CS772: Deep Learning for Natural Language Processing (DL-NLP)

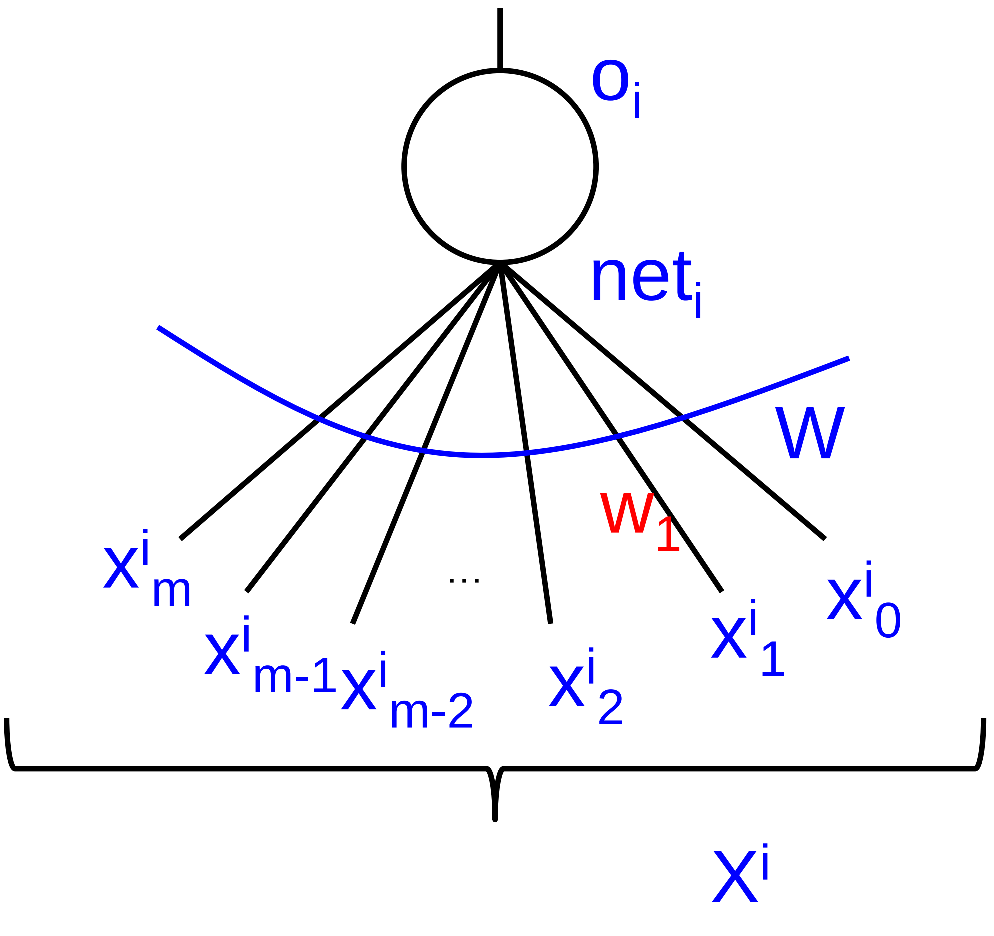
Word Vectors

Pushpak Bhattacharyya
Computer Science and Engineering Department
IIT Bombay

Week 4 of 23rd Jan, 2023

Re-cap

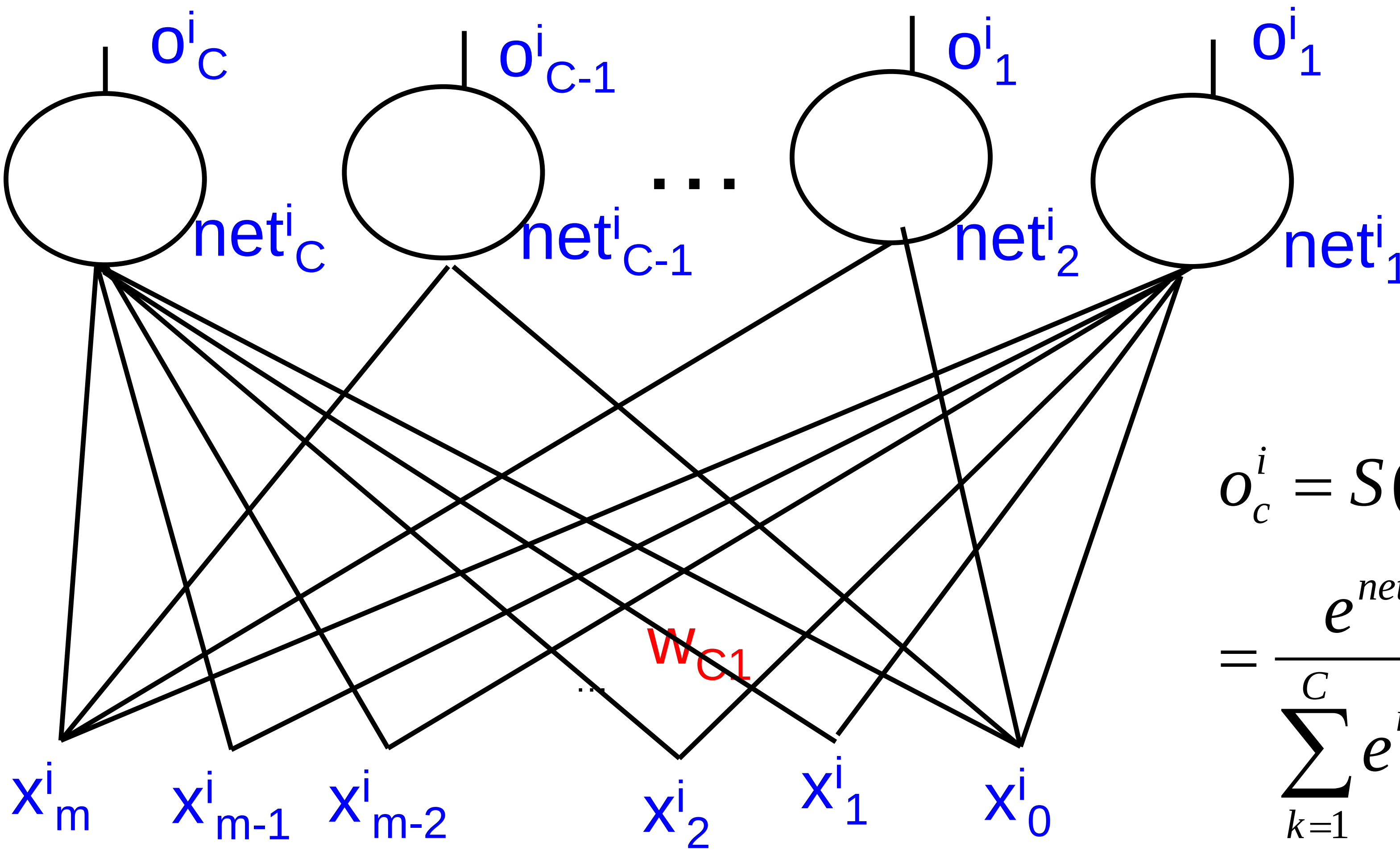
Sigmoid neuron



$$o^i = \frac{1}{1 + e^{-net^i}}$$

$$net_i = W.X^i = \sum_{j=0}^{m} w_j x_j^i$$

Softmax Neuron

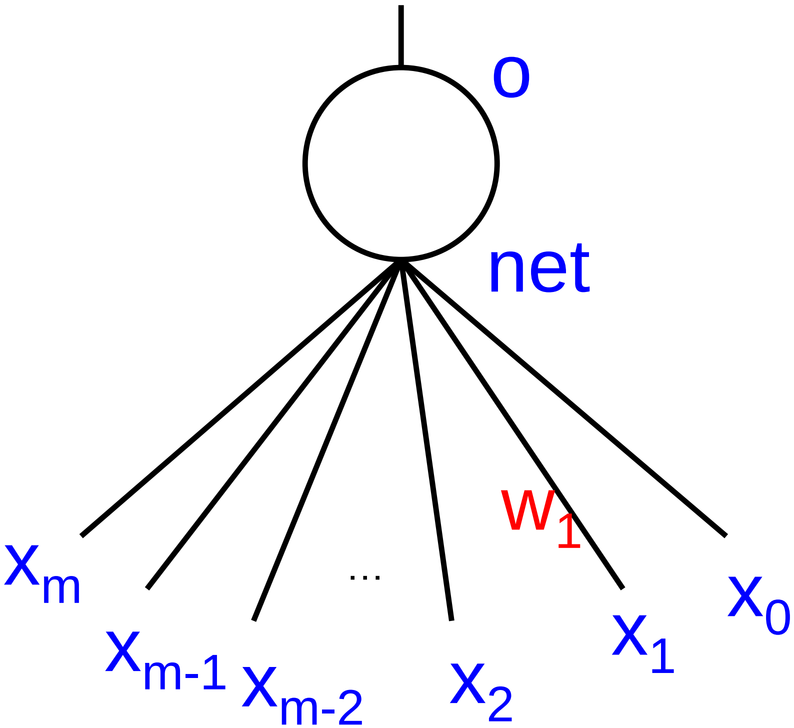


$$o_c^i = S(NET^i)_c$$

$$= \frac{e^{net_c^i}}{\sum_{k=1}^{C} e^{net_k^i}}$$

Output for class c (small c), c:1 to C

Single sigmoid neuron- weight change rule



$$\frac{\partial E}{\partial w_1} = \frac{\partial E}{\partial o} \cdot \frac{\partial o}{\partial net} \cdot \frac{\partial net}{\partial w_1}$$

$$E = -t \log o - (1-t) \log(1-o)$$

$$\Rightarrow \frac{\partial E}{\partial o} = -\frac{t}{o} + \frac{1-t}{1-o} = -\frac{t-o}{o(1-o)}$$

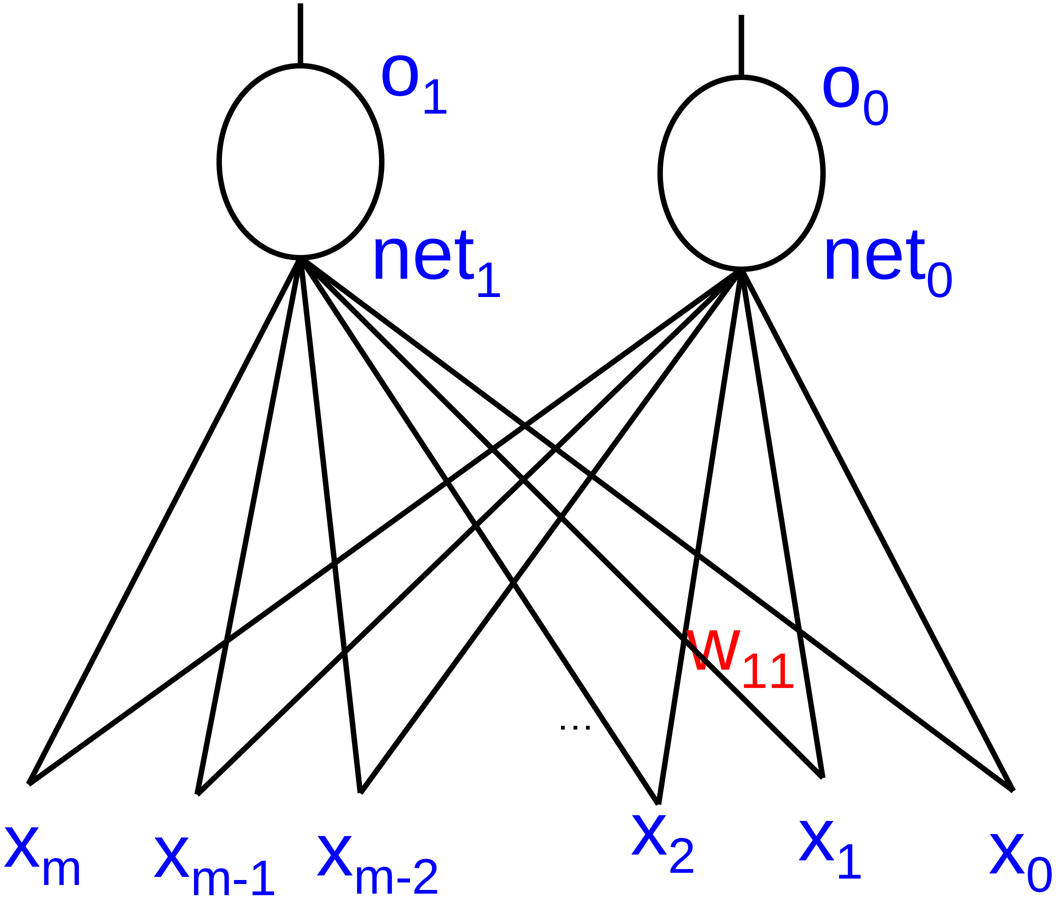
$$o = \frac{1}{1+e^{-net}} \; (sigmoid) \Rightarrow \frac{\partial o}{\partial net} = o(1-o)$$

$$net = \sum_{j=0}^{m} w_j x_j \;\; \Rightarrow \frac{\partial net}{\partial w_1} = x_1$$

$$\Rightarrow \Delta w_1 = \eta \frac{\partial E}{\partial w_1} = \eta(t-o)x_1$$

$\Delta w_1 = \eta(t\text{-}o)x_1$

Multiple neurons in the output layer: softmax+*cross entropy* loss (1/2): illustrated with 2 neurons and single training data point



$$O = < o_1, o_0 >$$

$$NET = < net_1, net_0 >$$

$$o_1 = \frac{e^{net_1}}{e^{net_1} + e^{net_0}},\ o_0 = \frac{e^{net_0}}{e^{net_1} + e^{net_0}}$$

$$\frac{\partial O}{\partial NET} = \begin{bmatrix} \frac{\partial o_0}{\partial net_0} & \frac{\partial o_1}{\partial net_0} \\ \frac{\partial o_0}{\partial net_1} & \frac{\partial o_1}{\partial net_1} \end{bmatrix}$$

$$= \begin{bmatrix} o_0(1 - o_0) & -o_0 o_1 \\ -o_1 o_0 & o_1(1 - o_1) \end{bmatrix}$$

Softmax and Cross Entropy (2/2)

$$E = -t_1 \log o_1 - t_0 \log o_0$$

$$o_1 = \frac{e^{net_1}}{e^{net_1} + e^{net_0}},\ o_0 = \frac{e^{net_0}}{e^{net_1} + e^{net_0}}$$

$$\frac{\partial E}{\partial w_{11}} = -\frac{t_1}{o_1}\frac{\partial o_1}{\partial w_{11}} - -\frac{t_0}{o_0}\frac{\partial o_0}{\partial w_{11}}$$

$$\frac{\partial o_1}{\partial w_{11}} = \frac{\partial o_1}{\partial net_1}.\frac{\partial net_1}{\partial w_{11}} + \frac{\partial o_1}{\partial net_0}.\frac{\partial net_0}{\partial w_{11}} = o_1(1-o_1)x_1 + 0$$

$$\frac{\partial o_0}{\partial w_{11}} = \frac{\partial o_0}{\partial net_1}.\frac{\partial net_1}{\partial w_{11}} + \frac{\partial o_0}{\partial net_0}.\frac{\partial net_0}{\partial w_{11}} = -o_1 o_0 x_1 + 0$$

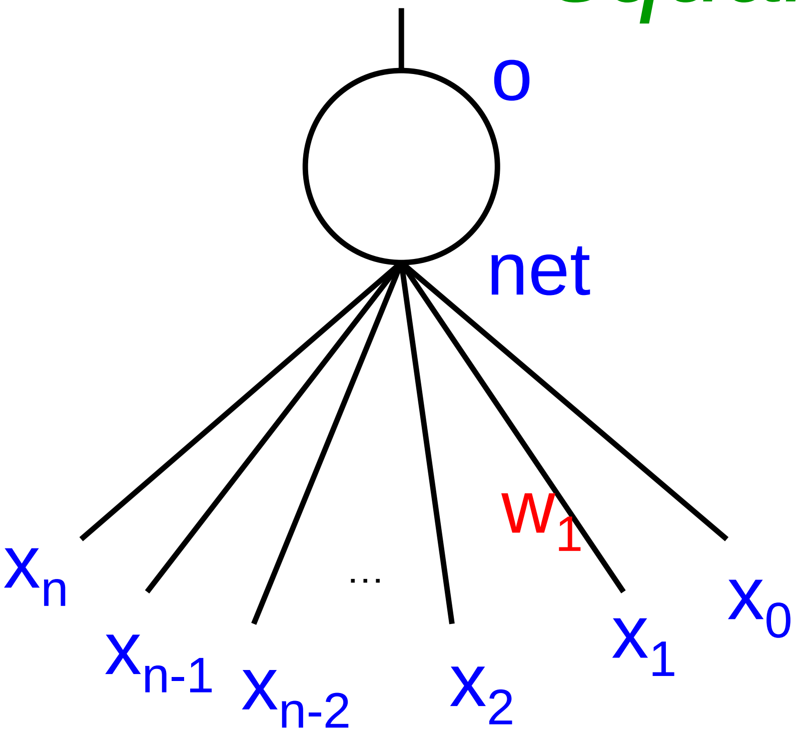
$$\Rightarrow \frac{\partial E}{\partial w_{11}} = -t_1(1-o_1)x_1 + t_0 o_1 x_1 = -t_1(1-o_1)x_1 + (1-t_1)o_1 x_1$$

$$= [-t_1 + t_1 o_1 + o_1 - t_1 o_1]x_1 = -(t_1 - o_1)x_1$$

$$\Delta w_{11} = -\eta \frac{\partial E}{\partial w_{11}} = \eta(t_1 - o_1)x_1$$

Weight change rule with TSS

Single neuron: *sigmoid+total sum square* (tss) loss



Lets consider wlg w_1. Change is weight $\Delta w_1 = -\eta \delta L / \delta w_1$

η= learning rate,

L=loss= $\frac{1}{2}(t-o)^2$,

t=target, o=observed output

$$\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial o} \cdot \frac{\partial o}{\partial net} \cdot \frac{\partial net}{\partial w_1}$$

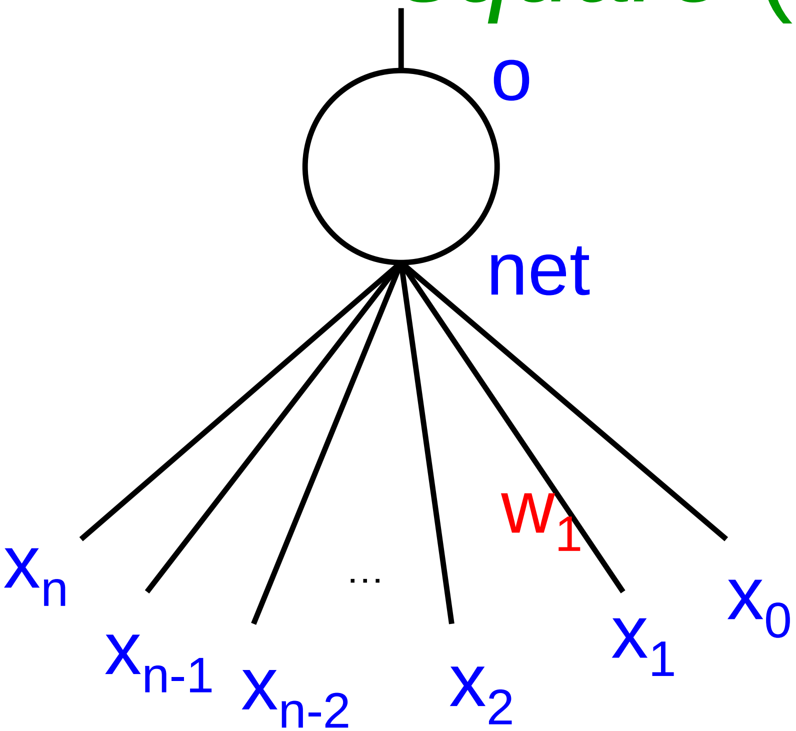
$$L = \frac{1}{2}(t-o)^2 \Rightarrow \frac{\partial L}{\partial o} = -(t-o) \quad (1)$$

$$o = \frac{1}{1+e^{-net}} \, (sigmoid) \Rightarrow \frac{\partial o}{\partial net} = o(1-o) \; (2)$$

$$net = \sum_{i=0}^{n} w_i x_i \Rightarrow \frac{\partial net}{\partial w_1} = x_1 \quad (3)$$

$$\Rightarrow \Delta w_1 = \eta (t-o) o (1-o) x_1$$

Single neuron: *sigmoid+total sum square* (tss) loss (cntd)



$$\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial o} \cdot \frac{\partial o}{\partial net} \cdot \frac{\partial net}{\partial w_1}$$

$$L = \frac{1}{2}(t-o)^2 \;\Rightarrow\; \frac{\partial L}{\partial o} = (t-o) \;\; (1)$$

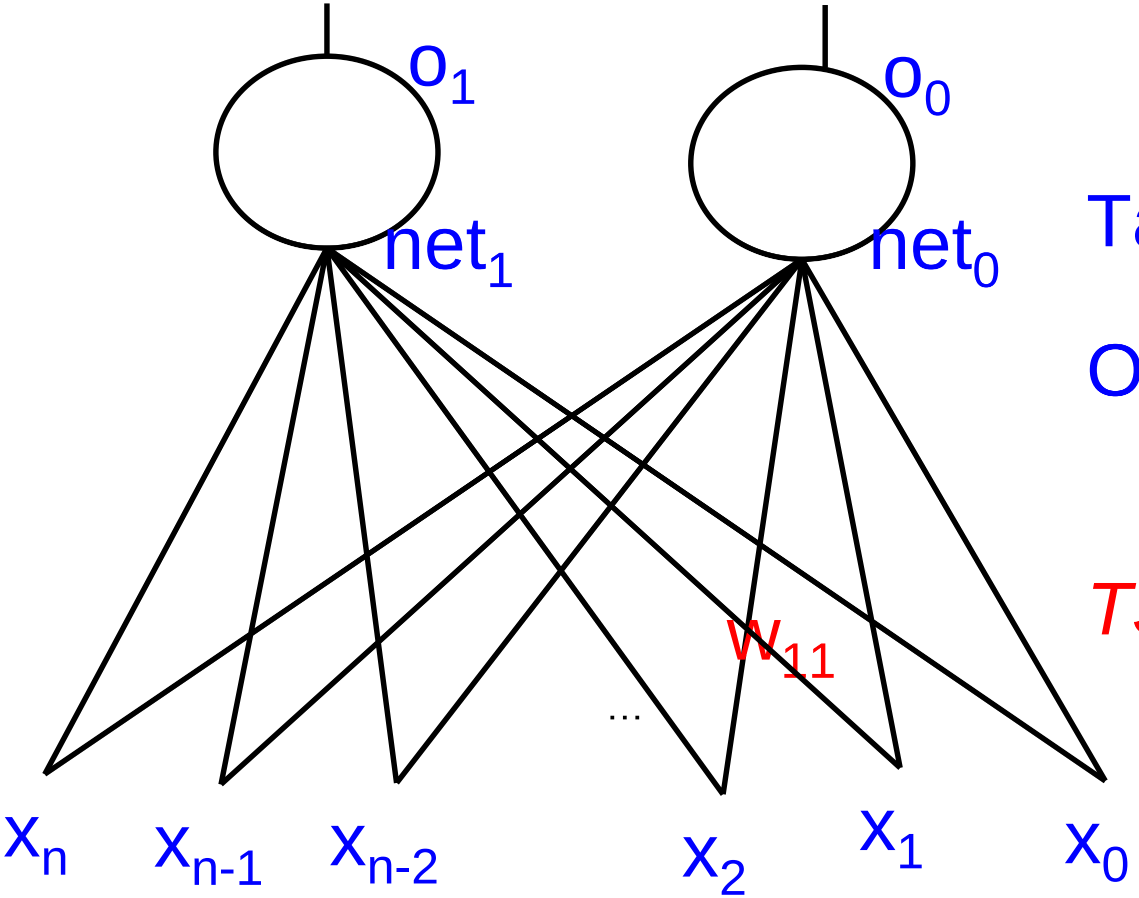
$$o = \frac{1}{1+e^{-net}} \;(sigmoid) \Rightarrow \frac{\partial o}{\partial net} = o(1-o) \; (2)$$

$$net = \sum_{i=0}^{n} w_i x_i \;\Rightarrow\; \frac{\partial net}{\partial w_1} = x_i \;\; (3)$$

$$\Rightarrow \Delta w_1 = \eta(t-o)o(1-o)x_i$$

$$\Delta w_1 = \eta(t\text{-}o)o(1\text{-}o)x_1$$

Multiple neurons in the output layer: *sigmoid+total sum square* (tss) loss



Target vector: $<t_1, t_0>$

Observed vector: $<o_1, o_0>$

TSS Loss$= ½[(t_1-o_1)^2+(t_0-o_0)^2]$

$$\Delta w_{11} = \eta(t_1-o_1)o_1(1-o_1)x_1$$

General Backpropagation Rule

- General weight updating rule:

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

- Where

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

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

Word Vectors

Deriving the word vector: setting

$$W^s : w_0^s, w_1^s, w_2^s, \ldots w_i^s, \ldots w_m^s$$

$$V_{w_i} : [v_0^i, v_1^i, v_2^i, \ldots v_k^i, \ldots v_d^i]$$

$$J = P(w_j \mid w_i)$$

$$L = -P(w_j \mid w_i)$$

$$P(w_j \mid w_i) = \frac{e^{V_{w_i} . V_{w_j}}}{\sum_{j'=1}^{|V|} e^{V_{w_i} . V_{w_{j'}}}}$$

$$LL = -V_{w_i} . V_{w_j} + \ln\left(\sum_{j'=1}^{|V|} e^{V_{w_i} . V_{w_{j'}}} \right)$$

W^S: word sequence in the s^{th} Sentence

V_{wi}: word vector of w_i

Deriving the word vector: Optimization (1/2)

$$V_{w_i} : [v_0^i, v_1^i, v_2^i, \ldots v_k^i, \ldots v_d^i] = [u_0, u_1, u_2, \ldots u_k, \ldots u_d]$$

$$V_{w_j} : [v_0^j, v_1^j, v_2^j, \ldots v_k^j, \ldots v_d^j] = [v_0, v_1, v_2, \ldots v_k, \ldots v_d]$$

$$V_{w_{j'}} : [v'_0, v'_1, v'_2, \ldots v'_k, \ldots v'_d]$$

$$V_{w_i} . V_{w_j} = \sum_{k=0}^{d} u_k v_k$$

$$\frac{\partial LL}{\partial u_k} = -v_k + \frac{\dfrac{\partial}{\partial u_k}\left(\displaystyle\sum_{j'=1}^{|V|} e^{\sum_{k=0}^{d} u_k v'_k}\right)}{\displaystyle\sum_{j'=1}^{|V|} e^{\sum_{k=0}^{d} u_k v'_k}}$$

Deriving the word vector: Optimization

$$= -v_k + \frac{\sum_{j'=1}^{|V|} \frac{\partial}{\partial u_k}\left(e^{\sum_{k=0}^{d} u_k v_k^{'}}\right)}{\sum_{j'=1}^{|V|} e^{\sum_{k=0}^{d} u_k v_k^{'}}} = -v_k + \frac{\sum_{j'=1}^{|V|} e^{\sum_{k=0}^{d} u_k v_k^{'}} \frac{\partial}{\partial u_k}\left(\sum_{k=0}^{d} u_k v_k^{'}\right)}{\sum_{j'=1}^{|V|} e^{\sum_{k=0}^{d} u_k v_k^{'}}}$$

$$= -v_k + \frac{\sum_{j'=1}^{|V|} e^{\sum_{k=0}^{d} u_k v_k^{'}} v_k^{'}}{\sum_{j'=1}^{|V|} e^{\sum_{k=0}^{d} u_k v_k^{'}}} = -v_k + \sum_{j'=1}^{|V|} P(w_{j'} \mid w_i).v_{k'} = -v_k + E(v_{k'})$$

Deriving the word vector, Gradient Descent: Δu_k

$$\Delta u_k = -\eta \frac{\partial LL}{\partial u_k} = \eta[v_k - E(v_{k'})]$$

Representation

How to input text to neural net? Issue of REPRESENTATION

- Inputs have to be sets of numbers
  - We will soon see why

- These numbers form **REPRESENTATIONS**

- What is a good representation? At what granularity: words, n-grams, phrases, sentences

Issues

- What is a good representation? At what granularity: words, n-grams, phrases, sentences
- Sentence is important- (a) *I <u>bank</u> with SBI; (b) I took a stroll on the river <u>bank</u>; (c) this <u>bank</u> sanctions loans quickly*
- Each ‘bank’ should have a different representation
- We have to LEARN these representations

Principle behind representation

- Proverb: “A man is known by the company he keeps”
- Similarly: “A word is known/represented by the company it keeps”
- “Company” → Distributional Similarity

Representation: to learn or not learn?

- 1-hot representation does not capture many nuances, e.g., semantic similarity
  - But is a good starting point
- Collocations also do not fully capture all the facets
  - But is a good starting point

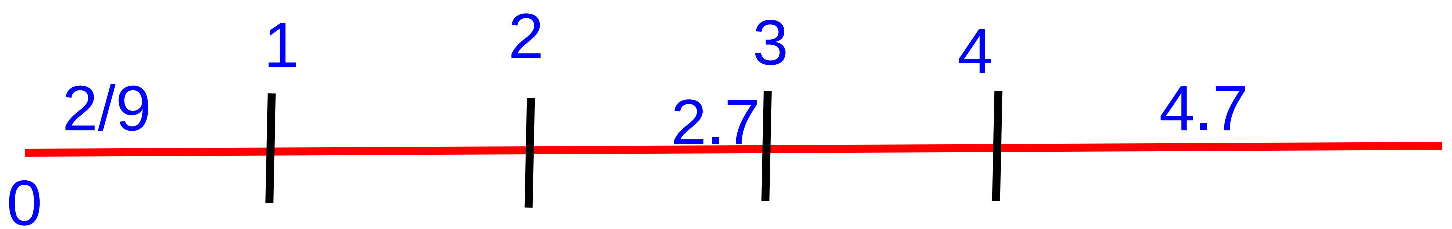
So learn the representation…

- Learning Objective

- ***MAXIMIZE CONTEXT PROBABILITY***

Foundations-1: Embedding

- Way of taking a discrete entity to a continuous space
- E.g., 1, 2, 3, 2.7, 2/9, $22^{1/2}$, … are numerical symbols
- But they are points on the real line
- Natural embedding
- Words' embedding not so intuitive!



Foundations-2: Purpose of Embedding

- Enter geometric space
- Take advantage of “distance measures”- Euclidean distance, Riemannian distance and so on
- “Distance” gives a way of computing similarity

Foundations-3: Similarity and difference

- Recognizing similarity and difference- foundation of intelligence
- Lot of Pattern Recognition is devoted to this task (Duda, Hart, Stork, 2nd Edition, 2000)
- Lot of NLP is based on Text Similarity
- Words, phrases, sentences, paras and so on (verticals)
- Lexical, Syntactic, Semantic, Pragmatic (Horizontal)

Similarity study in MT

English:

This blanket is very soft

Hindi:

yaha kambal bahut naram hai

Bangla:

ei kambal ti khub naram <null>

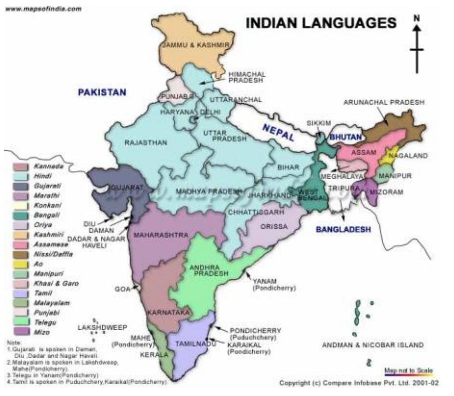
Marathi:

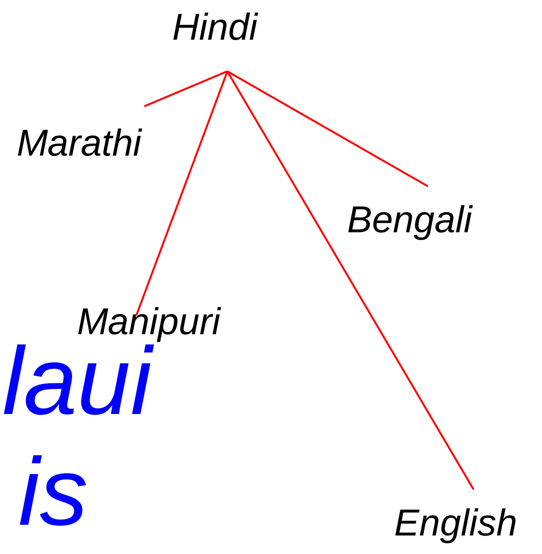
haa kambal khup naram aahe

Manipuri:

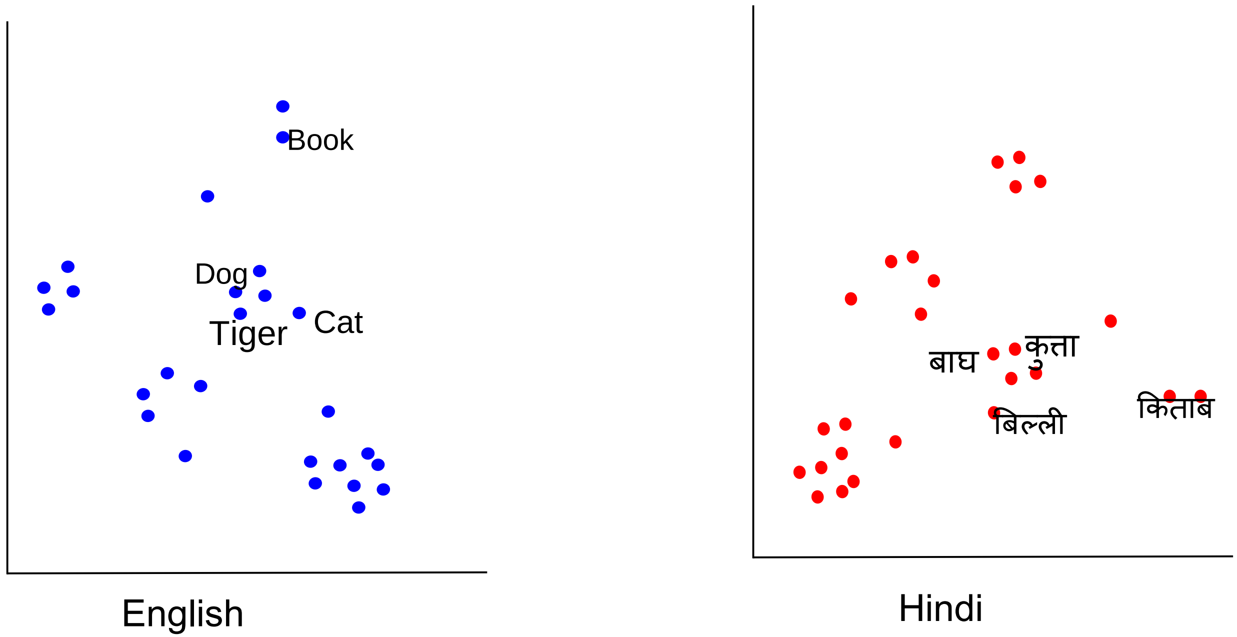
kampor asi mon mon laui

blanket this soft soft is





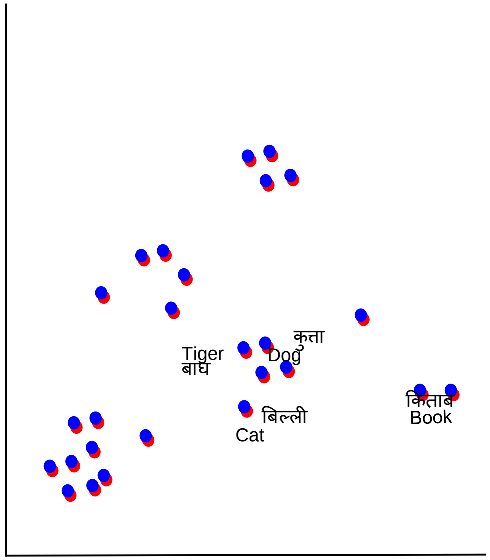
ISO-Metricity



Across Cross-lingual Mapping

This involves strong assumption that embedding spaces across languages are isomorphic, which is not true specifically for distance languages (Søgaard et al. 2018). However, without this assumption unsupervised NMT is not possible.

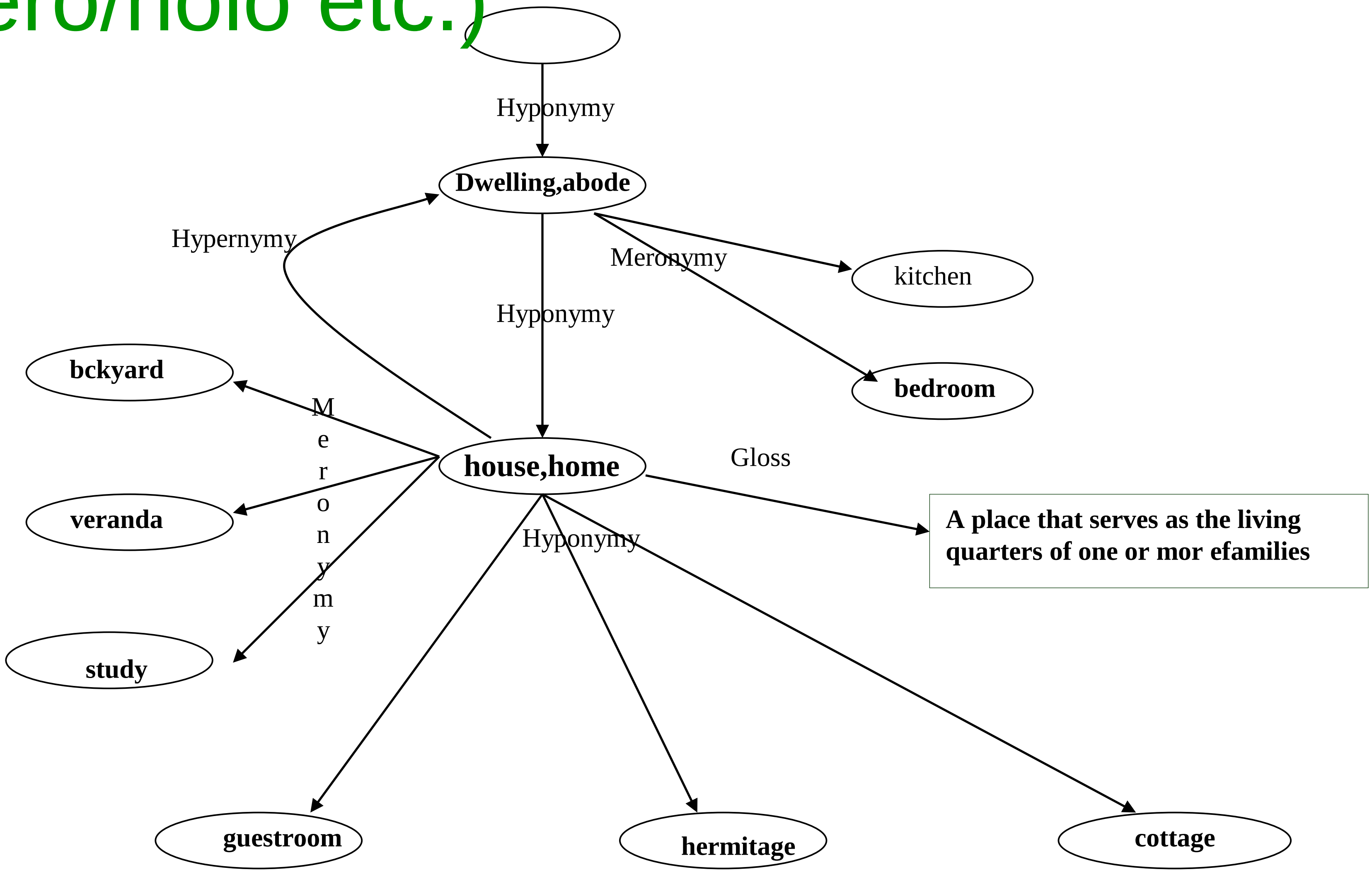
Søgaard, Anders, Sebastian Ruder, and Ivan Vulić. 2018. On the limitations of unsupervised bilingual dictionary induction. ACL



Foundations-4: Syntagmatic and Paradigmatic Relations

- Syntagmatic and paradigmatic relations
  - Lexico-semantic relations: synonymy, antonymy, hypernymy, mernymy, troponymy etc. **CAT is-a ANIMAL**
  - Coccurence: **CATS MEW**
- Wordnet: primarily paradigmatic relations
- ConceptNet: primarily Syntagmatic Relations

WordNet Sub-Graph with lexico-semantic relations (hyper/hypo, mero/holo etc.)



Lexical and Semantic relations in wordnet

1. Synonymy (e.g., *house, home*)
2. Hypernymy / Hyponymy (kind-of, e.g., *cat* ←→ *animal*)
3. Antonymy (e.g., *white and black*)
4. Meronymy / Holonymy (part of, e.g., *cat and tail*)
5. Gradation (e.g., *sleep*→*doze*→*wake up*)
6. Entailment (e.g., *snoring* → *sleeping*)
7. Troponymy (manner of, e.g., *whispering and talking*)

1, 3 and 5 are lexical (*word to word)*, rest are semantic (*synset to synset).*

‘Paradigmatic Relations’ and ‘Substitutability’

- Words in paradigmatic relations can substitute each other in the sentential context
- E.g., ‘The cat is drinking milk’ → ‘The animal is drinking milk’
- Substitutability is a foundational concept in linguistics and NLP

Foundations-5: Learning and Learning Objective

- Probability of getting the context words given the target should be maximized (skip gram)

- Probability of getting the target given context words should be maximized (CBOW)

Learning objective (skip gram)

$$J'(\theta) = \frac{1}{T} \prod_{t=1}^{T} \prod_{\substack{-m \le j \le m \\ j \ne 0}} p(w_{t+j} \mid w_t; \theta)$$

$$J(\theta) = -\frac{1}{T} \prod_{t=1}^{T} \prod_{\substack{-m \le j \le m \\ j \ne 0}} p(w_{t+j} \mid w_t; \theta)$$

$$Minimize \quad L = -\sum_{t=1}^{T} \sum_{\substack{-m \le j \le m \\ j \ne 0}} \log[p(w_{t+j} \mid w_t; \theta)]$$

Modelling *P(context word|input word)* *(1/2)*

- We want, say, *P('bark'|'dog')*
- Take the weight vector **FROM** 'dog' neuron **TO** projection layer (call this u_{dog})
- Take the weight vector **TO** 'bark' neuron **FROM** projection layer (call this v_{bark})
- When initialized u_{dog} and v_{bark} give the initial estimates of word vectors of 'dog' and 'bark'
- The weights and therefore the word vectors get fixed by back propagation

Modelling *P(context word|input word)* (2/2)

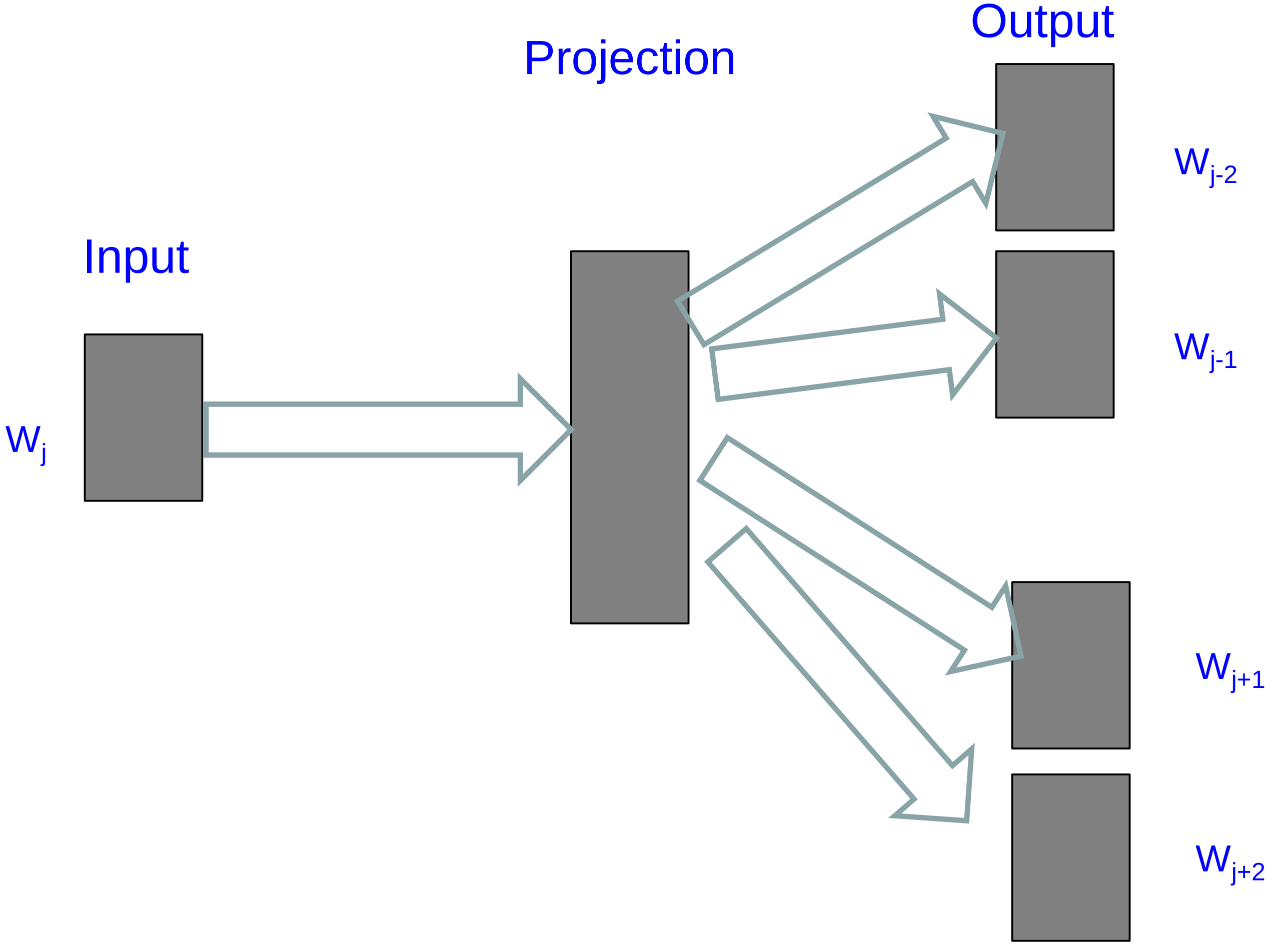
- To model the probability, first compute dot product of u_{dog} and v_{bark}
- Exponentiate the dot product
- Take softmax over all dot products over the whole vocabulary

$$P('bark'|'dog') = \frac{\exp(u_{dog}^{T} v_{bark})}{\sum_{v_k \varepsilon Vocabulary} \exp(u_{dog}^{T} v_k)}$$

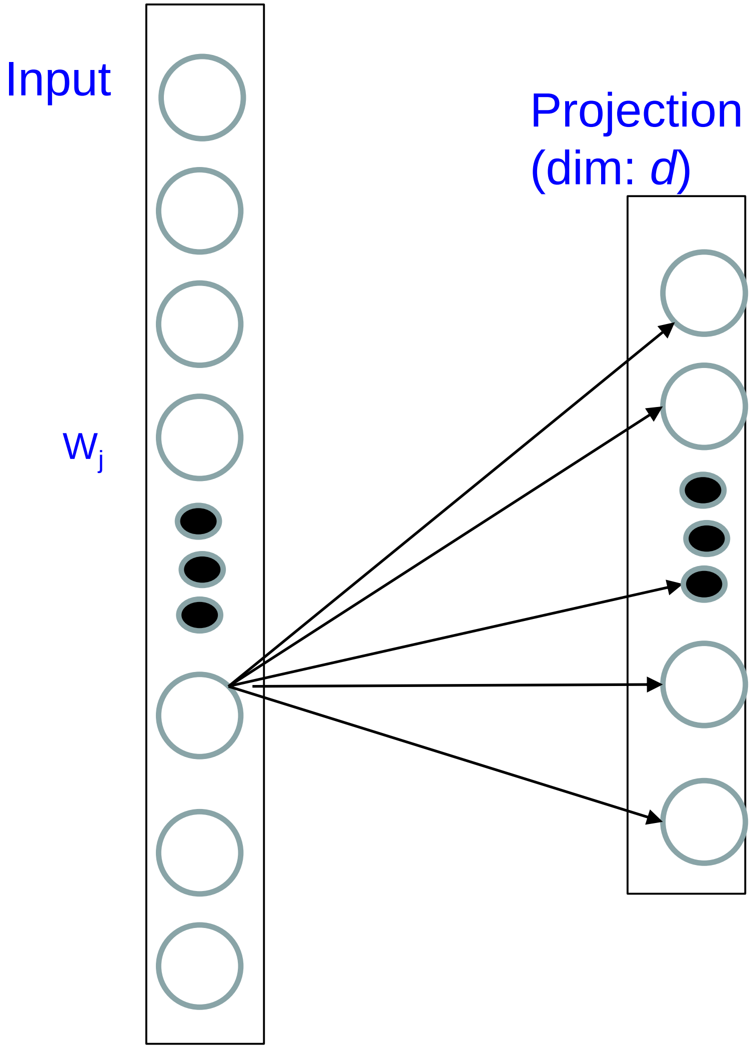
Exercise

- Why cannot you model *P('bark'|'dog')* as the ratio of counts of <bark, dog> and <dog> in the corpus?
- Why this way of modelling probability through dot product of weight vectors of input and output words, exponentiation and soft-maxing works?

Modelling $p(w_{t+j}|w_t)$

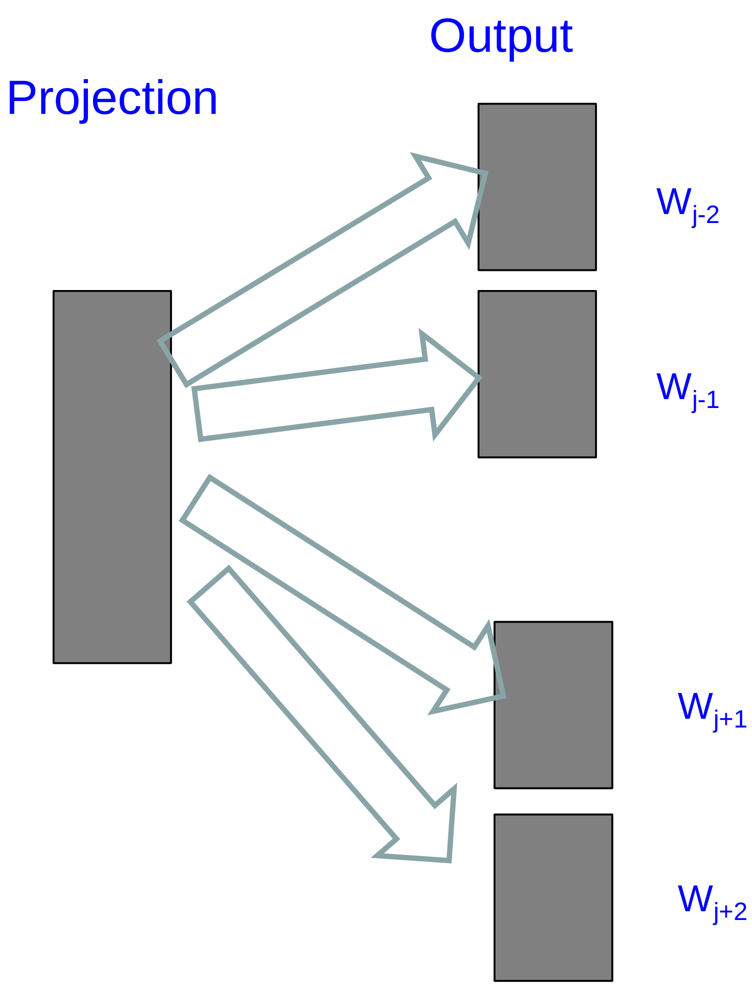


Input to Projection (shown for one neuron only)



- From each input neuron, a weight vector of dim *d*
- Input vector is of dim *V*, where V is the vocab size
- Input to projection we have a weight matrix *W* which is *V X d*
- Each row gives the weight vector of dim *d* REPRESENTING that word
- E.g., rows for ‘dog’, ‘cat, ‘lamp’, ‘table’ etc.

Projection to output



- From the whole projection layer a weight vector of dim *d* to each neuron in each compartment, where the compartment represents a context word
- Each fat arrow is a *d X V* matrix

Capturing word association

Basic concept: Co-occurrence Matrix

Corpora: I enjoy cricket. I like music. I like deep learning

	I	**enjoy**	**cricket**	**like**	**music**	**deep**	**learning**
I	-	1	1	2	1	1	1
enjoy	1	-	1	0	0	0	0
cricket	1	1	-	0	0	0	0
like	2	0	0	-	1	1	1
music	1	0	0	1	-	0	0
deep	1	0	0	1	0	-	1
learning	1	0	0	1	0	1	-

Co-occurence Matrix

Fundamental to NLP

Also called **Lexical Semantic Association (LSA)**

Very sparse, many 0s in each row

Apply Principal Component Analysis (PCA) or Singular Value Decomposition (SVD)

Do Dimensionality Reduction; merge columns with high internal affinity (e.g., *cricket* and *bat*)

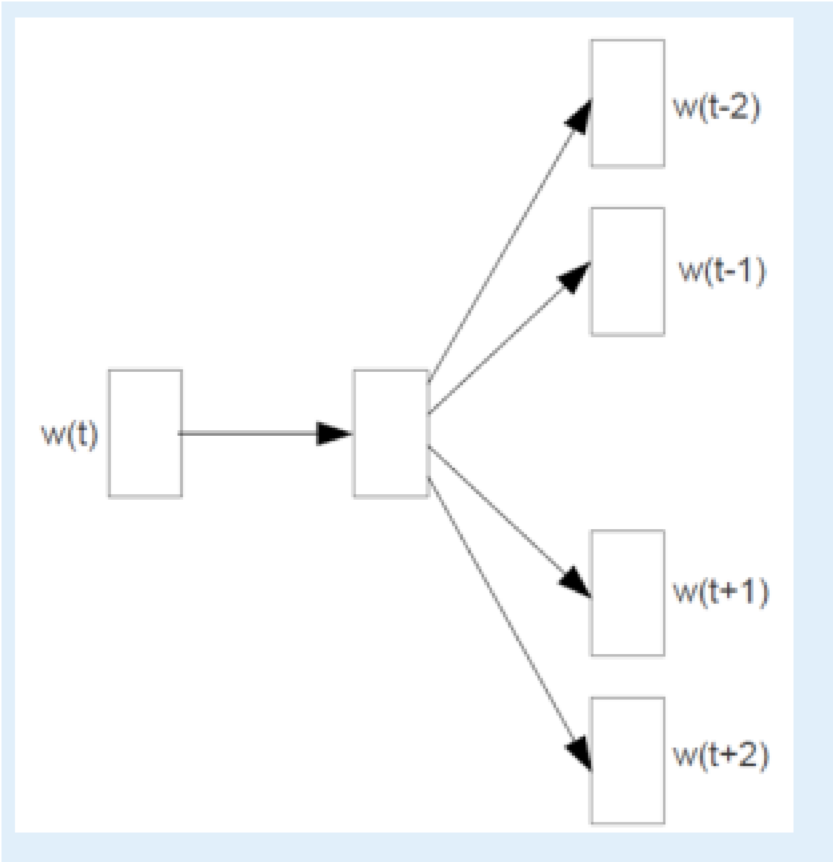
Compression achieves better semantics capture

Linguistic foundation of word representation by vectors

“Linguistics is the eye”: Harris Distributional Hypothesis

- Words with similar distributional properties have similar meanings. (Harris 1970)
- 1950s: Firth- “A word is known by the company its keeps”
- Model **differences** in meaning rather than the proper meaning itself

“Computation is the body”: Skip gram- predict context from word



For CBOW:

Just reverse the Input-Ouput

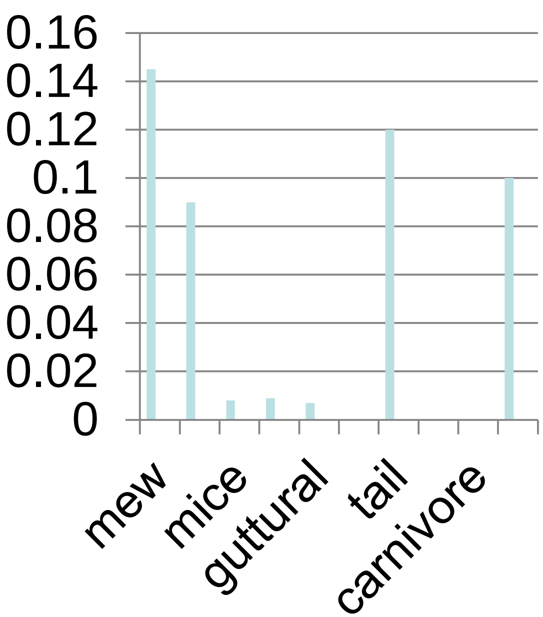
Dog – Cat - Lamp

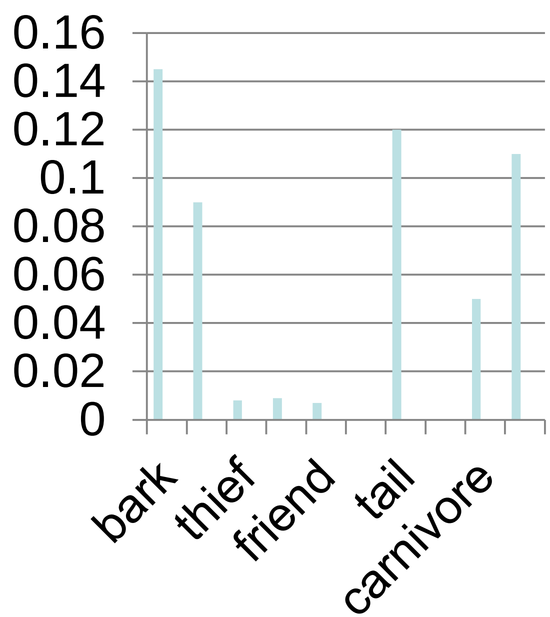
{bark, police, thief, vigilance, faithful, friend, animal, milk, carnivore)

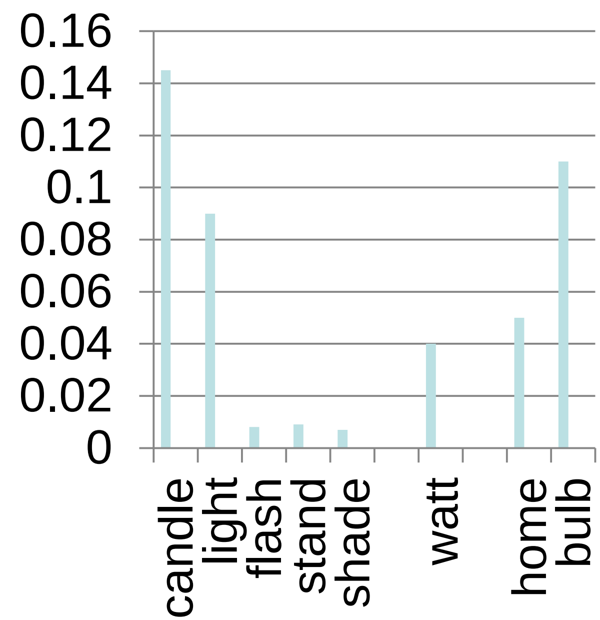
{mew, comfort, mice, furry, guttural, purr, carnivore, milk}

{candle, light, flash, stand, shade, Halogen}

Probability distributions of context words
CE(dog, lamp) > CE(dog, cat)







Test of representation

- **Similarity**
  - 'Dog' more similar to 'Cat' than 'Lamp', because
  - Input- vector('dog'), output- vectors of associated words
  - More similar to output from vector('cat') than from vector('lamp')

“Linguistics is the eye, Computation is the body”

The encode-decoder deep learning network is nothing but

the ***implementation*** of

Harris’s Distributional Hypothesis

Fine point in Harris Distributional Hypothesis

- Words with similar distributional properties have similar meanings. (Harris 1970)
- Harris does mentions that distributional approaches can model differences in meaning rather than the proper meaning itself

Representation Learning

Basics

- What is a good representation? At what granularity: words, n-grams, phrases, sentences
- Sentence is important- (a) *I <u>bank</u> with SBI; (b) I took a stroll on the river <u>bank</u>; (c) this <u>bank</u> sanctions loans quickly*
- Each ‘bank’ should have a different representation
- We have to LEARN these representations

Principle behind representation

- Proverb: “A man is known by the company he keeps”
- Similarly: “A word is known/represented by the company it keeps”
- “Company” → Distributional Similarity

Starting point: 1-hot representation

- Arrange the words in lexicographic order
- Define a vector *V* of size |*L*|, where *L* is the lexicon
- For word w_i in the i^{th} position, set the ith bit to 1, all other bits being 0.
- Problem: cosine similarity of ANY pair is 0; wrong picture!!

Representation: to learn or not learn?

- 1-hot representation does not capture many nuances, e.g., semantic similarity
  - But is a good starting point
- Co-occurences also do not fully capture all the facets
  - But is a good starting point

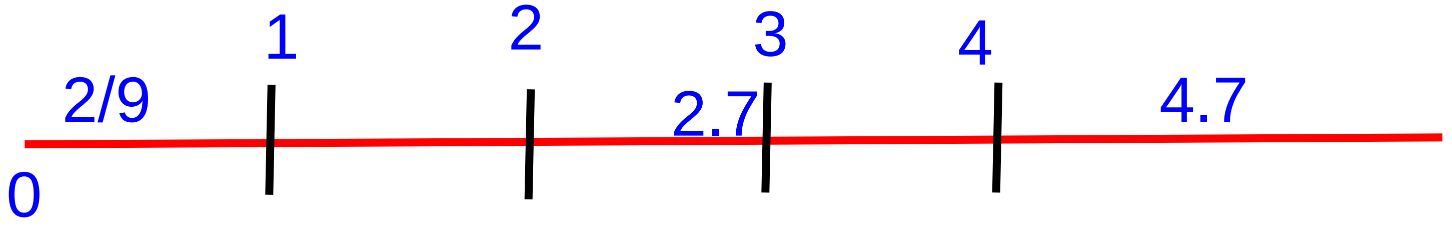
So learn the representation…

- Learning Objective

- ***MAXIMIZE CONTEXT PROBABILITY***

Foundations-1: Embedding

- Way of taking a discrete entity to a continuous space
- E.g., 1, 2, 3, 2.7, 2/9, $22^{1/2}$, … are numerical symbols
- But they are points on the real line
- Natural embedding
- Words' embedding not so intuitive!



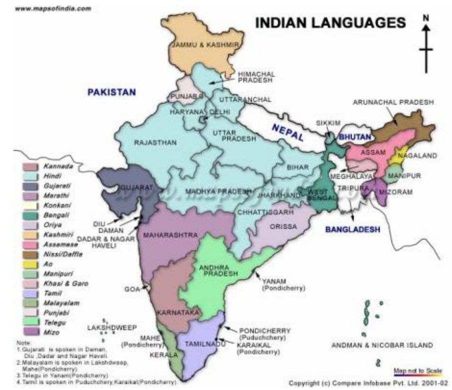
Foundations-2: Purpose of Embedding

- Enter geometric space
- Take advantage of “distance measures”- Euclidean distance, Riemannian distance and so on
- “Distance” gives a way of computing similarity

Foundations-3: Similarity and difference

- Recognizing similarity and difference- foundation of intelligence
- Lot of Pattern Recognition is devoted to this task (Duda, Hart, Stork, 2nd Edition, 2000)
- Lot of NLP is based on Text Similarity
- Words, phrases, sentences, paras and so on (verticals)
- Lexical, Syntactic, Semantic, Pragmatic (Horizontal)

Similarity study in MT



English:

This blanket is very soft

Hindi:

yaha kambal bahut naram hai

Bangla:

ei kambal ti khub naram <null>

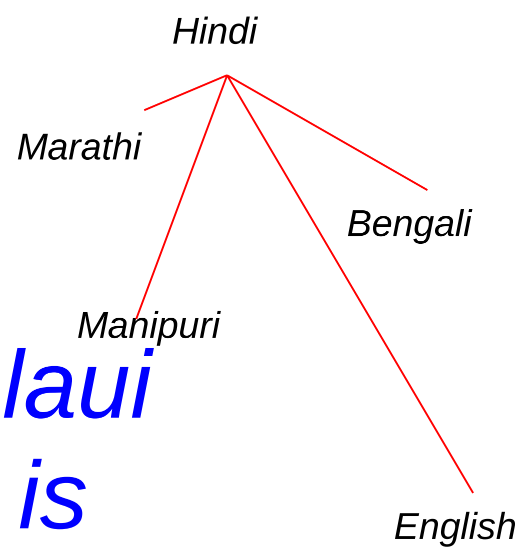
Marathi:

haa kambal khup naram aahe

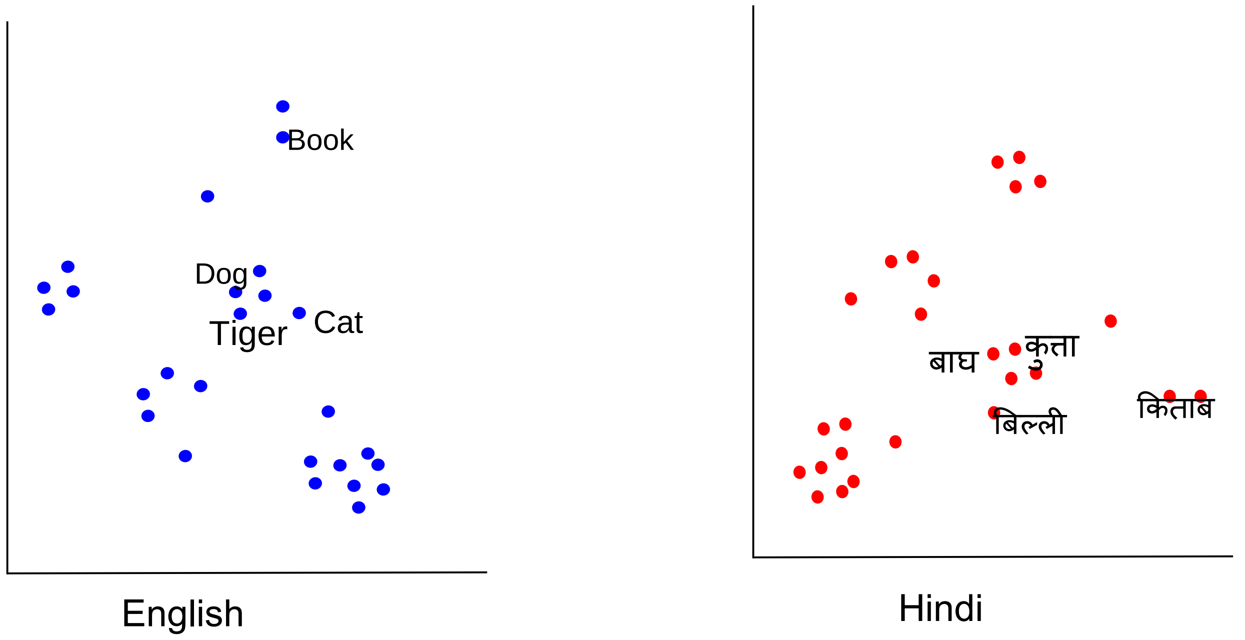
Manipuri:

kampor asi mon mon laui

blanket this soft soft is

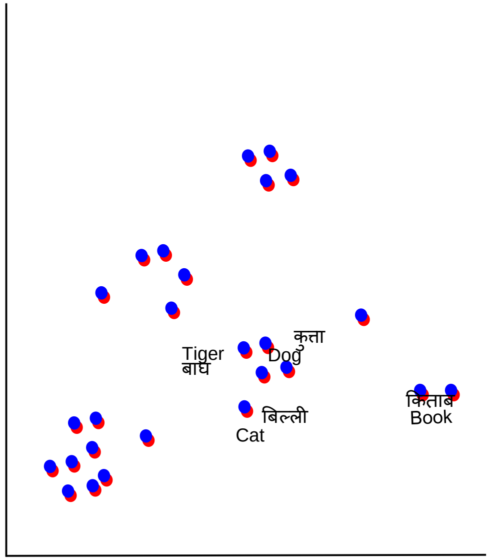


ISO-Metricity



Across Cross-lingual Mapping

This involves strong assumption that embedding spaces across languages are isomorphic, which is not true specifically for distance languages (Søgaard et al. 2018). However, without this assumption unsupervised NMT is not possible.

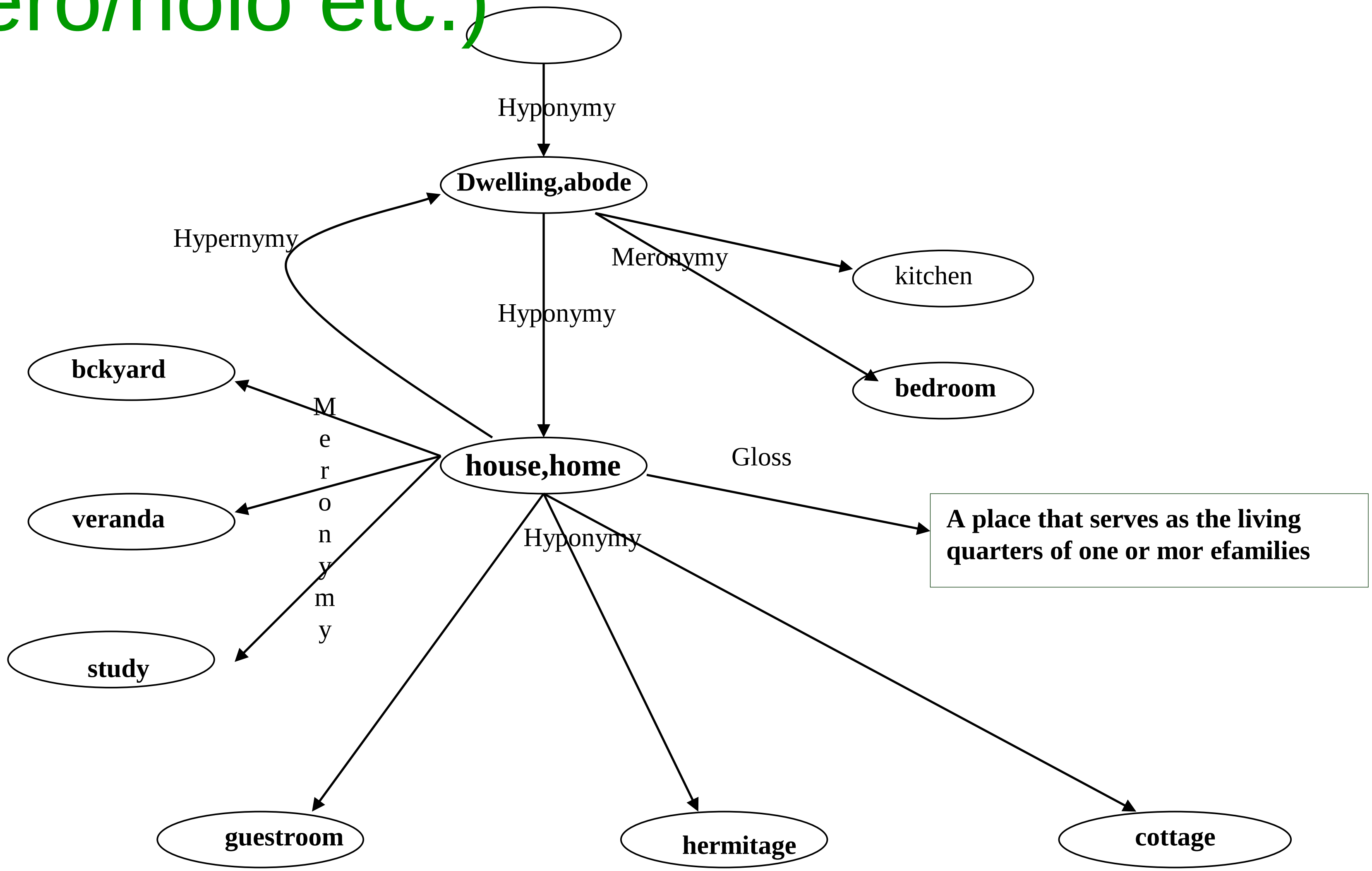


Søgaard, Anders, Sebastian Ruder, and Ivan Vulić. 2018. On the limitations of unsupervised bilingual dictionary induction. ACL

Foundations-4: Syntagmatic and Paradigmatic Relations

- Syntagmatic and paradigmatic relations
  - Lexico-semantic relations: synonymy, antonymy, hypernymy, mernymy, troponymy etc. **CAT is-a ANIMAL**
  - Coccurence: **CATS MEW**
- Wordnet: primarily paradigmatic relations
- ConceptNet: primarily Syntagmatic Relations

WordNet Sub-Graph with lexico-semantic relations (hyper/hypo, mero/holo etc.)



Lexical and Semantic relations in wordnet

1. Synonymy (e.g., *house, home*)
2. Hypernymy / Hyponymy (kind-of, e.g., *cat* ←→ *animal*)
3. Antonymy (e.g., *white and black*)
4. Meronymy / Holonymy (part of, e.g., *cat and tail*)
5. Gradation (e.g., *sleep*→*doze*→*wake up*)
6. Entailment (e.g., *snoring* → *sleeping*)
7. Troponymy (manner of, e.g., *whispering and talking*)

1, 3 and 5 are lexical (*word to word)*, rest are semantic (*synset to synset).*

‘Paradigmatic Relations’ and ‘Substitutability’

- Words in paradigmatic relations can substitute each other in the sentential context
- E.g., ‘The cat is drinking milk’ → ‘The animal is drinking milk’
- Substitutability is a foundational concept in linguistics and NLP

Foundations-5: Learning and Learning Objective

- Probability of getting the context words given the target should be maximized (skip gram)

- Probability of getting the target given context words should be maximized (CBOW)

Learning objective (skip gram)

$$J^{'}(\theta) = \frac{1}{T}\prod_{t=1}^{T}\prod_{\substack{-m \leq j \leq m \\ j \neq 0}} p(w_{t+j} \mid w_t; \theta)$$

$$J(\theta) = -\frac{1}{T}\prod_{t=1}^{T}\prod_{\substack{-m \leq j \leq m \\ j \neq 0}} p(w_{t+j} \mid w_t; \theta)$$

$$Minimize \quad L = -\sum_{t=1}^{T}\sum_{\substack{-m \leq j \leq m \\ j \neq 0}} \log[p(w_{t+j} \mid w_t; \theta)]$$

Modelling *P(context word|input word)* *(1/2)*

- We want, say, *P('bark'|'dog')*
- Take the weight vector **FROM** 'dog' neuron **TO** projection layer (call this u_{dog})
- Take the weight vector **TO** 'bark' neuron **FROM** projection layer (call this v_{bark})
- When initialized u_{dog} and v_{bark} give the initial estimates of word vectors of 'dog' and 'bark'
- The weights and therefore the word vectors get fixed by back propagation

Modelling *P(context word|input word)* *(2/2)*

- To model the probability, first compute dot product of u_{dog} and v_{bark}
- Exponentiate the dot product
- Take softmax over all dot products over the whole vocabulary

$$P('bark'|'dog') = \frac{\exp(u_{dog}^T v_{bark})}{\sum_{v_k \varepsilon Vocabulary} \exp(u_{dog}^T v_k)}$$

Exercise

- Why cannot you model *P('bark'|'dog')* as the ratio of counts of <bark, dog> and <dog> in the corpus?
- Why this way of modelling probability through dot product of weight vectors of input and output words, exponentiation and soft-maxing works?

Possible project ideas

Semantics Extraction using Universal Networking Language

Sentence: *I went with my friend, John, to the bank to withdraw some money but was disappointed to find it closed.*

Part Of Speech

↓

Named Entity Recognition

↓

Word Sense Disambiguation

↓

Co-reference

Current work:

Combine Machine learning with rule Based technique
(Janardhan)

Agt(go,I)
Ptn(go,friend)
Nam(friend,John)
Plt(go,bank)
Pur(go, withdraw)
Obj(withdraw,money0
Mod(money,some)
And(go,disappoint)

Sentiment Analysis

“The water is boiling.”: Objective

“He is boiling with anger.”: Negative

Current work:

1. *Tweet and Blog Sentiment*
2. *Indian Language Sentiment Analysis*
3. *Word Sense and Sentiment*
4. *Thwarting and*

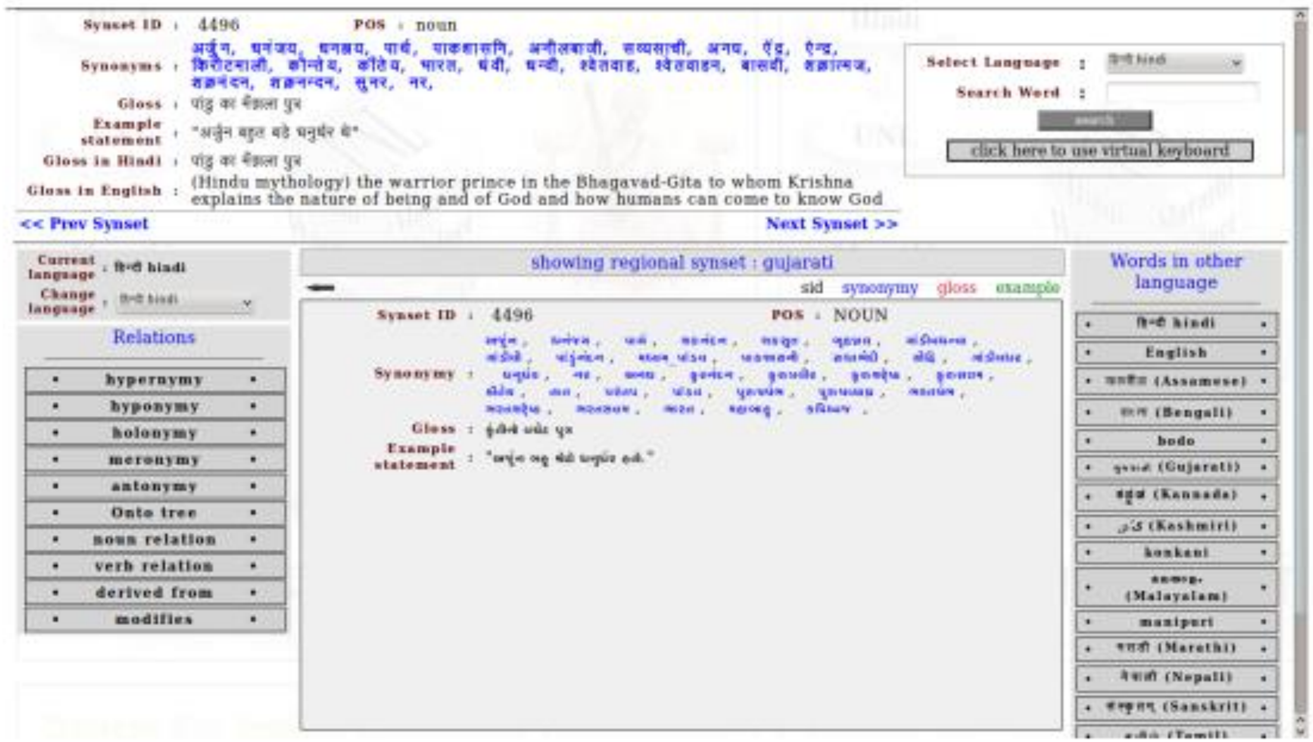
   (Subhabrata and Akshat, Balamurali)

Text Entailment

	TEXT	HYPOTHESIS	ENTAIL-MENT
1	*. The Hubble is the only large visible light and ultra-violet space telescope we have in operation.*	*Hubble is a Space telescope.*	True
2	*Google files for its long awaited IPO.*	*Google goes public.*	True
3	*After the deal closes, Teva will earn about $7 billion a year, the company said.*	*Teva earns $7 billion a year.*	False

Current work: Do entailment from Semantic Graphs (Arindam, Janradhan)

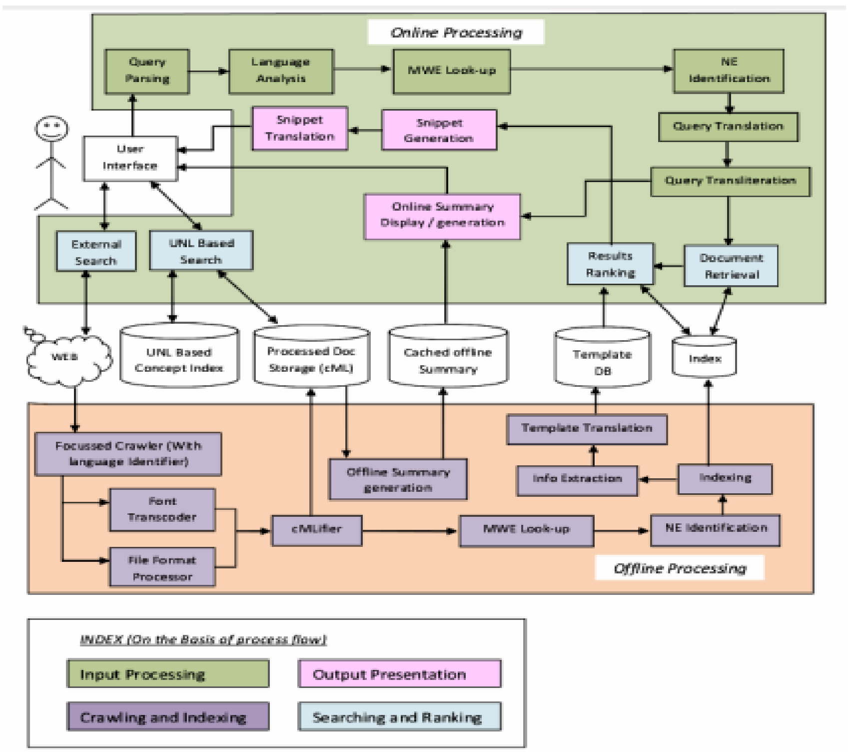
Indowordnet and Multilingual Word Sense Disambiguation



Current work: Linking wordnets with SUMO Ontology; using resources of one Language for another for WSD (Salil Joshi, Arindam Chatterjee, Brijesh, Mitesh)

Cross Lingual Information Retrieval

Architecture of Sandhan



Current work: Performance Enhancement; Query expansion and disambiguation
(Yogesh, Arjun, Swapnil)

Machine Translation

Large Projects funded by

Yahoo, Xerox, Ministry of IT

Current work:

1. *Indian Language to Indian Language*
2. *Statistical MT*
3. *Crowdsourcing and MT*
4. *Semantics and SMT*

   (Mitesh, Anoop, Victor, Somya, Abhijit, Raj, Rahul)

Sites:

http://www,cse.iitb.ac.in/~pb

http://www.cfilt.iitb.ac.in