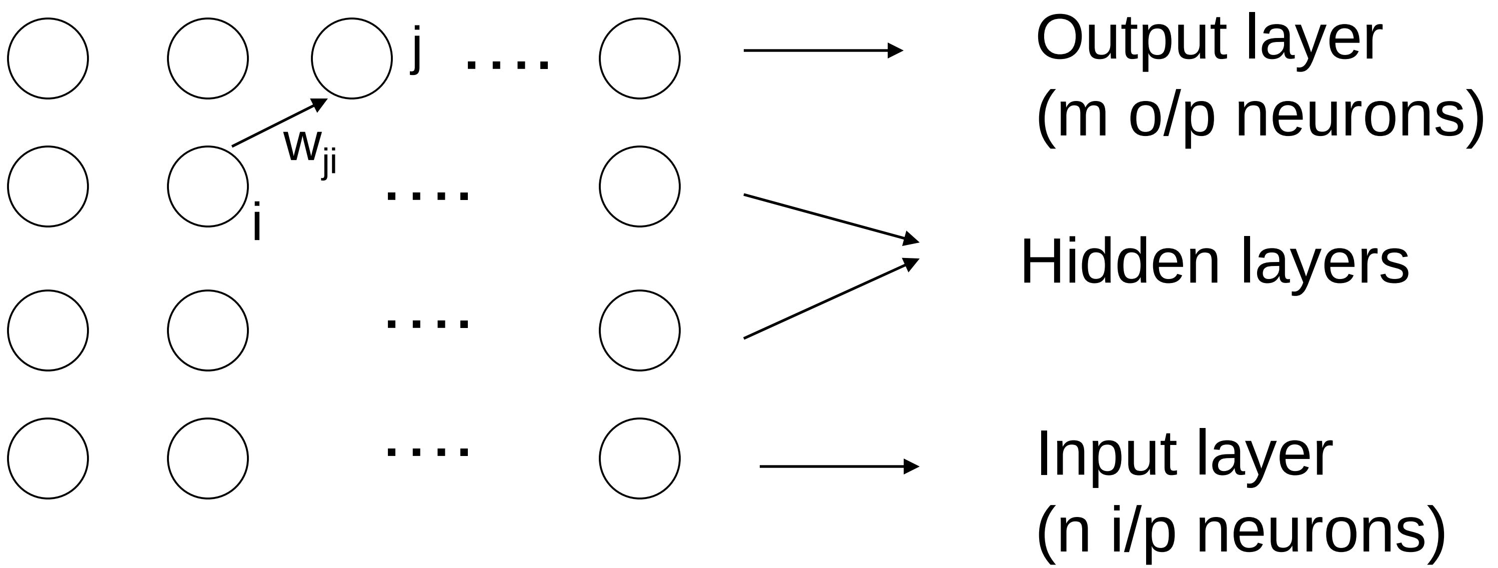


Probability of a path (e.g. Top most path) = *Product of $P(Y_i|Y_{i-1}, X)$*

Gradient Descent

Explaining through Feed Forward Neural Network and Backpropagation

Backpropagation algorithm



- Fully connected feed forward network
- Pure FF network (no jumping of connections over layers)

Gradient Descent Equations

$$\Delta w_{ji} = -\eta \frac{\delta E}{\delta w_{ji}} \; (\eta = \text{learning rate}, 0 \le \eta \le 1)$$

$$\frac{\delta E}{\delta w_{ji}} = \frac{\delta E}{\delta net_j} \times \frac{\delta net_j}{\delta w_{ji}} \; (net_j = \text{input at the j}^{th} \text{ layer})$$

$$\frac{\delta E}{\delta net_j} = -\delta j$$

$$\Delta w_{ji} = \eta \delta j \frac{\delta net_j}{\delta w_{ji}} = \eta \delta j o_i$$

Backpropagation – for outermost layer

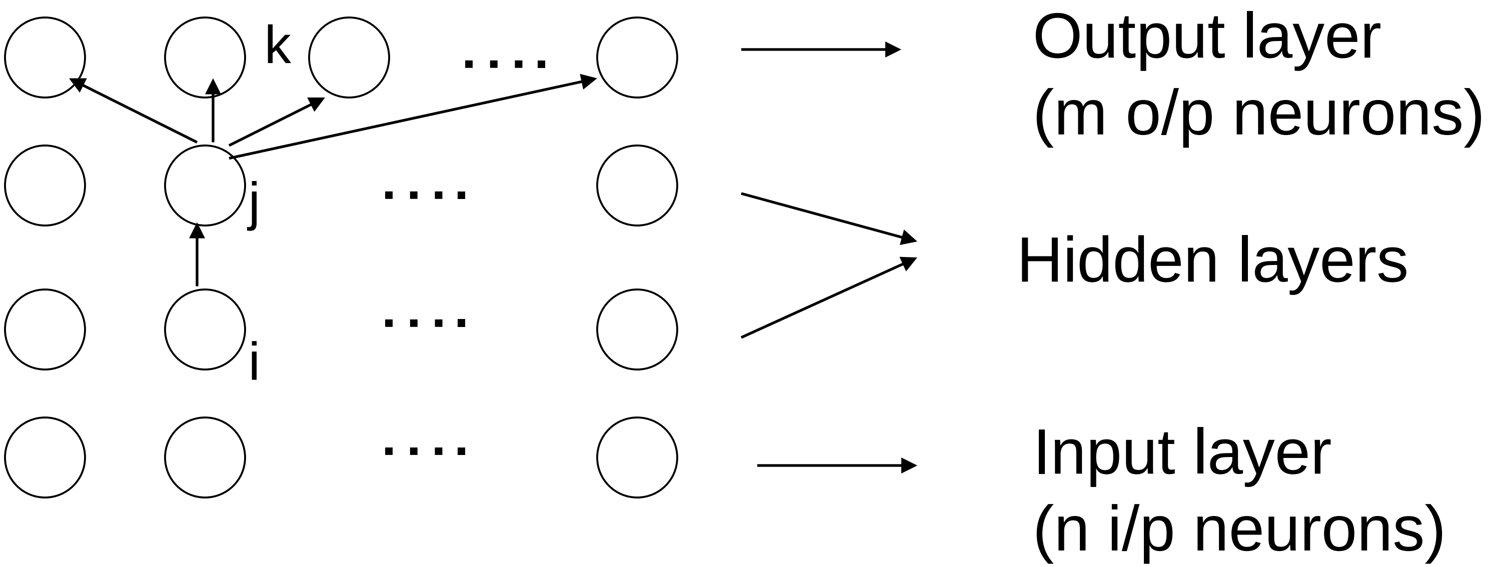
$$\delta j = -\frac{\delta E}{\delta net_j} = -\frac{\delta E}{\delta o_j} \times \frac{\delta o_j}{\delta net_j} (net_j = \text{input at the } \mathrm{j}^{th} \text{ layer})$$

$$E = \frac{1}{2}\sum_{p=1}^{m}(t_p - o_p)^2$$

$$\text{Hence, } \delta j = -(-(t_j - o_j)o_j(1 - o_j))$$

$$\Delta w_{ji} = \eta(t_j - o_j)o_j(1 - o_j)o_i$$

Backpropagation for hidden layers



δ_k is propagated backwards to find value of δ_j

Backpropagation – for hidden layers

$$\Delta w_{ji} = \eta \delta j o_i$$

$$\delta j = -\frac{\delta E}{\delta net_j} = -\frac{\delta E}{\delta o_j} \times \frac{\delta o_j}{\delta net_j}$$

$$= -\frac{\delta E}{\delta o_j} \times o_j (1 - o_j)$$

$$= -\sum_{k \in \text{next layer}} \left(\frac{\delta E}{\delta net_k} \times \frac{\delta net_k}{\delta o_j}\right) \times o_j (1 - o_j)$$

$$\text{Hence, } \delta_j = -\sum_{k \in \text{next layer}} (-\delta_k \times w_{kj}) \times o_j (1 - o_j)$$

$$= \sum_{k \in \text{next layer}} (w_{kj} \delta_k) o_j (1 - o_j) o_i$$

General Backpropagation Rule

- General weight updating rule:

$$\Delta w_{ji} = \eta \delta j o_i$$

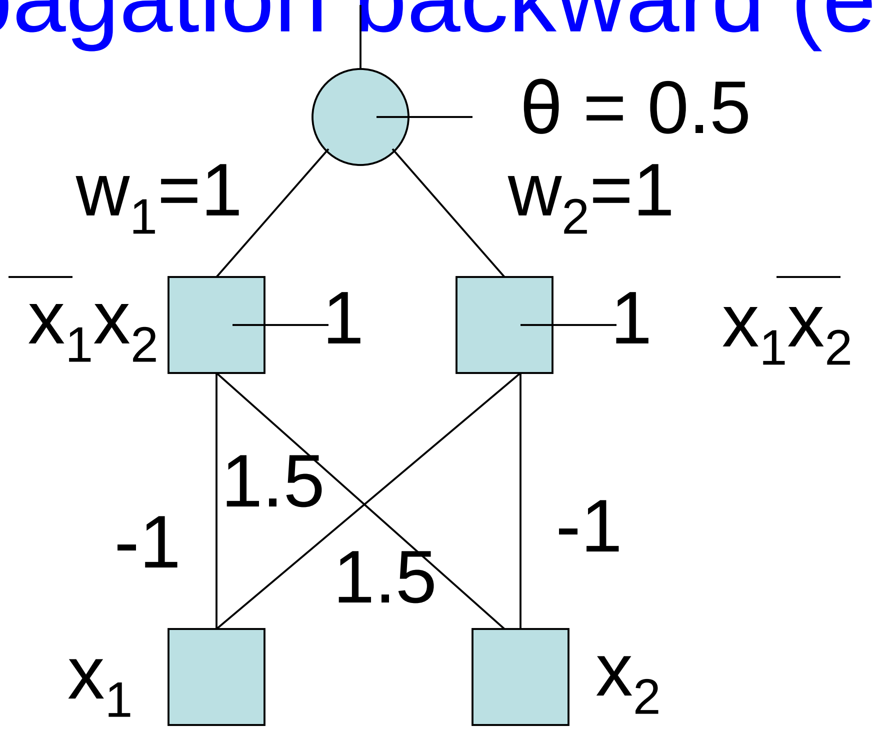
- Where

$$\delta_j = (t_j - o_j) o_j (1 - o_j) \quad \text{for outermost layer}$$

$$= \sum_{k \in \text{next layer}} (w_{kj} \delta_k) o_j (1 - o_j) o_i \text{ for hidden layers}$$

How does it work?

- Input propagation forward and error propagation backward (e.g. XOR)



Next Assignment

Chunking

- Input- Sentences
- Output- Chunk labels on sentences (only *B* and *I*), e.g.,
  - I/P- *Many birds were flying*
  - O/P- *B I B I*
- Goal- does POS tagging indeed help
- Do chunking with POS and without POS
- Compare accuracy (P, R, F)

Evaluation of POS Tagging

Typical POS tag steps

- Implementation of Viterbi – Unigram, Bigram.
- Five Fold Evaluation.
- Per POS Accuracy.
- Confusion Matrix.

Screen shot of typical Confusion Matrix

	AJ0	AJ0-AV0	AJ0-NN1	AJ0-VVD	AJ0-VVG	AJ0-VVN	AJC	AJS	AT0	AV0	AV0-AJ0	AVP
AJ0	2899	20	32	1	3	3	0	0	18	35	27	1
AJ0-AV0	31	18	2	0	0	0	0	0	0	1	15	0
AJ0-NN1	161	0	116	0	0	0	0	0	0	0	1	0
AJ0-VVD	7	0	0	0	0	0	0	0	0	0	0	0
AJ0-VVG	8	0	0	0	2	0	0	0	1	0	0	0
AJ0-VVN	8	0	0	3	0	2	0	0	1	0	0	0
AJC	2	0	0	0	0	0	69	0	0	11	0	0
AJS	6	0	0	0	0	0	0	38	0	2	0	0
AT0	192	0	0	0	0	0	0	0	7000	13	0	0
AV0	120	8	2	0	0	0	15	2	24	2444	29	11
AV0-AJ0	10	7	0	0	0	0	0	0	0	16	33	0
AVP	24	0	0	0	0	0	0	0	1	11	0	737

Computing P(.) values

Let us suppose annotated corpus has the following sentence

I	have	a	brown	bag	.
PRN	**VB**	**DT**	**JJ**	**NN**	**.**

$$P(NN \mid JJ) = \frac{Number\_of\_times\_JJ\_followed\_by\_NN}{Number\_of\_times\_JJ\_appeared}$$

$$P(Brown \mid JJ) = \frac{Number\_of\_times\_Brown\_tagged\_as\_JJ}{Number\_of\_times\_JJ\_appeared}$$

Why Ratios?

- This way of computing parameter probabilities: is this <u>correct</u>?
- What does “correct” mean?
- Is this principled?
- We are using Maximum Likelihood Estimate (<u>**MLE**</u>)
- Assumption: underlying distribution is multinomial

Explanation with coin tossing

- A coin is tossed 100 times, Head appears 40 times
- P(H)= 0.4
- Why?
- Because of maximum likelihood

N tosses, K Heads, parameter *P(H)=p*

- Construct Maximum Likelihood Expression
- Take log likelihood and take derivative
- Equate to 0 and Get *p*

$$L = p^K (1-p)^{N-K}$$

$$\Rightarrow LL = \log(L) = K \log p + (N-K) \log(1-p)$$

$$\Rightarrow \frac{d(LL)}{dp} = \frac{K}{p} - \frac{N-K}{1-p}$$

$$\Rightarrow \frac{d(LL)}{dp} = 0 \quad gives \quad p = \frac{K}{N}$$

Exercise

- Following the process for finding the probability of Head from N tosses of coin yielding K Heads, prove that the transition probabilities can be found from MLE
- Most important: get the likelihood expression
- Use chapter 2 of the book
  - Pushpak Bhattacharyya: Machine translation, CRC Press, Taylor & Francis Group, Boca Raton, USA, 2015, ISBN: 978-1-4398-9718-8