CS290D – Advanced Data Mining

Instructor: Xifeng Yan
Computer Science
University of California at Santa Barbara

Paraphrasing

Lecturer: Izzeddin Gur
Computer Science
University of California at Santa Barbara

Source of slides

- Shiqi Zhao's COLING Tutorial
- Richard Socher's EMNLP Presentation

Outline

- **Introduction**
- Paraphrase Identification
- Paraphrase Extraction

Definition

- Paraphrase
  - Noun
    - Alternative expression of the same meaning
  - Verb
    - Generate paraphrases for the input expression
- “same meaning” ?
  - Quite subjective
  - Different degrees of strictness
  - Depend on applications

Paraphrase (noun): Alternative expressions of the same meaning

Korean Kim Yuna won gold with a world-record score in women's figure skating at the Vancouver Olympics Thursday.

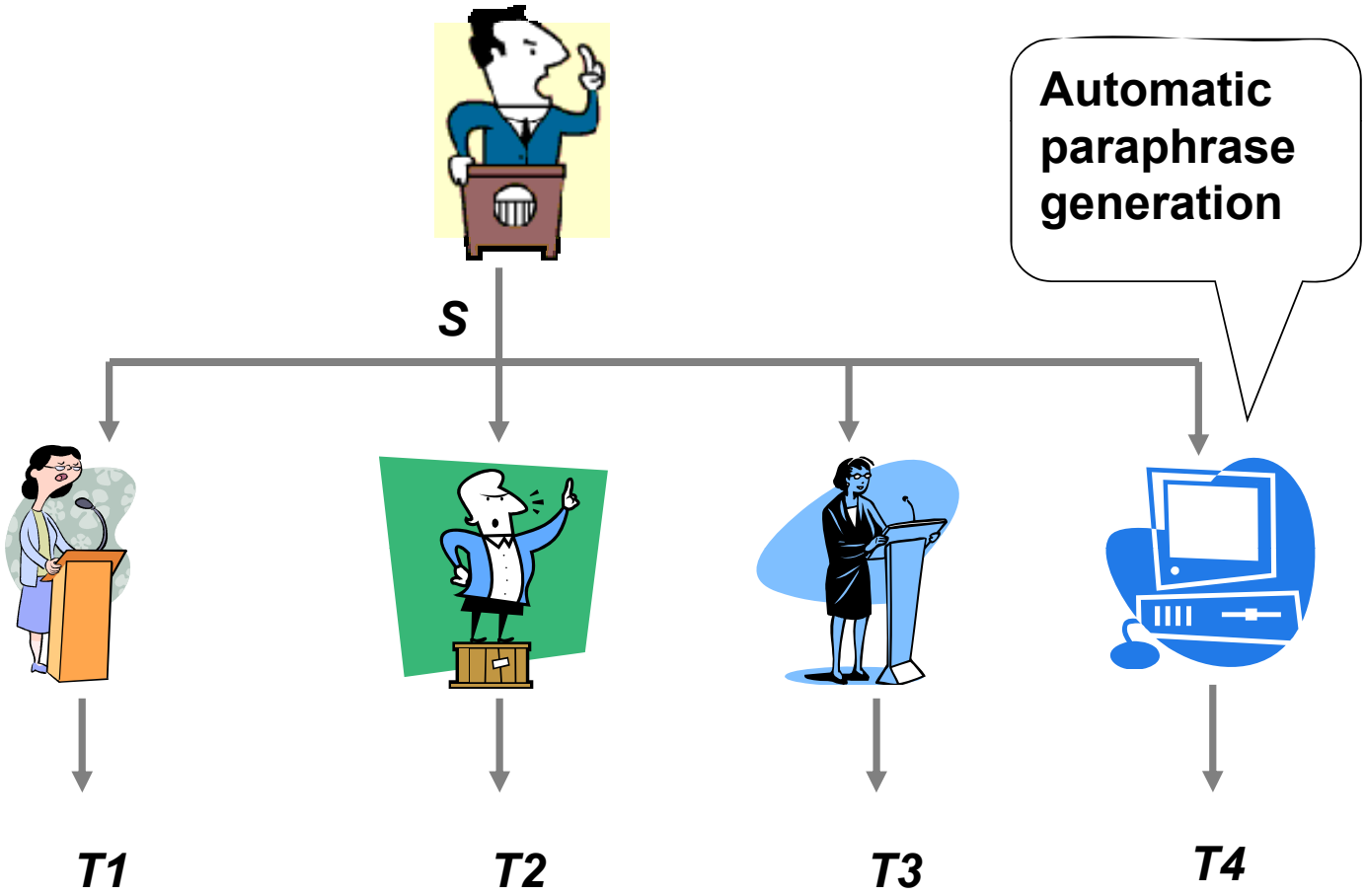
Korean figure skater Kim Yuna has won the gold medal of women's figure skating at the Winter Olympics in Vancouver

Kim Yu-Na (19) is a South Korean ice skater who took the gold medal at the Vancouver Olympics.

Yuna Kim of South Korea won the women's figure skating gold medal at the Vancouver Olympics in record fashion.

Kim Yuna, a South Korean figure skater has won the gold medal at the on-going Winter Olympics 2010.

Paraphrase (verb): Generate paraphrases for an input S



Classification of Paraphrases

- According to granularity
  - Surface paraphrases
    - Lexical level
    - Phrase level
    - Sentence level
    - Discourse level
  - Structural paraphrases
    - Pattern level
    - Collocation level

Examples

- Lexical paraphrases (generally synonyms)
  - *solve* and *resolve*
- Paraphrase phrases
  - *look after* and *take care of*
- Paraphrase sentences
  - *The table was set up in the carriage shed.*
  - *The table was laid under the cart-shed*.
- Paraphrase patterns
  - *[X] considers [Y]*
  - *[X] takes [Y] into consideration*
- Paraphrase collocations
  - *(turn on, OBJ, light)*
  - *(switch on, OBJ, light)*

Classification of Paraphrases

- According to paraphrase style
  - Trivial change
  - Phrase replacement
  - Phrase reordering
  - Sentence split & merge
  - Complex paraphrases

Examples

- Trivial change
  - all the members of and all members of
- Phrase replacement
  - He said there will be major cuts in the salaries of high-level civil servants.
  - He said there will be major cuts in the salaries of senior officials.
- Phrase reordering
  - Last night, I saw Tom in the shopping mall.
  - I saw Tom in the shopping mall last night.
- Sentence split & merge
  - He bought a computer, which is very expensive.
  - (1) He bought a computer. (2) The computer is very expensive.
- Complex paraphrases
  - He said there will be major cuts in the salaries of high-level civil servants.
  - He claimed to implement huge salary cut to senior civil servants.

Applications of Paraphrases

- Machine Translation (MT)
  - Simplify input sentences
  - Alleviate data sparseness
  - Parameter tuning
  - Automatic evaluation
- Question Answering (QA)
  - Question reformulation
- Information Extraction (IE)
  - IE pattern expansion
- Information Retrieval (IR)
  - Query reformulation
- Summarization
  - Sentence clustering
  - Automatic evaluation
- Natural Language Generation (NLG)
  - Sentence rewriting
- Others
  - Changing writing style
  - Text simplification
  - Identifying plagiarism
  - Text steganography

Research on Paraphrasing

- Paraphrase Identification
  - Identify (sentential) paraphrases
- Paraphrase extraction
  - Extract paraphrase instances (different granularities)
- Paraphrase generation
  - Generate sentential paraphrases
- Paraphrase application
  - Apply paraphrases in other areas

Textual Entailment - A similar direction

- Textual entailment
  - A directional relation between two text fragments
    - ***T***: the entailing text
    - ***H***: the entailed hypothesis
  - ***T*** entails ***H*** if, typically, a human reading ***T*** would infer that ***H*** is most likely true.
  - Compare entailment with paraphrase
    - Paraphrase is bidirectional entailment

Textual Entailment - A similar direction

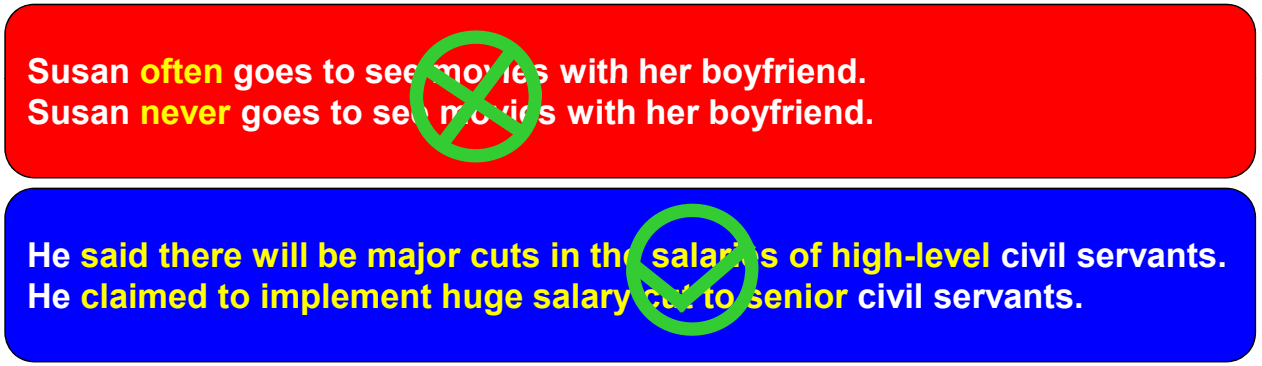
- **R**ecognizing **T**extual **E**ntailment Track (RTE)
  - RTE-1 (2004) to RTE-5 (2009)
  - RTE-6 (2010) is in progress
- Example
  - ***T*:** A shootout at the Guadalajara airport in May, 1993, killed Cardinal Juan Jesus Posadas Ocampo.
  - ***H*:** Juan Jesus Posadas Ocampo died in 1993.

Outline

- Introduction
- **Paraphrase Identification**
- Paraphrase Extraction

Paraphrase Identification

- Specially refers to sentential paraphrase identification
  - Given any pair of sentences, automatically identifies whether these two sentences are paraphrases
- Paraphrase identification is not trivial



Overview

- Alignment based methods
  - Align ***s***1 and ***s***2 first, and score the sentence pair based on the alignment results
- Unsupervised methods

- Supervised methods
  - Reviewed as a binary classification problem, i.e., input ***s***1 and ***s***2 to a classifier and output 0/1
  - Compute the similarities between ***s***1 and ***s***2 at different levels, which are then used as classification features

Corpus-based and Knowledge-based Measures (Mihalcea '06)

- Semantic Similarity of Text Segments
  - Reduced to word level matching
- Semantic Similarity of Words
  - Corpus-based Measures
  - Knowledge-based Measures

Corpus-based and Knowledge-based Measures

- Semantic Similarity of Text Segments
  - Match each word to most similar in other text
  - Weight w/ respect to specificity

$$sim(T_1, T_2) = \frac{1}{2}\left(\frac{\sum_{w \in \{T_1\}} (maxSim(w, T_2) * idf(w))}{\sum_{w \in \{T_1\}} idf(w)} + \frac{\sum_{w \in \{T_2\}} (maxSim(w, T_1) * idf(w))}{\sum_{w \in \{T_2\}} idf(w)}\right)$$

$$sim(T_1, T_2) \geq 0.5 \quad \text{paraphrase}$$

Corpus-based and Knowledge-based Measures

- Semantic Similarity of Words
- Corpus-based Measures
  - Pointwise Mutual Information
  - Latent Semantic Analysis
- Knowledge-based Measures

Corpus-based and Knowledge-based Measures

- Pointwise Mutual Information

$$PMI\text{-}IR(w_1, w_2) = log_2 \frac{p(w_1 \& w_2)}{p(w_1) * p(w_2)}$$

- Estimate probabilities

$$p_{NEAR}(w_1 \& w_2) \simeq \frac{hits(w_1\ NEAR\ w_2)}{WebSize} \qquad p(w_i) = hits(w_1)/WebSize$$

$$PMI\text{-}IR(w_1, w_2) \simeq log_2 \frac{hits(w_1\ AND\ w_2) * WebSize}{hits(w_1) * hits(w_2)}$$

Corpus-based and Knowledge-based Measures

- Knowledge-based measures (WordNet::Similarity)
- Leacock&Chodorow

$$Sim_{lch} = -\log \frac{length}{2 * D}$$

- Resnik

$$Sim_{res} = IC(LCS)$$

$$IC(c) = -\log P(c)$$

Corpus-based and Knowledge-based Measures

- Example

Text Segment 1: When the defendant and his lawyer walked into the court, some of the victim supporters turned their backs on him.
Text Segment 2: When the defendant walked into the courthouse with his attorney, the crowd turned their backs on him.

Text 1	Text 2	maxSim	idf
defendant	defendant	1.00	3.93
lawyer	attorney	0.89	2.64
walked	walked	1.00	1.58
court	courthouse	0.60	1.06
victims	courthouse	0.40	2.11
supporters	crowd	0.40	2.15
turned	turned	1.00	0.66
backs	backs	1.00	2.41

Word similarity scores and word specificity (idf)

Semantic Similarity Approach (Fernando '08)

- All similarity between all pairs
- Semantic Similarity Matrix (W)
  - Use knowledge-based measures to construct W
  - Resnik, Leacock, .etc
- Weighted cosine similarity

$$sim(\vec{a}, \vec{b}) = \frac{\vec{a} W \vec{b^T}}{|\vec{a}||\vec{b}|}$$

- Learn a threshold (*T*)

$$sim(\vec{a}, \vec{b}) \geq T$$

Semantic Similarity Approach

- Example

The dog sat on the mat

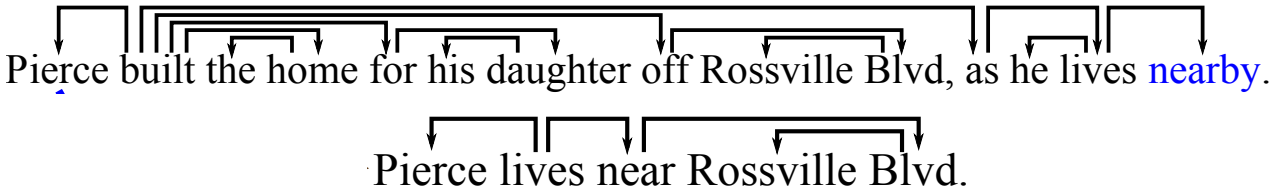
The mutt sat on the rug

	dog	mat	mutt	rug	sat
dog	1	0	0.8	0	0
mat	0	1	0	0.9	0
mutt	0.8	0	1	0	0
rug	0	0.9	0	1	0
sat	0	0	0	0	1

Sample similarity matrix showing similarity scores for content words from two sentences.

Tree Edit Models (Heilman '10)

- Each sentence is represented by its dependency tree



- Each sentence pair is represented by a sequence of tree edit operations
  - *INSERT, DELETE, RELABEL, etc.*
- Extract features from sequences
  - *#INSERT, #DELETE, etc.*
- Classify sequences using logistic regression

Tree Edit Models

- Tree Edit Sequences
  - Find short sequences of tree edits
  - Exponentially large
  - State space model with a Greedy Best -First search
    - T_c is the current tree (initially, DT of the first sentence)
    - T_t is the target tree (DT of the second sentence)
    - Explore next states (tree) n(T_c) by tree edit operations
    - Choose next state by $H(T_c^{`})$

Tree Edit Models

- Tree Edit Sequences
  - Objective : Find short sequences of tree edits
  - Exponentially large
  - State space model with a Greedy Best -First search
    - T_c is the current tree (initially, DT of the first sentence)
    - T_t is the target tree (DT of the second sentence)
    - Explore next states (tree) n(T_c) by tree edit operations
    - Choose next state by $H(T_c^{`})$

Tree Edit Models

Tree Edit Operations

Operation	Arguments	Description
INSERT-CHILD	node index j, new lemma l, POS p, edge label e, side $s \in \{left, right\}$	Insert a node with lemma l, POS p, and edge label e as the last child (i.e., farthest from parent) on side s of $T(j)$.
INSERT-PARENT	non-root node index j, new lemma l, new POS p, edge label e, side $s \in \{left, right\}$	Create a node with lemma l, POS p, and edge label e. Make $T(j)$ a child of the new node on side s. Insert the new node as a child of the former parent of $T(j)$ in the same position.
DELETE-LEAF	leaf node index j	Remove the leaf node $T(j)$.
DELETE-&-MERGE	node index j (s.t. $T(j)$ has exactly 1 child)	Remove $T(j)$. Insert its child as a child of $T(j)$'s former parent in the same position.
RELABEL-NODE	node index j, new lemma l, new POS p	Set the lemma of $T(j)$ to be l and its POS to be p.
RELABEL-EDGE	node index j, new edge label e	Set the edge label of $T(j)$ to be e.
MOVE-SUBTREE	node index j, node index k (s.t. $T(k)$ is not a descendant of $T(j)$), side $s \in \{left, right\}$	Move $T(j)$ to be the last child on the s side of $T(k)$.
NEW-ROOT	non-root node index j, side $s \in \{left, right\}$	Make $T(j)$ the new root node of the tree. Insert the former root as the last child on the s side of $T(j)$.
MOVE-SIBLING	non-root node index j, side $s \in \{left, right\}$, position $r \in \{first, last\}$	Move $T(j)$ to be the r child on the s side of its parent.

Tree Edit Models

- Tree Kernel Heuristic

$$H(T_c) = 1 - \frac{K(T_c, T_t)}{\sqrt{K(T_c, T_c) \times K(T_t, T_t)}}$$

- Recursive Tree Kernel

$$K(T_1, T_2) = \sum_{n_1 \in \{N_{T_1}\}} \sum_{n_2 \in \{N_{T_2}\}} \Delta(n_1, n_2)$$

$$\Delta(n_1, n_2) = \mu \left(\lambda^2 s(n_1, n_2) + \sum_{J_1, J_2, |J_1| = |J_2|} \prod_{i=1}^{l(J_1)} \Delta(c_{n_1}[J_{1i}], c_{n_2}[J_{2i}]) \right)$$

- Node similarity

$$s(n_1, n_2) = \delta(l(n_1), l(n_2)) \times \sum_{f \in \{l,e,p,s\}} \delta(f(n_1), f(n_2))$$

Tree Edit Models

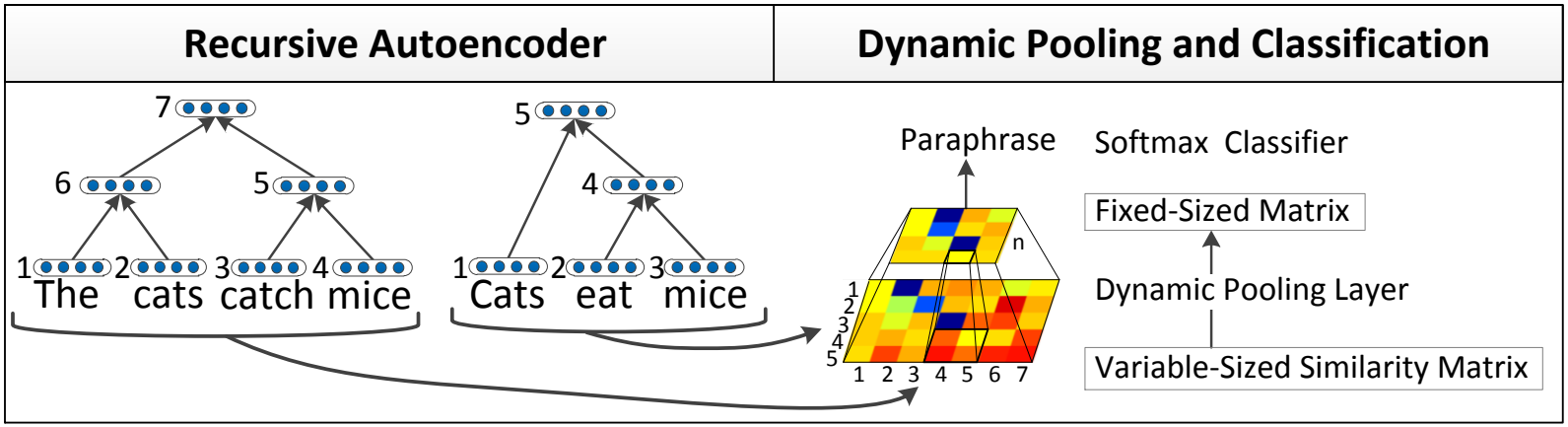
- Classification
  - Extract features

Feature	Description
`totalEdits`	# of edits in the sequence.
X`Edits`	#s of X edits (where X is one of the nine edit types in Table 1).
`relabelSamePOS`, `relabelSameLemma`, `relablePronoun`, `relabelProper`, `relabelNum`	#s of `RELABEL-NODE` edits that: preserve POS, preserve lemmas, convert between nouns and pronouns, change proper nouns, change numeric values by more than 5% (to allow rounding), respectively.
`insertVorN`, `insertProper`	#s of `INSERT-CHILD` or `INSERT-PARENT` edits that: insert nouns or verbs, insert proper nouns, respectively.

  - Train a logistic regression

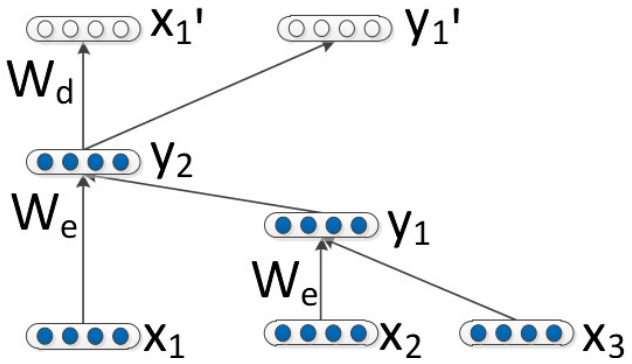
Recursive AutoEncoders (RAE) (Socher '11)

- Given binary parse trees of two sentences
- Build a RAE on the parse trees
- Estimate similarity matrix
- Build a classifier on similarity matrix



Recursive AutoEncoders (RAE)

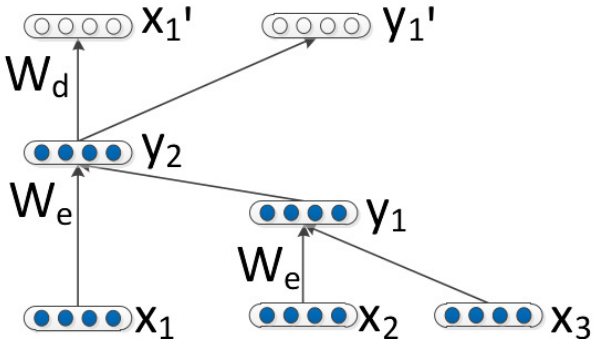
- A binary parse tree is a list of triplets of parents with children:
  $(p \rightarrow (c_1, c_2))$
- c_1, c_2 are either a terminal word vector $x_i \in R^n$ or a non-terminal parent $y_1 \in R^n$



Parse tree for $((y_1 \rightarrow x_2 x_3), (y_2 \rightarrow x_1 y_1)), \forall x, y \in \mathbb{R}^n$

Recursive AutoEncoders (RAE)

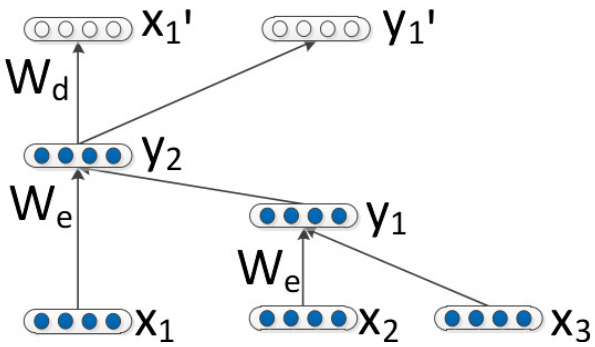
- Non-terminal parent p computed as
  $$p = f(W_e[c1;c2]+b)$$
- f is an activation function (eg. sigmoid, tanh)
- $W_e \in R^{n \times 2n}$ the encoding matrix to be learned
- $[c1;c2] \in R^{2n}$ is the concatenated children
- b is a bias term



$$y_2 = f\left(W_e\left[\begin{array}{c} x_1 \\ x_2 \\ x_3 \end{array}\right] + b_1\right)\Bigg] + b_2\Bigg)$$

Recursive AutoEncoders (RAE)

- W_d inverts W_e s.t. $[c_1' ; c_2']=f(W_d p+b_d)$ is the decoding of p
- $E_{rec}(p) = \|[c_1; c_2] - [c_1' ; c_2']\|$
- To train :
  - Minimize $E_{rec}(\mathcal{T}) = \sum_{p \in \mathcal{T}} E_{rec}(p) = E_{rec}(y_1) + E_{rec}(y_2)$

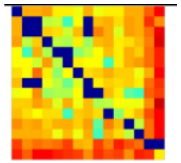


$$y_2 = f\left(W_e \left[f\left(W_e \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + b_1\right) \right] + b_2\right)$$

Recursive AutoEncoders (RAE)

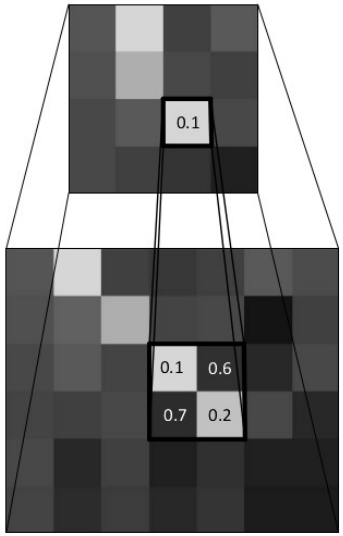
- Sentence Similarity Matrix
  - For two sentences S1,S2 of lengths n and m, concatenate terminal x_i s (in sentence order) with non-terminal y_i s (depth-first, right-to-left)
  - Compute similarity matrix $\mathcal{S} \in \mathbb{R}^{(2v-1)\times(2w-1)}$, where $\mathcal{S}_{i,j}$ is the ℓ_2-norm between the ith element from $\mathcal{S}1$'s feature vector and the jth element from $S2$'s feature vector

(1) LLEYTON Hewitt yesterday traded his tennis racquet for his first sporting passion - Australian football - as the world champion relaxed before his Wimbledon title defence
(2) LLEYTON Hewitt yesterday traded his tennis racquet for his first sporting passion- Australian rules football-as the world champion relaxed ahead of his Wimbledon defence



Recursive AutoEncoders (RAE)

- Sentence lengths may vary $\implies$ $\mathcal{S}$ dimensionality may vary.
  Want to map $\mathcal{S} \in \mathbb{R}^{(2n-1)\times(2m-1)}$ to $\mathcal{S}_{pooled} \in \mathbb{R}^{n_p \times n_p}$ with n_p constant
- *Dynamically* partition rows and columns of $\mathcal{S}$ into n_p equal parts
- Min. pool over each part
- Normalize $\mu = 0, \sigma = 1$ and pass on to classifier (e.g. softmax)



Machine Translation (MT) Metrics (Madnani '12)

- Consider the first sentence S1 as ground truth and second sentence S2 as a translation from a hypothetical language
- Evaluate the effectiveness of the translation by a MT metric
- Each metric gives a probability estimation
- Build a meta-classifier by averaging
  - Logistic regression
  - SVM
  - Instance based Nearest Neighbor

Machine Translation (MT) Metrics

- MT Metrics
  - **BLEU**
    - Amount of n-gram overlap
  - **NIST**
    - Amount of n-gram overlap with idf weighting
  - **TER**
    - # of string edits (insert, delete, substitutions)
  - **TERp**
    - # of string edits (+ stemming, synonymy, paraphrase)

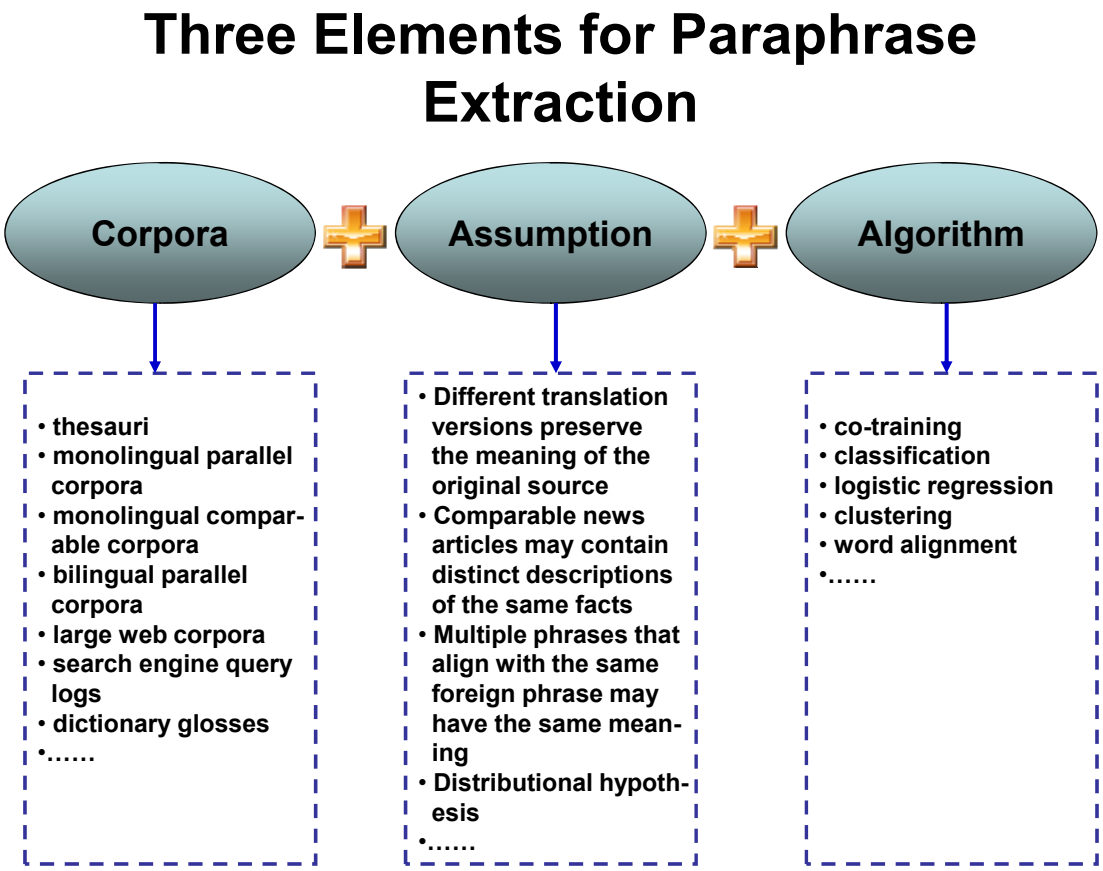
Machine Translation (MT) Metrics

- MT Metrics (con't)
  - **METEOR**
    - BLEU + stemming, synonymy, paraphrase
  - **SEPIA**
    - n-gram overlap with long spans
  - **BADGER**
    - Utilizing Burrows Wheeler Transformation
  - **MAXSIM**
    - Bipartite graph matching

Outline

- Introduction
- Paraphrase Identification
- **Paraphrase Extraction**

Paraphrase Extraction



References

- Corpus-based and Knowledge-based Measures of Text Semantic Similarity, R. Mihalcea, C. Corley, C. Strapparava
- A Semantic Similarity Approach to Paraphrase Detection, S. Fernando, M. Stevenson
- Tree Edit Models for Recognizing Textual Entailments, Paraphrases, and Answers to Questions, M. Heilman, N. A. Smith
- Dynamic Pooling and Unfolding Recursive Autoencoders for Paraphrase Detection, R. Socher, E. H. Huang, J. Pennington, A. Y. Ng, C.D. Manning
- Re-examining Machine Translation Metrics for Paraphrase Identification, N. Madnani, J. Tetreault