CS544: Applications of Latent Semantic Analysis

February 28, 2013

Zornitsa Kozareva
USC/ISI
Marina del Rey, CA
kozareva@isi.edu
www.isi.edu/~kozareva

Noun Compound Disambiguation

- What is a noun compound?
  - construction in NLP consisting of a sequence of two or more nouns which together function syntactically as a noun

(a) (cantilever (swing wing))

(b) ((information retrieval) experiment)

Structure (a) is right branching, while (b) is left branching

Small Exercise

Noun Compound	Branching
Ami Pro document	
volunteer rescue workers	
tourist exchange rates	
cluster analysis procedure	
data base subcommittee	
Windows Control Panel	

Small Exercise

Noun Compound	Branching
((Ami Pro) document)	left
(volunteer (rescue workers))	right
(tourist (exchange rates))	right
((cluster analysis) procedure)	left
((data base) subcommittee)	left
(Windows (Control Panel))	right

How is Bracketing Solved?

- Majority of approaches to compound disambiguation use corpus statistics
- **Adjacency algorithm** which compares *acceptability* of immediately adjacent noun pairs (Lauer, 1995)
- Given a sequence of three nouns n1 n2 n3
  - if (n2 n3) is a more acceptable constituent than (n1 n2), then build (n1 (n2 n3))
  - else build ((n1 n2) n3)

How to measure “acceptability”?

- Collect statistics on the occurrence frequency of structurally unambiguous two-noun compounds to inform the analysis of the ambiguous compound.

- For example, given the compound “**computer data bases**”, the structure **(computer (data bases))** would be preferred if **(data bases)** occurred more frequently than **(computer data)** in the corpus

How is Bracketing Solved?

- **Dependency algorithm** (Lauer, 1995) which operates as given a sequence of three nouns n1 n2 n3
  - if (n1 n3) is more acceptable than (n1 n2), then build (n1 (n2 n3));
  - else build ((n1 n2) n3)

Results on Corpus Statistics

- Lauer tested both the dependency and adjacency algorithms on 244 three-noun compounds
- Dependency algorithm outperformed the adjacency algorithm, achieving a maximum of 81% accuracy
- Estimating distribution of concepts rather than that of individual nouns resulted in superior performance, this shows that conceptual association in important for noun compound disambiguation.

Noun Compound Bracketing with LSA?

- Build the word-by-document matrix using four different document collections (corpora)
  - Ami Pro Word Processor for Windows User's a software manual (AmiPro);
  - document abstracts in library science (CISI);
  - document abstracts on aeronautics (CRAN);
  - articles from Time magazine (Time).
- Run SVD
- Calculate "acceptability"

“Acceptability” using LSA

- Calculate the cosine of the angle between each pair of word vectors.
  - Cosine ranges (-1:0 to 1:0); higher cosine indicates stronger association between each word in a pair

- Test adjacency algorithm measuring cosine of (n1 n2) and (n2 n3)

- Test dependency algorithm measuring cosine of (n1 n2) and (n1 n3)

Data Set

- Manually identified and disambiguated three-noun compounds for each one of the four different data collections

Collection Name	AmiPro	CISI	CRAN	Time
Number of Documents	704	1,460	1,400	425
Number of Tokens	138,091	187,696	217,035	252,808
Mean Tokens per Type	46.3	18.7	26.2	11.5
Number of test compounds	307	235	223	214

Evaluation

- What is a sensible baseline to compare against?

Evaluation

- What is a sensible baseline to compare against?
  - left branching is the more common structure (Lauer, 1995; Resnik, 1993)
  - default strategy of always assume a left branching

Evaluation

- What is a sensible baseline to compare against?
  - left branching is the more common structure (Lauer, 1995; Resnik, 1993)
  - default strategy of always assume a left branching

Name	AmiPro	CISI	CRAN	Time
Baseline	58%	63%	74%	48%
Adjacency	84% (280)	73% (800)	75% (700)	62% (370)
Dependency	70% (200)	70% (1100)	75% (600)	62% (240)

Evaluation

- What is a sensible baseline to compare against?
  - left branching is the more common structure (Lauer, 1995; Resnik, 1993)
  - default strategy of always assume a left branching

Name	AmiPro	CISI	CRAN	Time
Baseline	58%	63%	74%	48%
Adjacency	84% (280)	73% (800)	75% (700)	62% (370)
Dependency	70% (200)	70% (1100)	75% (600)	62% (240)

Analysis / Observations

- Substantial differences in the performances of the adjacency and dependency algorithms were only observed for the AmiPro collection
- Suggests that the increase of the dependency algorithm in Lauer's (1995) study was largely corpus-dependent

Evaluation

- What is a sensible baseline to compare against?
  - left branching is the more common structure (Lauer, 1995; Resnik, 1993)
  - default strategy of always assume a left branching

Name	AmiPro	CISI	CRAN	Time
Baseline	58%	63%	74%	48%
Adjacency	84% (280)	73% (800)	75% (700)	62% (370)
Dependency	70% (200)	70% (1100)	75% (600)	62% (240)

Analysis / Observations

- Time collection has more right-branching (62%) than left branching (48%) compounds
- Comparative study on left-branching compounds, suggests that the choice for default branching must be corpus-dependent

Evaluation

- What is a sensible baseline to compare against?
  - left branching is the more common structure (Lauer, 1995; Resnik, 1993)
  - default strategy of always assume a left branching

Name	AmiPro	CISI	CRAN	Time
Baseline	58%	63%	74%	48%
Adjacency	84% (280)	73% (800)	75% (700)	62% (370)
Dependency	70% (200)	70% (1100)	75% (600)	62% (240)

Evaluation

Collection Name	AmiPro	CISI	CRAN	Time
Number of Documents	704	1,460	1,400	425
Number of Tokens	138,091	187,696	217,035	252,808
Mean Tokens per Type	46.3	18.7	26.2	11.5
Number of test compounds	307	235	223	214

Name	AmiPro	CISI	CRAN	Time
Baseline	58%	63%	74%	48%
Adjacency	84% (280)	73% (800)	75% (700)	62% (370)
Dependency	70% (200)	70% (1100)	75% (600)	62% (240)

Analysis / Observations

- Positive relationship between performance and the token-type ratio.
- The number of tokens per type in the AmiPro collection was **46.3**;
- The worst performance was found for the Time collection, which had only **11.5** tokens per type.

Analysis / Observations

- There were more samples of each word type in the AmiPro collection—this may have helped LSI to construct vectors which were more representative of each word's contextual usage, hence better performance

- LSI constructs a single vector for each token—if a particular token is polysemous in text then its vector will be "noisy", which will lead to poor performance.

Automatic Short Answer Grading

- Task Definition: automatically assign a grade to an answer provided by a student through a comparison with one or more correct answers

Automatic Short Answer Grading

Sample questions, correct answers, and student answers	Grade
Question: What is the role of a prototype program in problem solving? *Correct answer: To simulate the behavior of portions of the desired software product.*	
Student answer 1: A prototype program is used in problem solving to collect data for the problem. *Student answer 2: It simulates the behavior of portions of the desired software product.* *Student answer 3: To find problem and errors in a program before it is finalized.*	1, 2 5, 5 2, 2
Question: What are the main advantages associated with object-oriented programming? *Correct answer: Abstraction and reusability.*	
Student answer 1: They make it easier to reuse and adapt previously written code and they separate complex programs into smaller, easier to understand classes. *Student answer 2: Object oriented programming allows programmers to use an object with classes that can be changed and manipulated while not affecting the entire object at once.* *Student answer 3: Reusable components, Extensibility, Maintainability, it reduces large problems into smaller more manageable problems.*	5, 4 1, 1 4, 4

Data Set

- Collect questions from introductory data structures course assignments with answers provided by a class of undergraduate students
- Three assignments, per assignment there are seven short-answer questions, thirty students submitting answers = 630 (3x7x30)
- Two TAs marked the answers from 0 (completely incorrect) to 5 (perfect answer)
- Agreement of two TAs ~.64

Short Answer Grading with LSA?

- Build the word-by-document matrix using different document collections (corpora)
  - BNC **(LSA BNC)**
  - the entire English Wikipedia **(LSA Wikipedia)**
  - a subset of consisting of articles that contain any of the words: computer, computing, computation, algorithm, recursive, or recursion **(LSA Wikipedia CS)**
  - lecture notes associated with the class textbook, specifically covering topics that are used as questions in the sample **(LSA slides)**

Why so Many Different Corpora?

- To capture the role of **domain** and **size** of the document collections

Results

Measure - Corpus	Size	Correlation
Training on generic corpora		
LSA BNC	566.7MB	0.4071
LSA Wikipedia	1.8GB	0.4286
LSA Wikipedia (small)	0.3MB	0.3518
ESA Wikipedia	1.8GB	0.4681
Training on domain-specific corpora		
LSA Wikipedia CS	77.1MB	0.4628
LSA slides	0.3MB	0.4146
ESA Wikipedia CS	77.1MB	0.4385

Results

Measure - Corpus	Size	Correlation
Training on generic corpora		
LSA BNC	566.7MB	0.4071
LSA Wikipedia	1.8GB	0.4286
LSA Wikipedia (small)	0.3MB	0.3518
ESA Wikipedia	1.8GB	0.4681
Training on domain-specific corpora		
LSA Wikipedia CS	77.1MB	0.4628
LSA slides	0.3MB	0.4146
ESA Wikipedia CS	77.1MB	0.4385

Observations

- Assuming a corpus of comparable size, we expect a measure trained on a domain-specific corpus to outperform one that relies on a generic one.

- Correlation of in-domain is r=0.4146, which is higher than the correlation of r=0.3518 for open-domain on the same corpus size

Results

Measure - Corpus	Size	Correlation
Training on generic corpora		
LSA BNC	566.7MB	0.4071
LSA Wikipedia	1.8GB	0.4286
LSA Wikipedia (small)	0.3MB	0.3518
ESA Wikipedia	1.8GB	0.4681
Training on domain-specific corpora		
LSA Wikipedia CS	77.1MB	0.4628
LSA slides	0.3MB	0.4146
ESA Wikipedia CS	77.1MB	0.4385

Observations

- The effect of the domain is even more pronounced when one compares the performance obtained with LSA Wikipedia CS (r=0.4628) with the one obtained with the full LSA Wikipedia (r=0.4286)

- The small domain specific corpus performs better than the generic corpus which is 23 times larger and is a superset of the smaller corpus.

- **What does this mean?**

Observations

- The effect of the domain is even more pronounced when we compare the performance obtained with LSA Wikipedia CS (r=0.4628) with the one obtained with the full LSA Wikipedia (r=0.4286)

- The small domain specific corpus performs better than the generic corpus which is 23 times larger and is a superset of the smaller corpus.

- **For LSA the quality of the texts is vastly more important than their quantity**

LSA Fills-in-the-blank

- Task Definition: Select the word that best completes the sentence

  American students _____ 50% of the class.

  (a) comprise

  (b) input

  (c) investigate

  (d) refine

  (e) structure

- By answering fill-in-the-blank questions one can measure the vocabulary knowledge of students (in our case LSA)

LSA for Learning Word Meaning Representations

- Task Definition: Given a test word, the model had to choose the most highly associated answer from a group of four choices
- In LSA setting, measure the cosine between the given word and all possible synonym candidates
- Results show that LSA performed on a par with proficient non-native English speakers

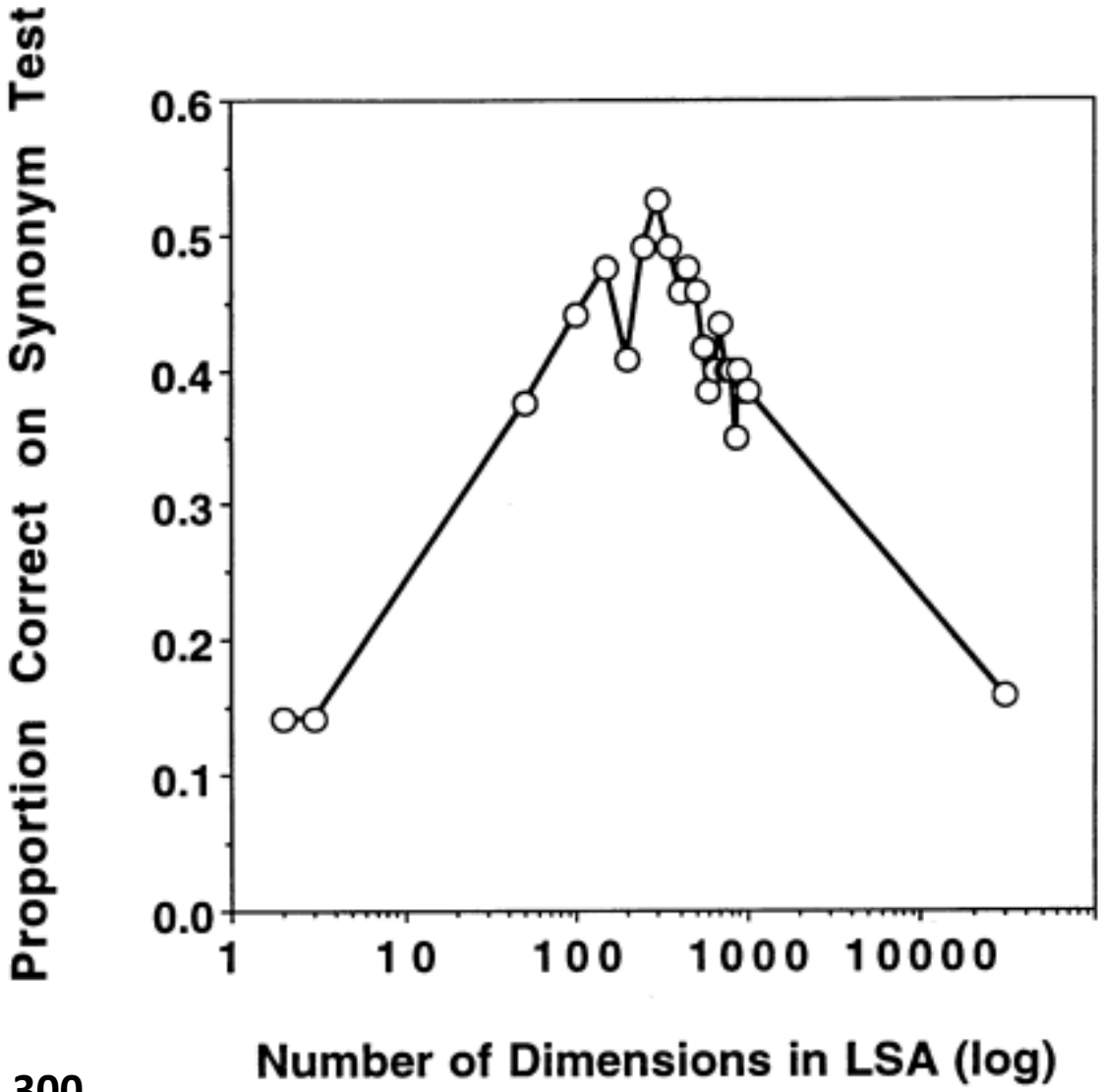
LSA for Synonym Test

- Analyzed an encyclopedia and a sample of triples from a synonymy and antinomy dictionary,
  - "physician"," patient" and "besides" appeared to be well related with a cosine > .5
  - synonyms and antonyms had cosines of about .18 , 12 times larger than between unrelated words from the same set.
- LSA also had a performance similar to that of foreign applicants on TOEFL vocabulary test

LSA for Synonym Test

- Some of LSA's errors were:
  - more sensitive to contextual/paradigmatic associations and less to contrastive semantics it means that LSA prefers "nurse" (cos=.47) to "doctor" (cos=.41) as an associate for "physician"

SVD Dimensions in Synonym Test



52.7% correct with 300
13.5% correct with 2 or 3

Word sorting and relatedness judgment

- Laham and Landauer use LSA to simulate a classic word sorting study ran by Anglin(1970)
  - A word set containing subsets of nouns, verbs, prepositions and adjectives, including common words such as boy, girl, horse, flower was used to point the use of abstract versus concrete similarity relations
  - Children and adults should sort these words by meaning
  - Adults used more abstract categories than children

Information Retrieval

- LSA perfectly matches queries to documents of similar topical meaning
- Represent text as a Matrix of terms by documents
  - Each word and document is represented as a reduced dimensionality vector, usually 50-400 dimensions
  - A Query is is represented as a pseudo document, or weighted average of the vector
- In a test with collection of documents and query results from various applications, LSI performs 30% better than well-known systems

Modeling Human Conceptual Knowledge

- LSA is a good predictor of query-document topic similarity judgment
- LSA is a great simulator of agreed upon word-word relations and human vocabulary test
- LSA simulates human choices of subject-matter multiple choice tests
- LSA is also suitable for text coherence and interpretation, rating of text properties, word =/= word, word =/= passage relations.
- LSA mimics synonymy, antonyms, singular-plural relations

What problems will you solve with LSA?