Lexical Semantics & WSD

Ling571
Deep Processing Techniques for NLP
February 15, 2017

Roadmap

- Distributional models
  - Representation
  - Compression
  - Integration
- Dictionary-based models
- Thesaurus-based similarity models
  - WordNet
  - Distance & Similarity in a Thesaurus
- Classifier models

Distributional Similarity Questions

- What is the right neighborhood?
  - What is the context?
- How should we weight the features?
- How can we compute similarity between vectors?

Feature Vector Design

- Window size:
  - How many words in the neighborhood?
    - Tradeoff:
      - +/- 500 words: 'topical context'
      - +/- 1 or 2 words: collocations, predicate-argument
      - Only words in some grammatical relation
        - Parse text (dependency)
        - Include subj-verb; verb-obj; adj-mod
          - NxR vector: word x relation

Context Windows

- Same corpus, different windows
  - BNC
  - Nearest neighbors of “dog”

- 2-word window:
  - Cat, horse, fox, pet, rabbit, pig, animal, mongrel, sheep, pigeon
- 30-word window:
  - Kennel, puppy, pet, terrier, Rottweiler, canine, cat, to bark, Alsatian

Example Lin Relation Vector

	subj-of, absorb	*subj-of*, adapt	*subj-of*, behave	...	*pobj-of*, inside	*pobj-of*, into	...	*nmod-of*, abnormality	*nmod-of*, anemia	*nmod-of*, architecture	...	*obj-of*, attack	*obj-of*, call	*obj-of*, come from	*obj-of*, decorate	...	*nmod*, bacteria	*nmod*, body	*nmod*, bone marrow
cell	1	1	1		16	30		3	8	1		6	11	3	2		3	2	2

Weighting Features

- Baseline: Binary (0/1)
  - Minimally informative
  - Can't capture intuition that frequent features informative
- Frequency or Probability:

$$P(f \mid w) = \frac{count(f,w)}{count(w)}$$

  - Better but,
  - Can overweight a priori frequent features
    - Chance cooccurrence

Pointwise Mutual Information

$$assoc_{PMI}(w,f) = \log_2 \frac{P(w,f)}{P(w)P(f)}$$

PMI:

- Contrasts observed cooccurrence
- With that expected by chance (if independent)

- Generally only use positive values
  - Negatives inaccurate unless corpus huge
- Can also rescale/smooth context values

$$p_{ij} = \frac{f_{ij}}{\sum_{i=1}^{W}\sum_{j=1}^{C} f_{ij}} \quad p_{i*} = \frac{\sum_{j=1}^{C} f_{ij}}{\sum_{i=1}^{W}\sum_{j=1}^{C} f_{ij}} \quad p_{*j} = \frac{\sum_{i=1}^{W} f_{ij}}{\sum_{i=1}^{W}\sum_{j=1}^{C} f_{ij}}$$

$$PPMI_{ij} = \max(\log_2 \frac{p_{ij}}{p_{i*}p_{*j}}, 0)$$

Vector Similarity

- Euclidean or Manhattan distances:
  - Too sensitive to extreme values
- Dot product: $sim_{dot-product}(\vec{v},\vec{w}) = \vec{v} \bullet \vec{w} = \sum_{i=1}^{N} v_i \times w_i$
  - Favors long vectors:
    - More features or higher values
- Cosine: $sim_{\cos ine}(\vec{v},\vec{w}) = \frac{\sum_{i=1}^{N} v_i \times w_i}{\sqrt{\sum_{i=1}^{N} v_i^2}\sqrt{\sum_{i=1}^{N} w_i^2}}$

Alternative Weighting Schemes

- Models have used alternate weights of computing similarity based on weighted overlap

$$\text{sim}_{\text{cosine}}(\vec{v},\vec{w}) = \frac{\vec{v}\cdot\vec{w}}{|\vec{v}||\vec{w}|} = \frac{\sum_{i=1}^{N} v_i \times w_i}{\sqrt{\sum_{i=1}^{N} v_i^2}\sqrt{\sum_{i=1}^{N} w_i^2}} \tag{20.47}$$

$$\text{sim}_{\text{Jaccard}}(\vec{v},\vec{w}) = \frac{\sum_{i=1}^{N} \min(v_i,w_i)}{\sum_{i=1}^{N} \max(v_i,w_i)} \tag{20.48}$$

$$\text{sim}_{\text{Dice}}(\vec{v},\vec{w}) = \frac{2\times\sum_{i=1}^{N} \min(v_i,w_i)}{\sum_{i=1}^{N}(v_i+w_i)} \tag{20.49}$$

Results

- Based on Lin dependency model
  - Hope (N): optimism, chance, expectation, prospect, dream, desire, fear
  - Hope (V): would like, wish, plan, say, believe, think

  - Brief (N): legal brief, affidavit, filing, petition, document, argument, letter
  - Brief (A): lengthy, hour-long, short, extended, frequent, recent, short-lived, prolonged, week-long

Curse of Dimensionality

- Vector representations:
  - Sparse
  - Very high dimensional:
    - # words in vocabulary
    - # relations x # words, etc
- Google1T5 corpus:
  - 1M x 1M matrix: < 0.05% non-zero values
- Computationally hard to manage
  - Lots of zeroes
  - Can miss underlying relations

Reducing Dimensionality

- Feature selection:
  - Desirable traits:
    - High frequency
    - High variance
- Filtering:
  - Can exclude terms with too few occurrences
  - Can include only top X most frequent terms
  - Chi-squared selection
- Cautions:
  - Feature correlations
  - Joint feature selection complex, expensive

Reducing Dimensionality

- Projection into lower dimensional space:
  - Principal Components Analysis (PCA), Locality Preserving Projections (LPP), Singular Value Decomposition, etc
- Create new lower dimensional space that
  - Preserves distances between data points
    - Keep like with like
  - Approaches differ on exactly what is preserved.

SVD

- Enables creation of reduced dimension model
  - Low rank approximation of original matrix
    - Best-fit at that rank (in least-squares sense)

- Motivation:
  - Original matrix: high dimensional, sparse
    - Similarities missed due to word choice, etc
  - Create new projected space
    - More compact, better captures important variation
  - Landauer et al argue identifies underlying “concepts”
    - Across words with related meanings

Document Context

- All models so far:
  - Term x term (or term x relation)
- Alternatively:
  - Term x document
    - Vectors of occurrences (association) in "document"
      - Document can be:
        - Typically: article, essay, etc
        - Also, utterance, dialog act
- Well-known term x document model:
  - Latent Semantic Analysis (LSA)

LSA Document Contexts

- (Deerwester et al, 1990)
- Titles of scientific articles

Example of text data: Titles of Some Technical Memos

c1:	*Human* machine *interface* for ABC *computer* applications
c2:	A *survey* of *user* opinion of *computer system response time*
c3:	The *EPS user interface* management *system*
c4:	*System* and *human system* engineering testing of *EPS*
c5:	Relation of *user* perceived *response time* to error measurement
m1:	The generation of random, binary, ordered *trees*
m2:	The intersection *graph* of paths in *trees*
m3:	*Graph minors* IV: Widths of *trees* and well-quasi-ordering
m4:	*Graph minors*: A *survey*

Document Context Representation

- Term x document:

	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	1	0	0	1	0	0	0	0	0
interface	1	0	1	0	0	0	0	0	0
computer	1	1	0	0	0	0	0	0	0
user	0	1	1	0	1	0	0	0	0
system	0	1	1	2	0	0	0	0	0
response	0	1	0	0	1	0	0	0	0
time	0	1	0	0	1	0	0	0	0
EPS	0	0	1	1	0	0	0	0	0
survey	0	1	0	0	0	0	0	0	1
trees	0	0	0	0	0	1	1	1	0
graph	0	0	0	0	0	0	1	1	1
minors	0	0	0	0	0	0	0	1	1

Document Context Representation

- Term x document:
  - Corr(human,user) = -0.38; corr(human,minors)=-0.29

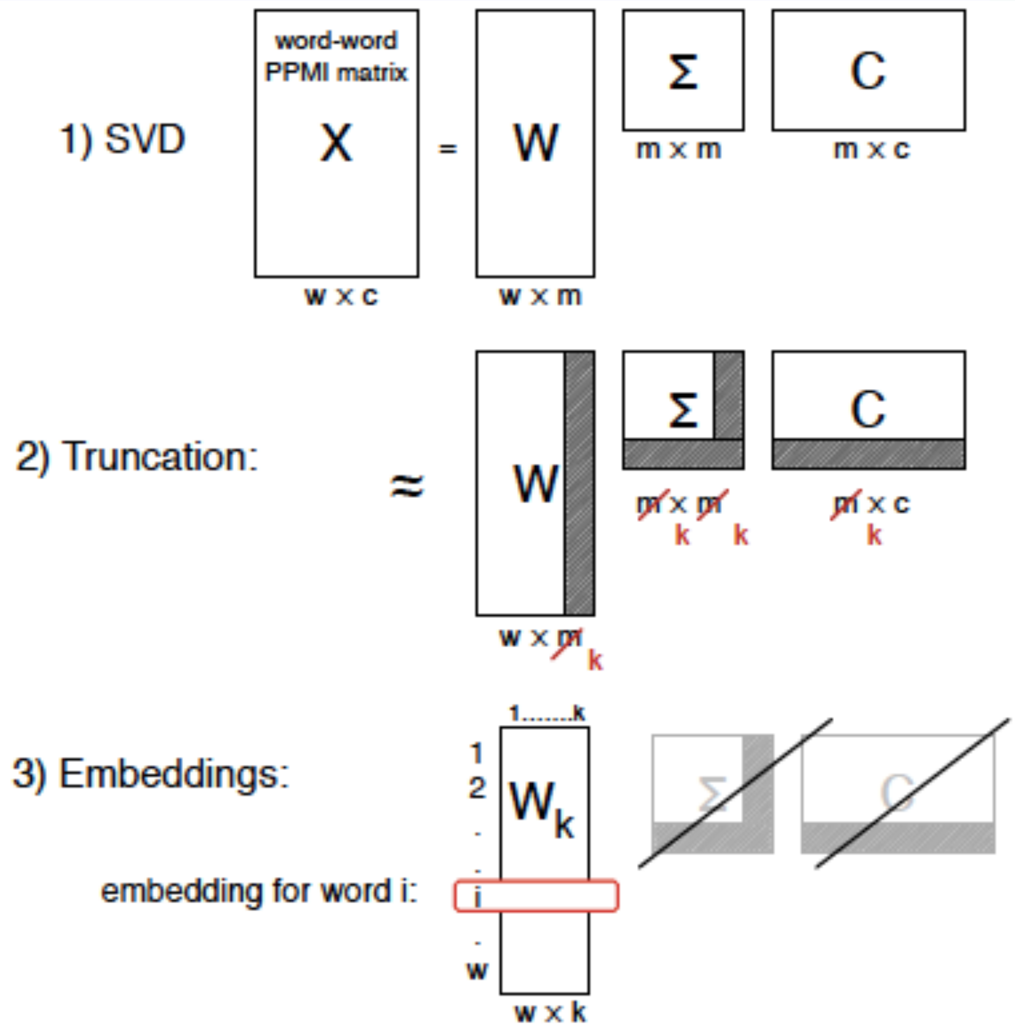
	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	1	0	0	1	0	0	0	0	0
interface	1	0	1	0	0	0	0	0	0
computer	1	1	0	0	0	0	0	0	0
user	0	1	1	0	1	0	0	0	0
system	0	1	1	2	0	0	0	0	0
response	0	1	0	0	1	0	0	0	0
time	0	1	0	0	1	0	0	0	0
EPS	0	0	1	1	0	0	0	0	0
survey	0	1	0	0	0	0	0	0	1
trees	0	0	0	0	0	1	1	1	0
graph	0	0	0	0	0	0	1	1	1
minors	0	0	0	0	0	0	0	1	1

Improved Representation

- Reduced dimension projection:
  - Corr(human,user) = 0.98; corr(human,minors)=-0.83

	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	0.16	0.40	0.38	0.47	0.18	-0.05	-0.12	-0.16	-0.09
interface	0.14	0.37	0.33	0.40	0.16	-0.03	-0.07	-0.10	-0.04
computer	0.15	0.51	0.36	0.41	0.24	0.02	0.06	0.09	0.12
user	0.26	0.84	0.61	0.70	0.39	0.03	0.08	0.12	0.19
system	0.45	1.23	1.05	1.27	0.56	-0.07	-0.15	-0.21	-0.05
response	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
time	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
EPS	0.22	0.55	0.51	0.63	0.24	-0.07	-0.14	-0.20	-0.11
survey	0.10	0.53	0.23	0.21	0.27	0.14	0.31	0.44	0.42
trees	-0.06	0.23	-0.14	-0.27	0.14	0.24	0.55	0.77	0.66
graph	-0.06	0.34	-0.15	-0.30	0.20	0.31	0.69	0.98	0.85
minors	-0.04	0.25	-0.10	-0.21	0.15	0.22	0.50	0.71	0.62

SVD Embedding Sketch



Prediction-based Embeddings

- SVD models: good but expensive to compute
- Skip-gram and Continuous Bag of Words model
  - Popular, efficient implementation in word2vec

- Intuition:
  - Words with similar meanings near each other in text
  - Neural language models learn to predict context words
  - Models train embeddings that make current word
    - More like nearby words and less like distant words
  - Provably related to PPMI models under SVD

Skip-gram Model

- Learns two embeddings
  - W: word, and C: context of some fixed dimension

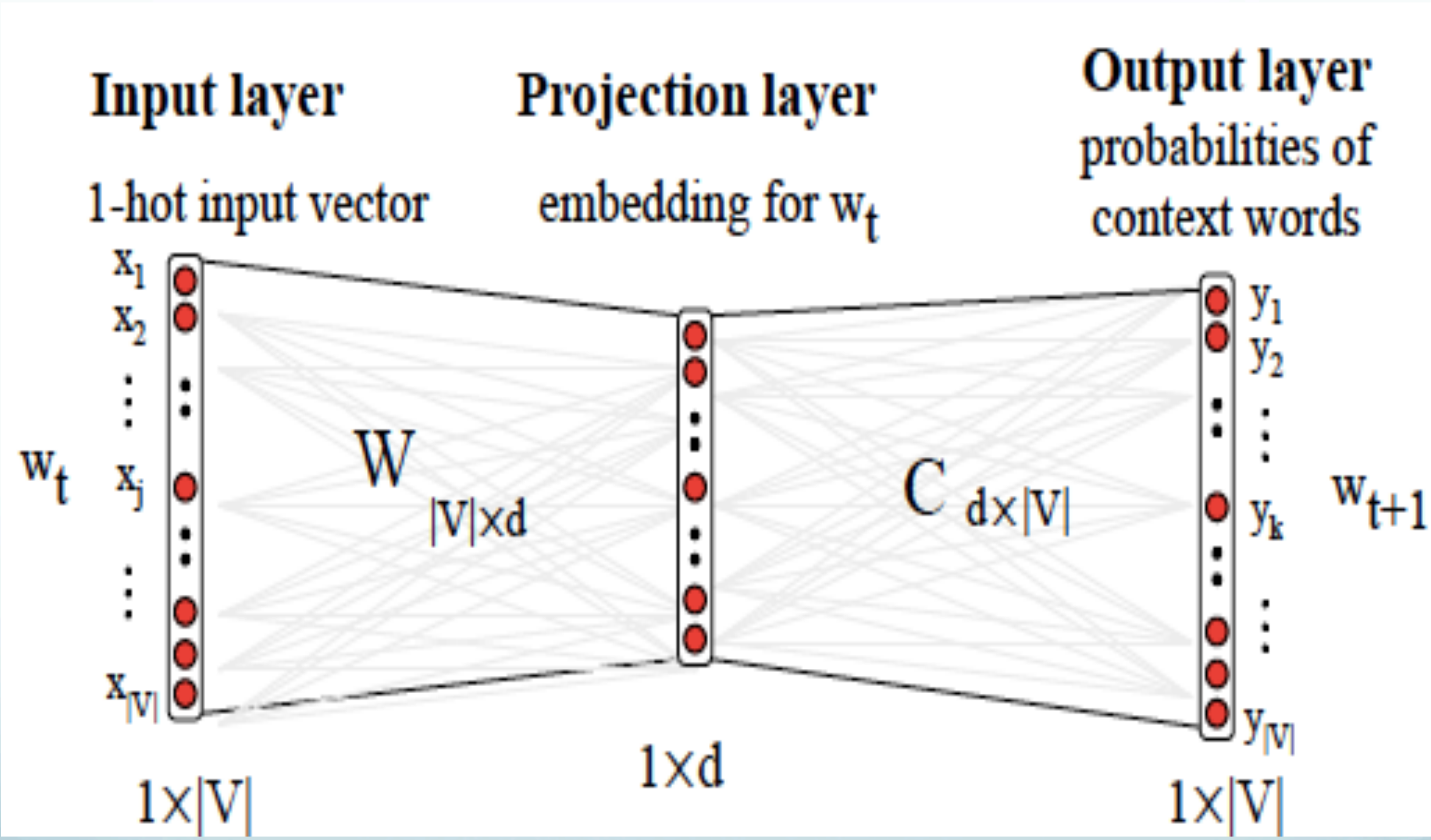
- Prediction task:
  - Given a word, predict each neighbor word in window
  - Compute $p(w_k|w_j)$ represented as $c_k \cdot v_j$
    - For each context position
  - Convert to probability via softmax

$$p(w_k \mid w_j) = \frac{\exp(c_k \bullet v_j)}{\sum_{i \in |V|} \exp(c_i \bullet v_j)}$$

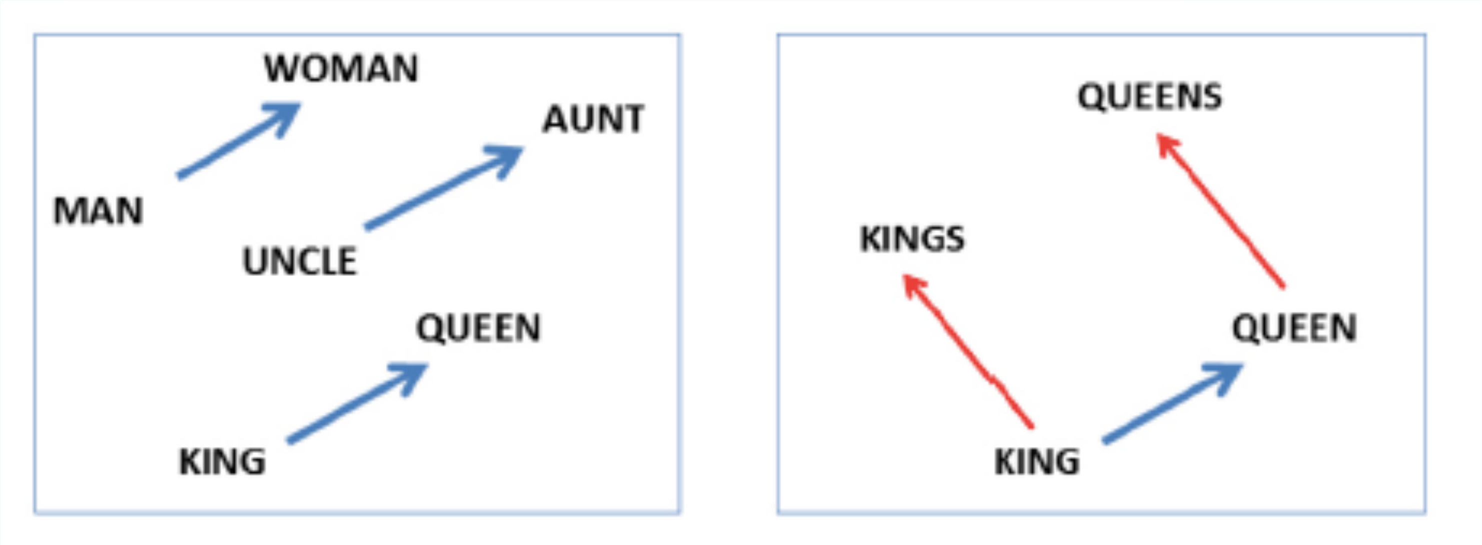
Training the Model

- Issue:
  - Denominator computation is very expensive
- Strategy:
  - Approximate by negative sampling
  - + ex: true context; -- ex: k other words, draw by prob
- Approach:
  - Randomly initialize W, C
  - Iterate over corpus, update w/stoch gradient desc
  - Update embeddings to improve
- Use trained embeddings directly as word rep.

Network Visualization



Relationships via Offsets



Diverse Applications

- Unsupervised POS tagging
- Word Sense Disambiguation
- Essay Scoring
- Document Retrieval
- Unsupervised Thesaurus Induction
- Ontology/Taxonomy Expansion
- Analogy tests, word tests
- Topic Segmentation

Distributional Similarity for Word Sense Disambiguation

Word Space

- Build a co-occurrence matrix
  - Restrict Vocabulary to 4 letter sequences
    - Similar effect to stemming
    - Exclude Very Frequent - Articles, Affixes
  - Entries in 5000-5000 Matrix
    - Apply Singular Value Decomposition (SVD)
    - Reduce to 97 dimensions
- Word Context
  - 4grams within 1001 Characters

Word Representation

- 2nd order representation:

  - Identify words in context of *w*

  - For each x in context of w
    - Compute x's vector representation

  - Compute centroid of those x vector representations

Computing Word Senses

- Compute context vector for each occurrence of word in corpus
- Cluster these context vectors
  - # of clusters = # number of senses
- Cluster centroid represents word sense

- Link to specific sense?
  - Pure unsupervised: no sense tag, just i^{th} sense
  - Some supervision: hand label clusters, or tag training

Disambiguating Instances

- To disambiguate an instance t of w:
  - Compute context vector for the instance
  - Retrieve all senses of w
  - Assign w sense with closest centroid to t

There are more kinds of plants and animals in the rainforests than anywhere else on Earth. Over half of the millions of known species of plants and animals live in the rainforest. Many are found nowhere else. There are even plants and animals in the rainforest that we have not yet discovered.

Biological Example

The Paulus company was founded in 1938. Since those days the product range has been the subject of constant expansions and is brought up continuously to correspond with the state of the art. We're engineering, manufacturing and commissioning world-wide ready-to-run plants packed with our comprehensive know-how. Our Product Range includes pneumatic conveying systems for carbon, carbide, sand, lime and many others. We use reagent injection in molten metal for the…

Industrial Example

Label the First Use of "Plant"

Example Sense Selection for Plant Data

- Build a Context Vector
  - 1,001 character window - Whole Article
- Compare Vector Distances to Sense Clusters
  - Only 3 Content Words in Common
  - Distant Context Vectors
  - Clusters - Build Automatically, Label Manually
- Result: 2 Different, Correct Senses
  - 92% on Pair-wise tasks

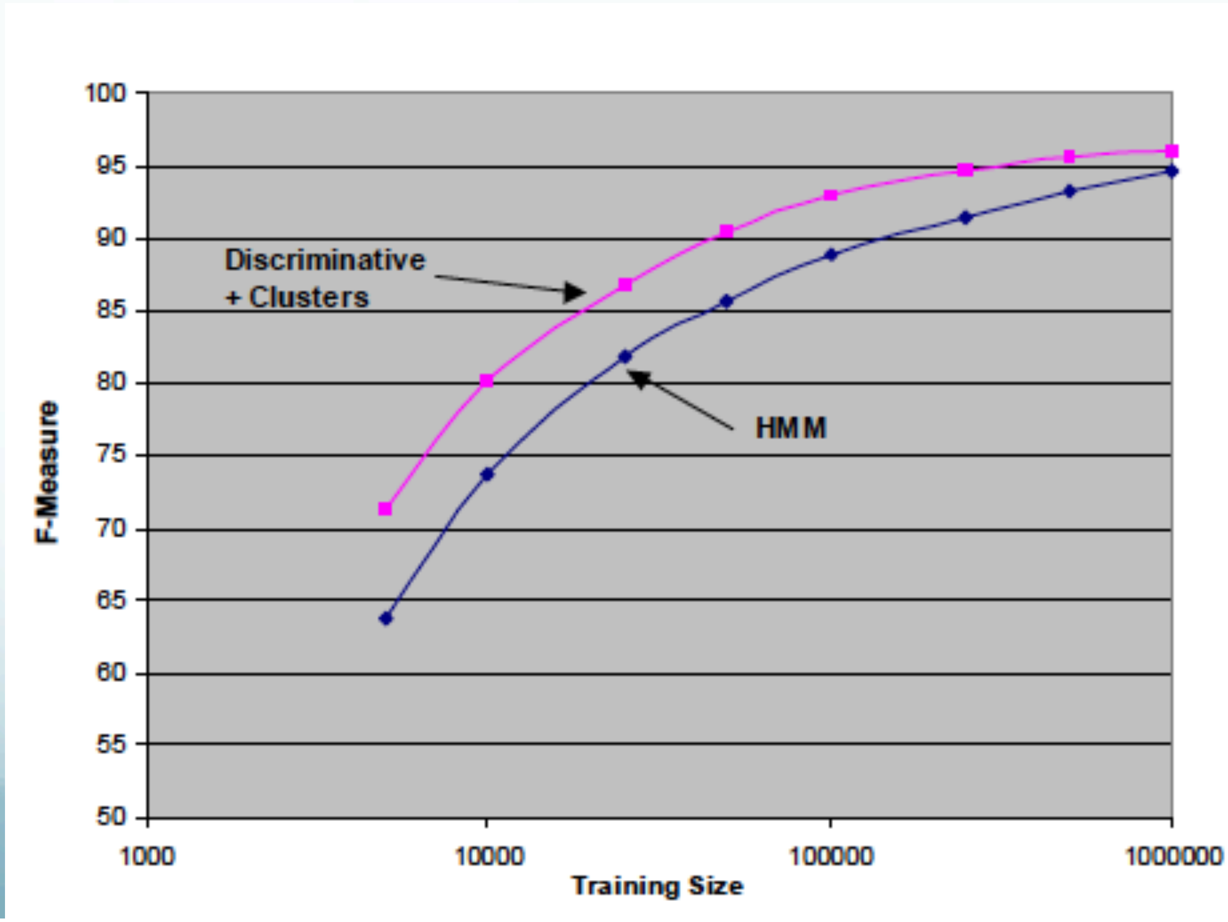
Local Context Clustering

- "Brown" (aka IBM) clustering (1992)
  - Generative model over adjacent words
  - Each w_i has class c_i
  - $\log P(W) = \Sigma_i \log P(w_i|c_i) + \log P(c_i|c_{i-1})$
    - (Familiar??)

  - Greedy clustering
    - Start with each word in own cluster
    - Merge clusters based on log prob of text under model
      - Merge those which maximize P(W)

Clustering Impact

- Improves downstream tasks
  - Here Named Entity Recognition vs HMM (Miller et al '04)



Distributional Models: Summary

- Upsurge in distributional compositional models
  - Embeddings:
    - Discriminatively trained, low dimensional reps
    - E.g. word2vec
      - Skipgrams etc over large corpora
  - Composition:
    - Methods for combining word vector models
      - Capture phrasal, sentential meanings

Resource-based Models

Dictionary-Based Approach

- (Simplified) Lesk algorithm
  - “How to tell a pine cone from an ice cream cone”
- Compute ‘signature’ of word senses:
  - Words in gloss and examples in dictionary
- Compute context of word to disambiguate
  - Words in surrounding sentence(s)
- Compare overlap b/t signature and context
- Select sense with highest (non-stopword) overlap

Applying Lesk

- *The bank can guarantee deposits will eventually cover future tuition costs because it invests in mortgage securities.*

$bank^1$	Gloss:	a financial institution that accepts deposits and channels the money into lending activities
	Examples:	"he cashed a check at the bank", "that bank holds the mortgage on my home"
$bank^2$	Gloss:	sloping land (especially the slope beside a body of water)
	Examples:	"they pulled the canoe up on the bank", "he sat on the bank of the river and watched the currents"

- $Bank^1$: 2
- $Bank^2$: 0

Improving Lesk

- Overlap score:
  - All words equally weighted (excluding stopwords)

- Not all words equally informative
  - Overlap with unusual/specific words – better
  - Overlap with common/non-specific words – less good
- Employ corpus weighting:
  - IDF: inverse document frequency
    - $Idf_i = \log (Ndoc/nd_i)$