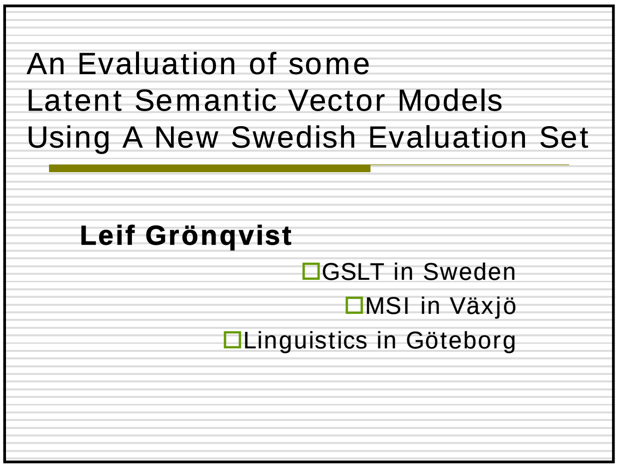


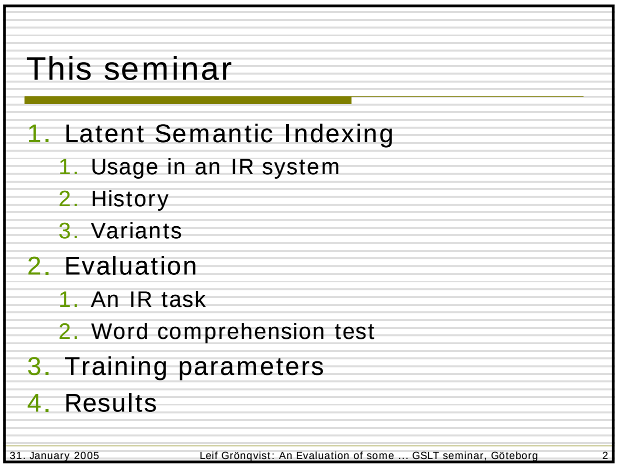
An Evaluation of some Latent Semantic Vector Models Using A New Swedish Evaluation Set

Leif Grönqvist

□GSLT in Sweden

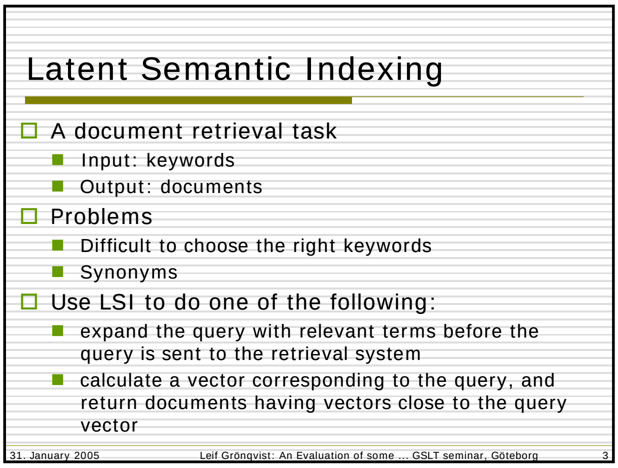
□MSI in Växjö

□Linguistics in Göteborg



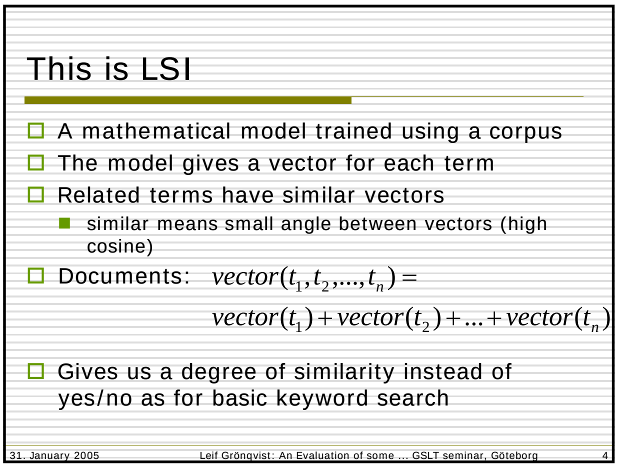
This seminar

1. Latent Semantic Indexing
   1. Usage in an IR system
   2. History
   3. Variants
2. Evaluation
   1. An IR task
   2. Word comprehension test
3. Training parameters
4. Results



Latent Semantic Indexing

- □ A document retrieval task
  - ■ Input: keywords
  - ■ Output: documents
- □ Problems
  - ■ Difficult to choose the right keywords
  - ■ Synonyms
- □ Use LSI to do one of the following:
  - ■ expand the query with relevant terms before the query is sent to the retrieval system
  - ■ calculate a vector corresponding to the query, and return documents having vectors close to the query vector

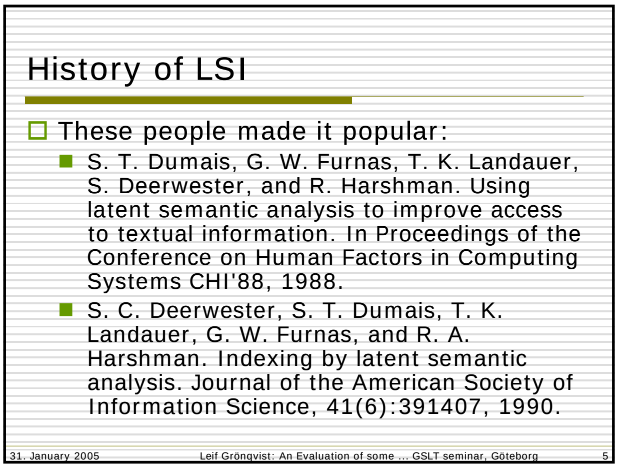


This is LSI

- □ A mathematical model trained using a corpus
- □ The model gives a vector for each term
- □ Related terms have similar vectors
  - ■ similar means small angle between vectors (high cosine)
- □ Documents: $vector(t_1,t_2,...,t_n) =$

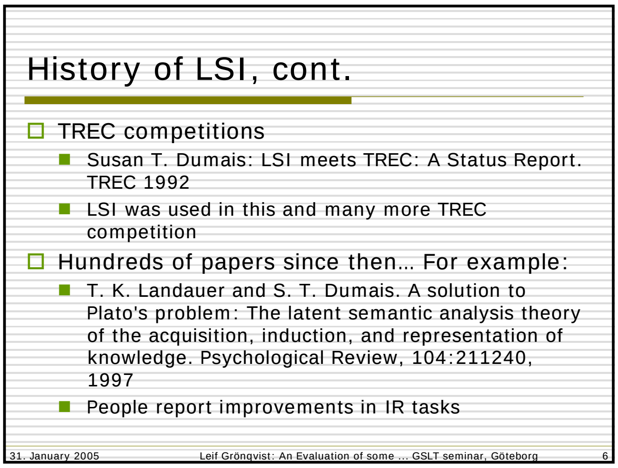
$$vector(t_1)+vector(t_2)+...+vector(t_n)$$

- □ Gives us a degree of similarity instead of yes/no as for basic keyword search



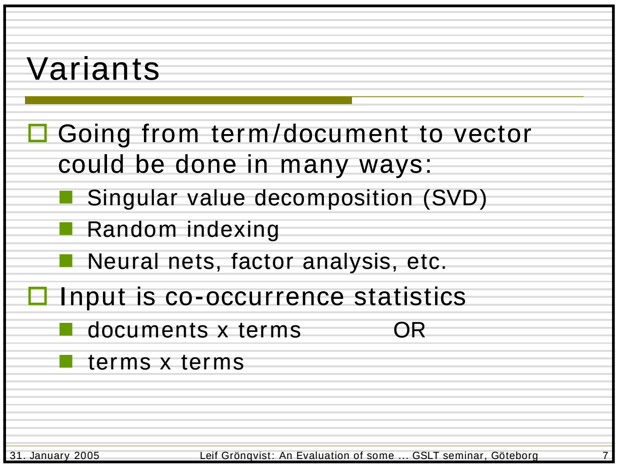
History of LSI

- □ These people made it popular:
  - ■ S. T. Dumais, G. W. Furnas, T. K. Landauer, S. Deerwester, and R. Harshman. Using latent semantic analysis to improve access to textual information. In Proceedings of the Conference on Human Factors in Computing Systems CHI'88, 1988.
  - ■ S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. Journal of the American Society of Information Science, 41(6):391407, 1990.



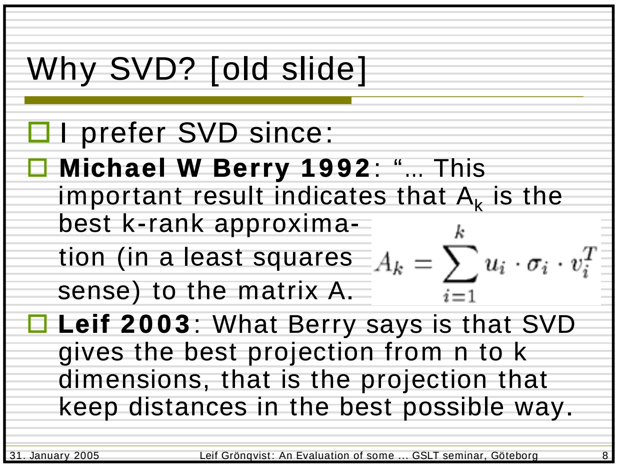
History of LSI, cont.

- □ TREC competitions
  - ■ Susan T. Dumais: LSI meets TREC: A Status Report. TREC 1992
  - ■ LSI was used in this and many more TREC competition
- □ Hundreds of papers since then… For example:
  - ■ T. K. Landauer and S. T. Dumais. A solution to Plato's problem: The latent semantic analysis theory of the acquisition, induction, and representation of knowledge. Psychological Review, 104:211240, 1997
  - ■ People report improvements in IR tasks



Variants

- Going from term/document to vector could be done in many ways:
  - Singular value decomposition (SVD)
  - Random indexing
  - Neural nets, factor analysis, etc.
- Input is co-occurrence statistics
  - documents x terms OR
  - terms x terms

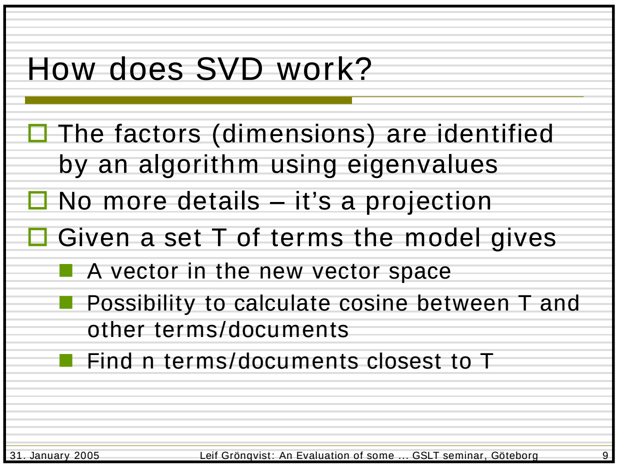


Why SVD? [old slide]

- I prefer SVD since:
- **Michael W Berry 1992**: "... This important result indicates that A_k is the best k-rank approximation (in a least squares sense) to the matrix A.

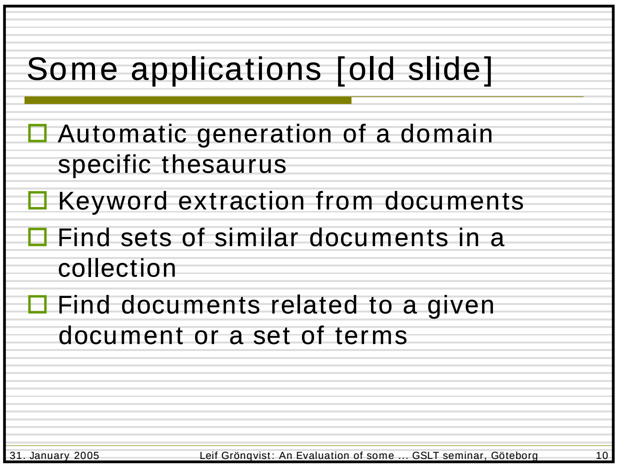
$$A_k = \sum_{i=1}^{k} u_i \cdot \sigma_i \cdot v_i^T$$

- **Leif 2003**: What Berry says is that SVD gives the best projection from n to k dimensions, that is the projection that keep distances in the best possible way.



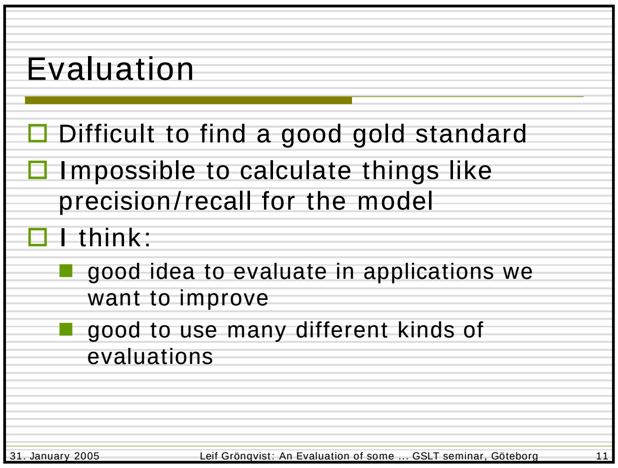
How does SVD work?

- The factors (dimensions) are identified by an algorithm using eigenvalues
- No more details – it's a projection
- Given a set T of terms the model gives
  - A vector in the new vector space
  - Possibility to calculate cosine between T and other terms/documents
  - Find n terms/documents closest to T



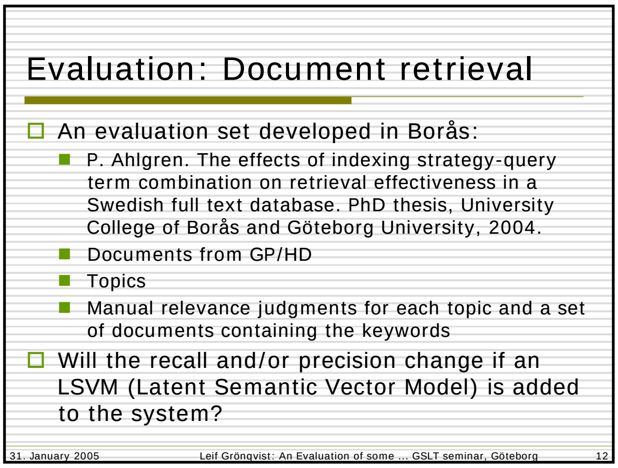
Some applications [old slide]

- Automatic generation of a domain specific thesaurus
- Keyword extraction from documents
- Find sets of similar documents in a collection
- Find documents related to a given document or a set of terms



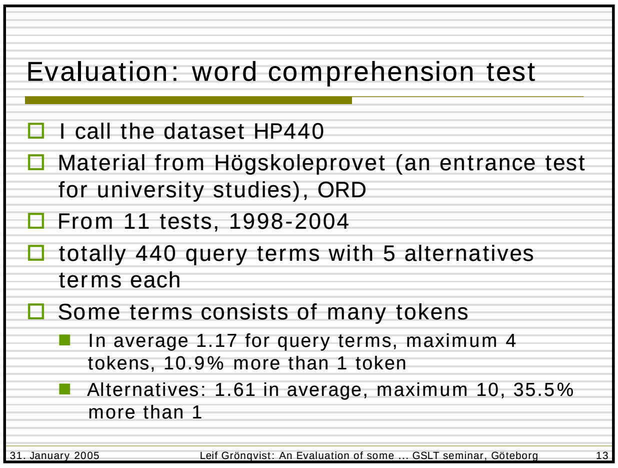
Evaluation

- Difficult to find a good gold standard
- Impossible to calculate things like precision/recall for the model
- I think:
  - good idea to evaluate in applications we want to improve
  - good to use many different kinds of evaluations



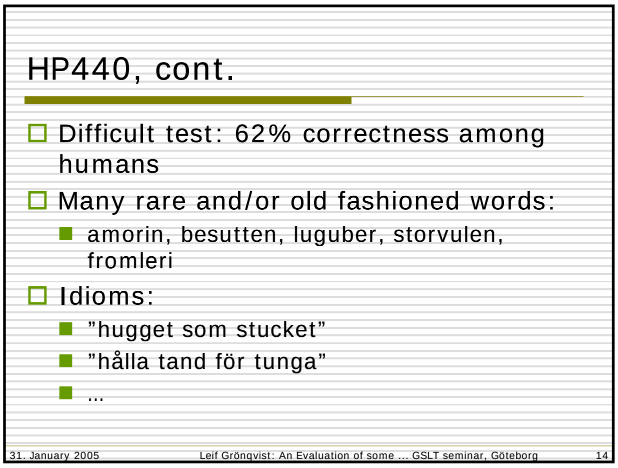
Evaluation: Document retrieval

- An evaluation set developed in Borås:
  - P. Ahlgren. The effects of indexing strategy-query term combination on retrieval effectiveness in a Swedish full text database. PhD thesis, University College of Borås and Göteborg University, 2004.
  - Documents from GP/HD
  - Topics
  - Manual relevance judgments for each topic and a set of documents containing the keywords
- Will the recall and/or precision change if an LSVM (Latent Semantic Vector Model) is added to the system?



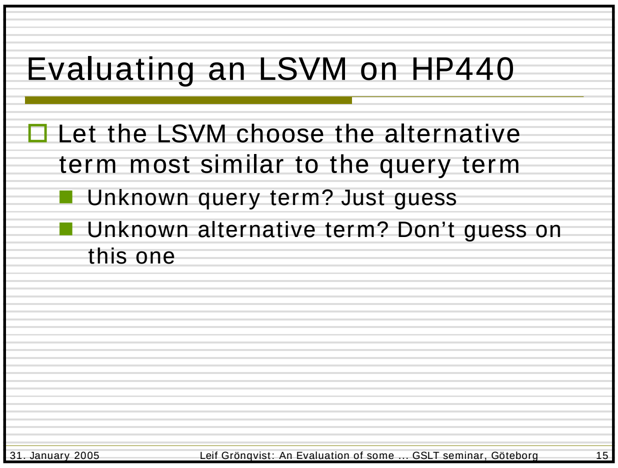
Evaluation: word comprehension test

- I call the dataset HP440
- Material from Högskoleprovet (an entrance test for university studies), ORD
- From 11 tests, 1998-2004
- totally 440 query terms with 5 alternatives terms each
- Some terms consists of many tokens
  - In average 1.17 for query terms, maximum 4 tokens, 10.9% more than 1 token
  - Alternatives: 1.61 in average, maximum 10, 35.5% more than 1



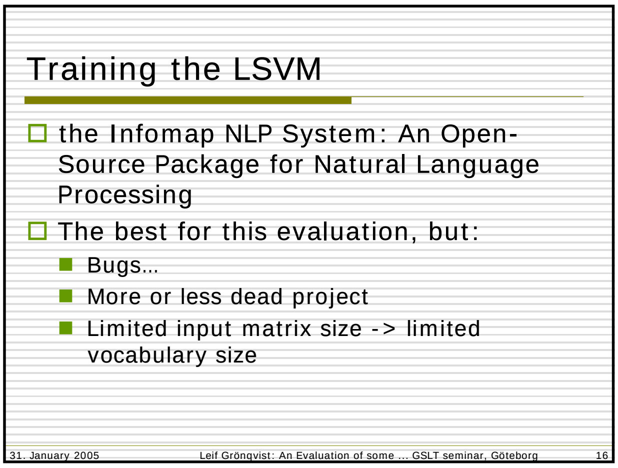
HP440, cont.

- Difficult test: 62% correctness among humans
- Many rare and/or old fashioned words:
  - amorin, besutten, luguber, storvulen, fromleri
- Idioms:
  - ”hugget som stucket”
  - ”hålla tand för tunga”
  - …



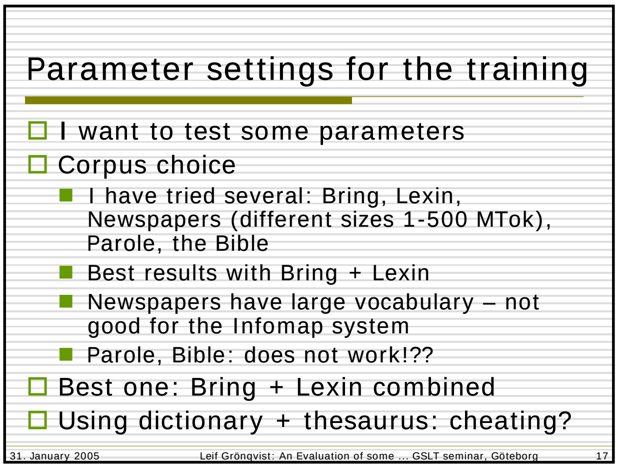
Evaluating an LSVM on HP440

- Let the LSVM choose the alternative term most similar to the query term
  - Unknown query term? Just guess
  - Unknown alternative term? Don't guess on this one



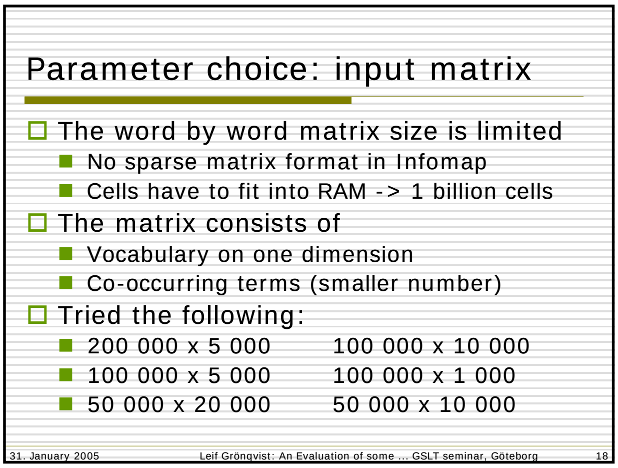
Training the LSVM

- the Infomap NLP System: An Open-Source Package for Natural Language Processing
- The best for this evaluation, but:
  - Bugs…
  - More or less dead project
  - Limited input matrix size -> limited vocabulary size



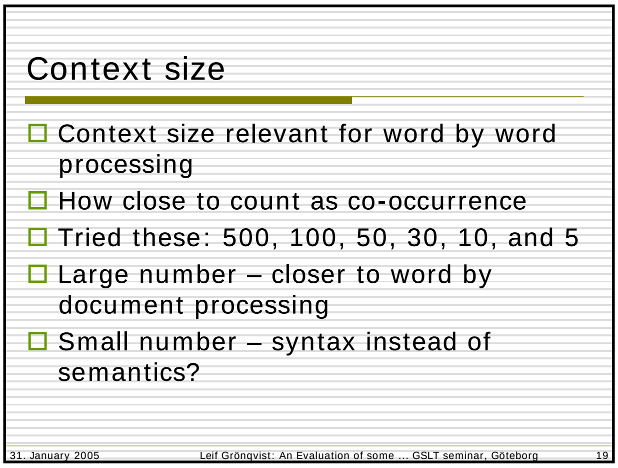
Parameter settings for the training

- I want to test some parameters
- Corpus choice
  - I have tried several: Bring, Lexin, Newspapers (different sizes 1-500 MTok), Parole, the Bible
  - Best results with Bring + Lexin
  - Newspapers have large vocabulary – not good for the Infomap system
  - Parole, Bible: does not work!??
- Best one: Bring + Lexin combined
- Using dictionary + thesaurus: cheating?



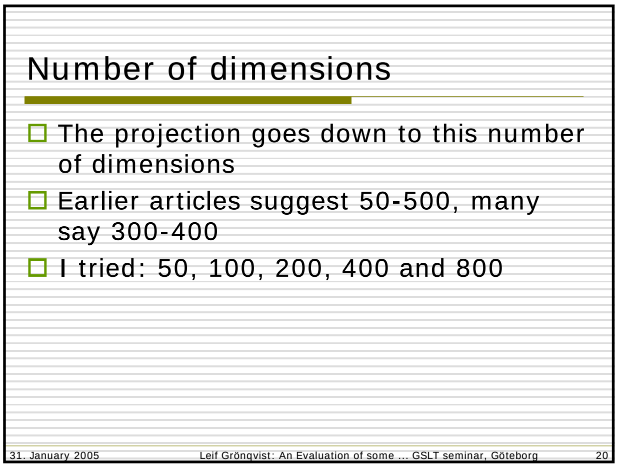
Parameter choice: input matrix

- The word by word matrix size is limited
  - No sparse matrix format in Infomap
  - Cells have to fit into RAM -> 1 billion cells
- The matrix consists of
  - Vocabulary on one dimension
  - Co-occurring terms (smaller number)
- Tried the following:
  - 200 000 x 5 000 100 000 x 10 000
  - 100 000 x 5 000 100 000 x 1 000
  - 50 000 x 20 000 50 000 x 10 000



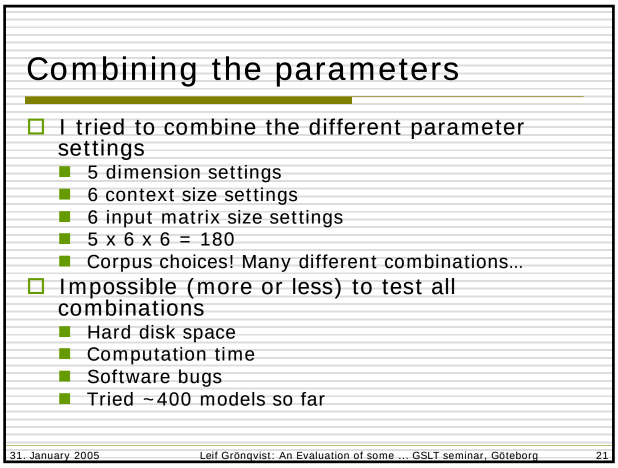
Context size

- Context size relevant for word by word processing
- How close to count as co-occurrence
- Tried these: 500, 100, 50, 30, 10, and 5
- Large number – closer to word by document processing
- Small number – syntax instead of semantics?



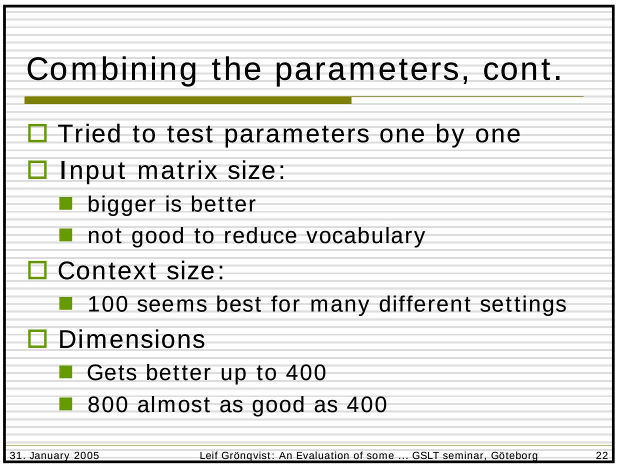
Number of dimensions

- The projection goes down to this number of dimensions
- Earlier articles suggest 50-500, many say 300-400
- I tried: 50, 100, 200, 400 and 800



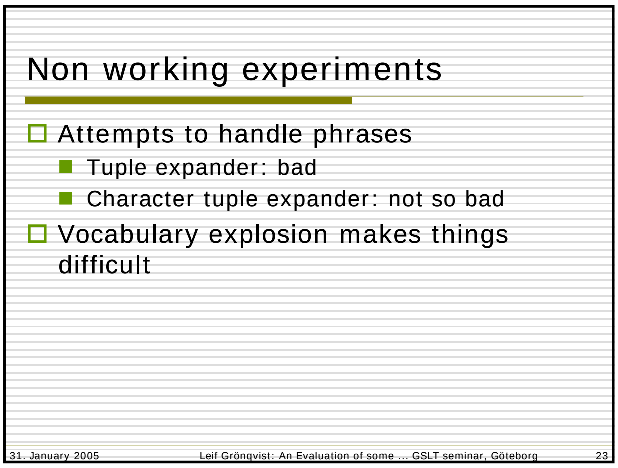
Combining the parameters

- I tried to combine the different parameter settings
  - 5 dimension settings
  - 6 context size settings
  - 6 input matrix size settings
  - 5 x 6 x 6 = 180
  - Corpus choices! Many different combinations…
- Impossible (more or less) to test all combinations
  - Hard disk space
  - Computation time
  - Software bugs
  - Tried ~400 models so far



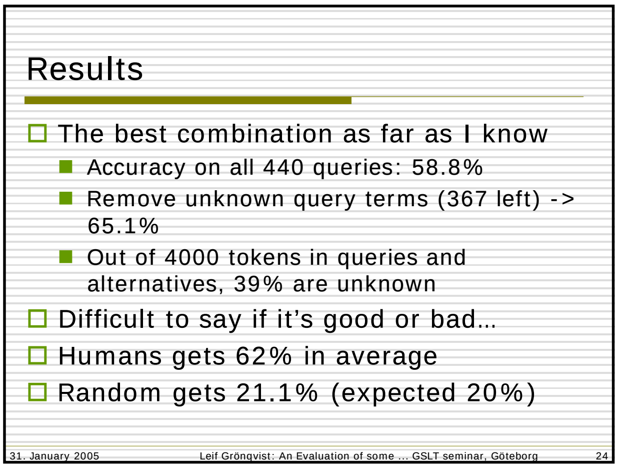
Combining the parameters, cont.

- Tried to test parameters one by one
- Input matrix size:
  - bigger is better
  - not good to reduce vocabulary
- Context size:
  - 100 seems best for many different settings
- Dimensions
  - Gets better up to 400
  - 800 almost as good as 400



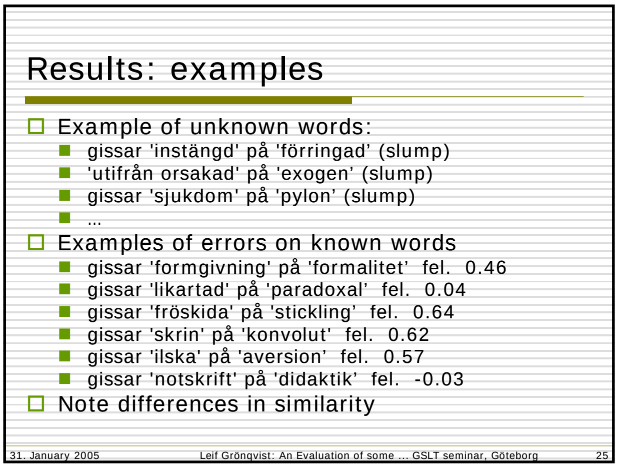
Non working experiments

- Attempts to handle phrases
  - Tuple expander: bad
  - Character tuple expander: not so bad
- Vocabulary explosion makes things difficult



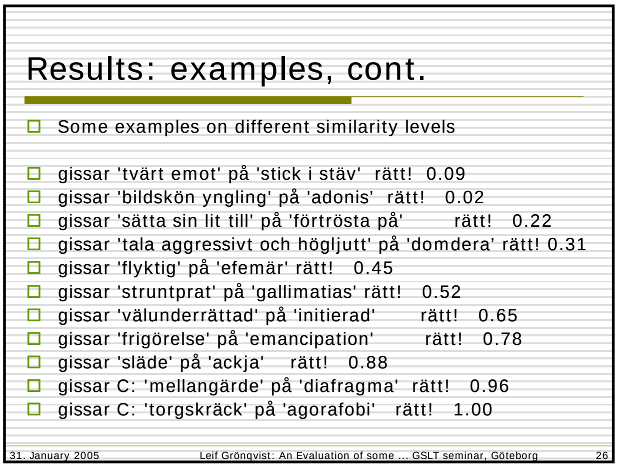
Results

- The best combination as far as I know
  - Accuracy on all 440 queries: 58.8%
  - Remove unknown query terms (367 left) -> 65.1%
  - Out of 4000 tokens in queries and alternatives, 39% are unknown
- Difficult to say if it's good or bad…
- Humans gets 62% in average
- Random gets 21.1% (expected 20%)



Results: examples

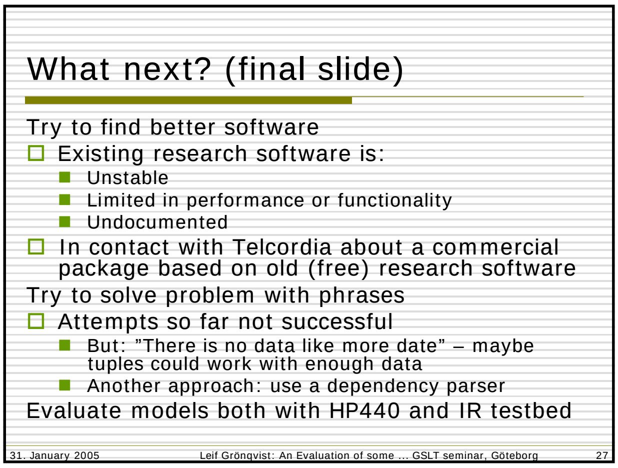
- Example of unknown words:
  - gissar 'instängd' på 'förringad' (slump)
  - 'utifrån orsakad' på 'exogen' (slump)
  - gissar 'sjukdom' på 'pylon' (slump)
  - …
- Examples of errors on known words
  - gissar 'formgivning' på 'formalitet' fel. 0.46
  - gissar 'likartad' på 'paradoxal' fel. 0.04
  - gissar 'fröskida' på 'stickling' fel. 0.64
  - gissar 'skrin' på 'konvolut' fel. 0.62
  - gissar 'ilska' på 'aversion' fel. 0.57
  - gissar 'notskrift' på 'didaktik' fel. -0.03
- Note differences in similarity



Results: examples, cont.

- Some examples on different similarity levels

- gissar 'tvärt emot' på 'stick i stäv' rätt! 0.09
- gissar 'bildskön yngling' på 'adonis' rätt! 0.02
- gissar 'sätta sin lit till' på 'förtrösta på' rätt! 0.22
- gissar 'tala aggressivt och högljutt' på 'domdera' rätt! 0.31
- gissar 'flyktig' på 'efemär' rätt! 0.45
- gissar 'struntprat' på 'gallimatias' rätt! 0.52
- gissar 'välunderrättad' på 'initierad' rätt! 0.65
- gissar 'frigörelse' på 'emancipation' rätt! 0.78
- gissar 'släde' på 'ackja' rätt! 0.88
- gissar C: 'mellangärde' på 'diafragma' rätt! 0.96
- gissar C: 'torgskräck' på 'agorafobi' rätt! 1.00



What next? (final slide)

Try to find better software

- Existing research software is:
  - Unstable
  - Limited in performance or functionality
  - Undocumented
- In contact with Telcordia about a commercial package based on old (free) research software

Try to solve problem with phrases

- Attempts so far not successful
  - But: "There is no data like more date" – maybe tuples could work with enough data
  - Another approach: use a dependency parser

Evaluate models both with HP440 and IR testbed