COMP6237 Data Mining

Semantic Spaces

Jonathon Hare
jsh2@ecs.soton.ac.uk

Introduction

- Distributional Semantics
- Latent Semantic Analysis
- Mining across feature domains
  - Cross-Language LSA
  - Multimodal LSA

Mining Distributional Semantics

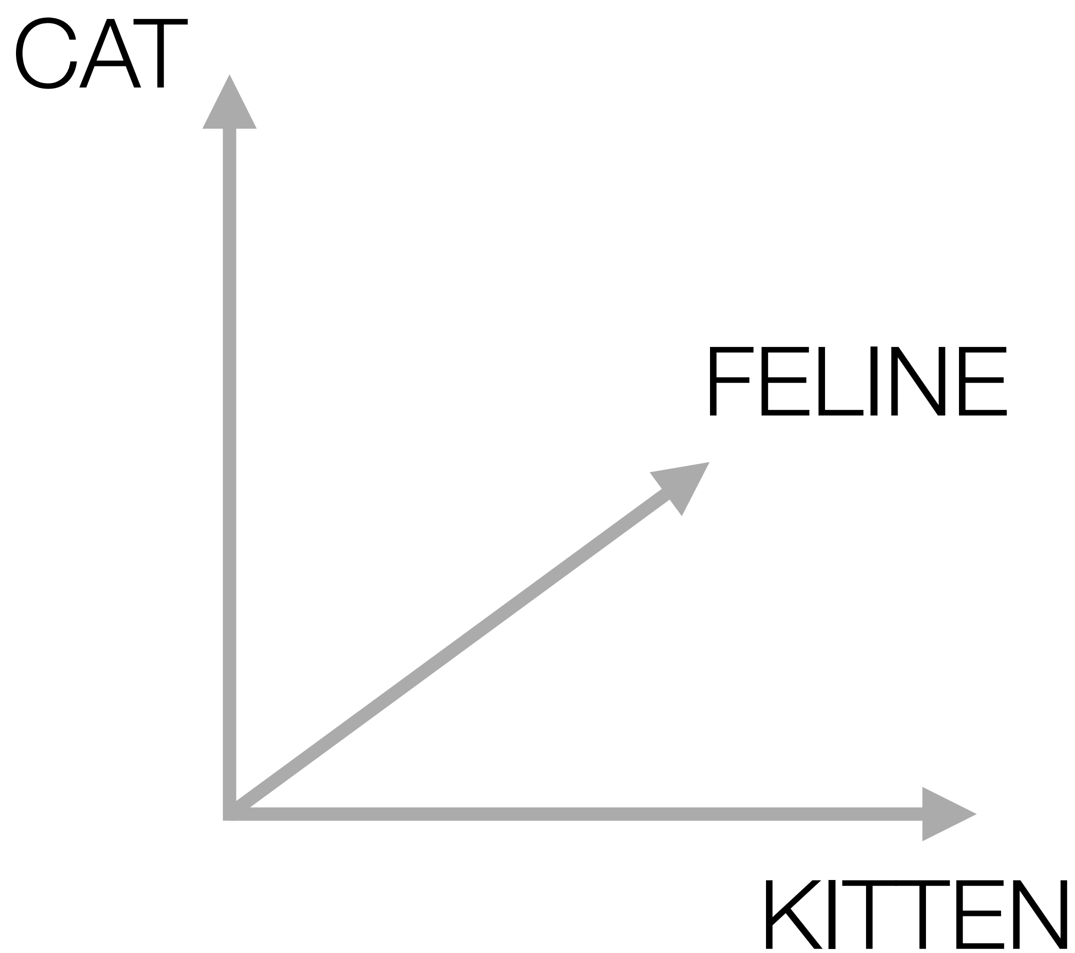
Problem statement I: relating words and texts

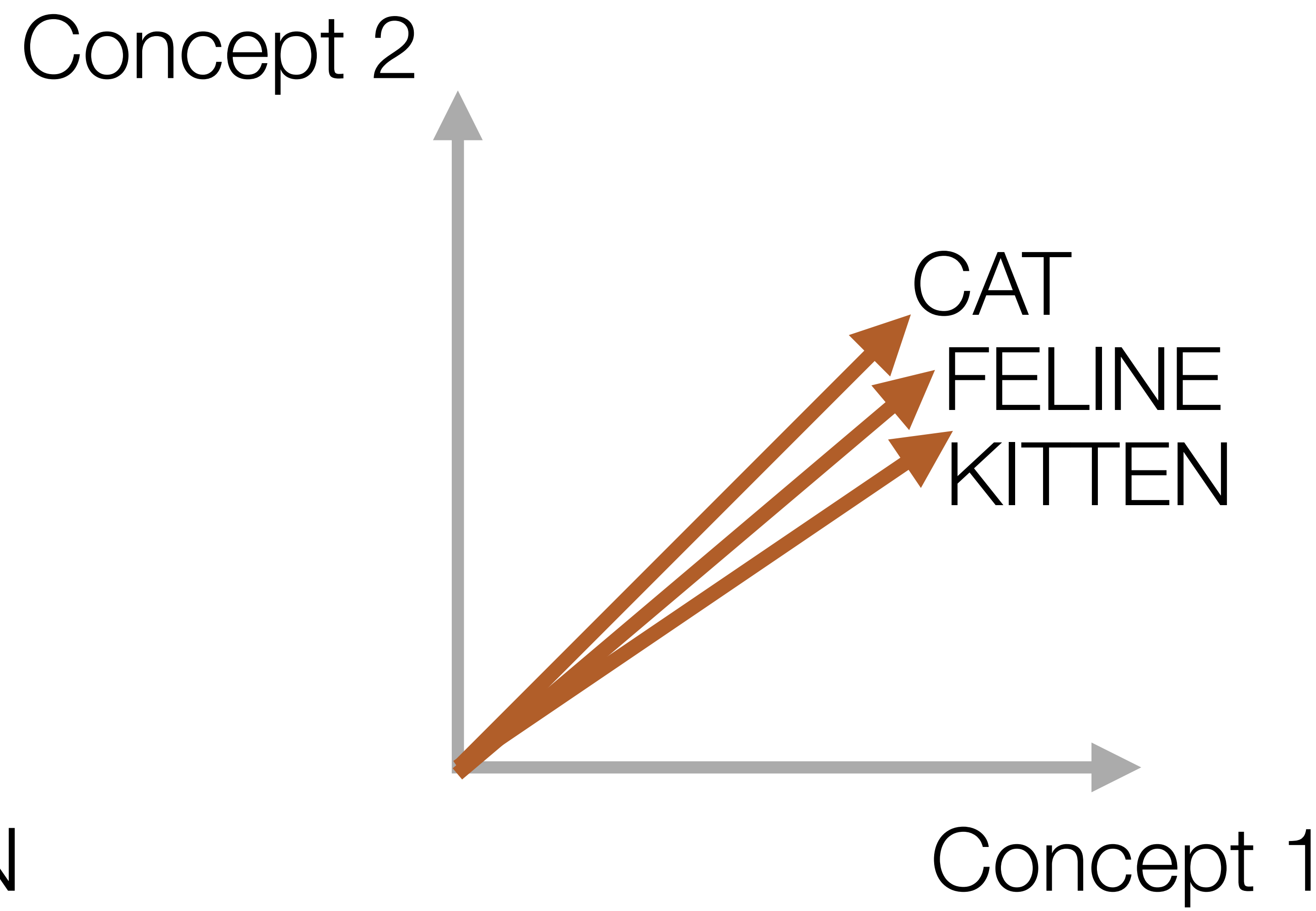
- Thus far when analysing text we've built BOW models that assume independence of terms
- How could we measure the similarity of words?
  - if we know which words are similar could we use this to improve measurements between documents?

Distributional semantics

- **Distributional Hypothesis**:
  - words that are close in meaning will occur in similar pieces of text
- Exploit this to uncover *hidden meaning*
  - **Latent Semantics Analysis**
  - Topic Modelling

Concept Spaces/Semantic Spaces





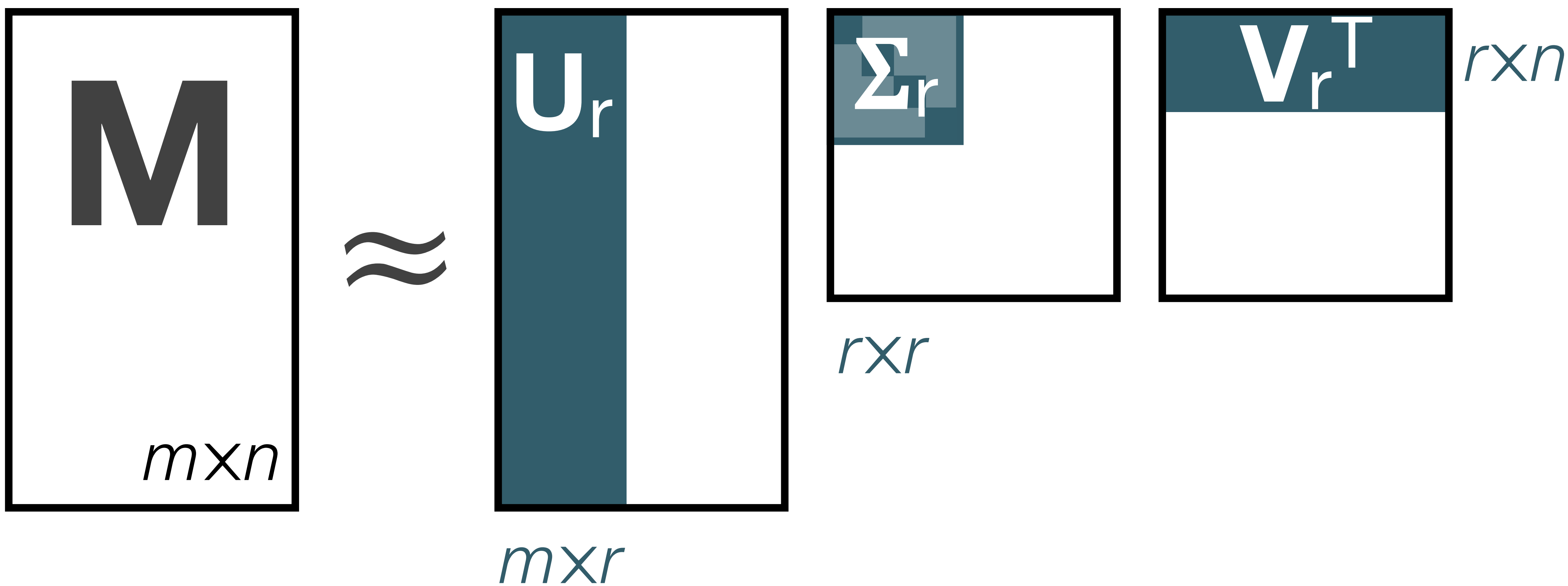
Latent Semantic Analysis

- Consider a **term-document matrix** which described occurrences of terms in documents
  - Clearly going to be sparse
  - Could be weighted (c.f. TF-IDF)

Latent Semantic Analysis

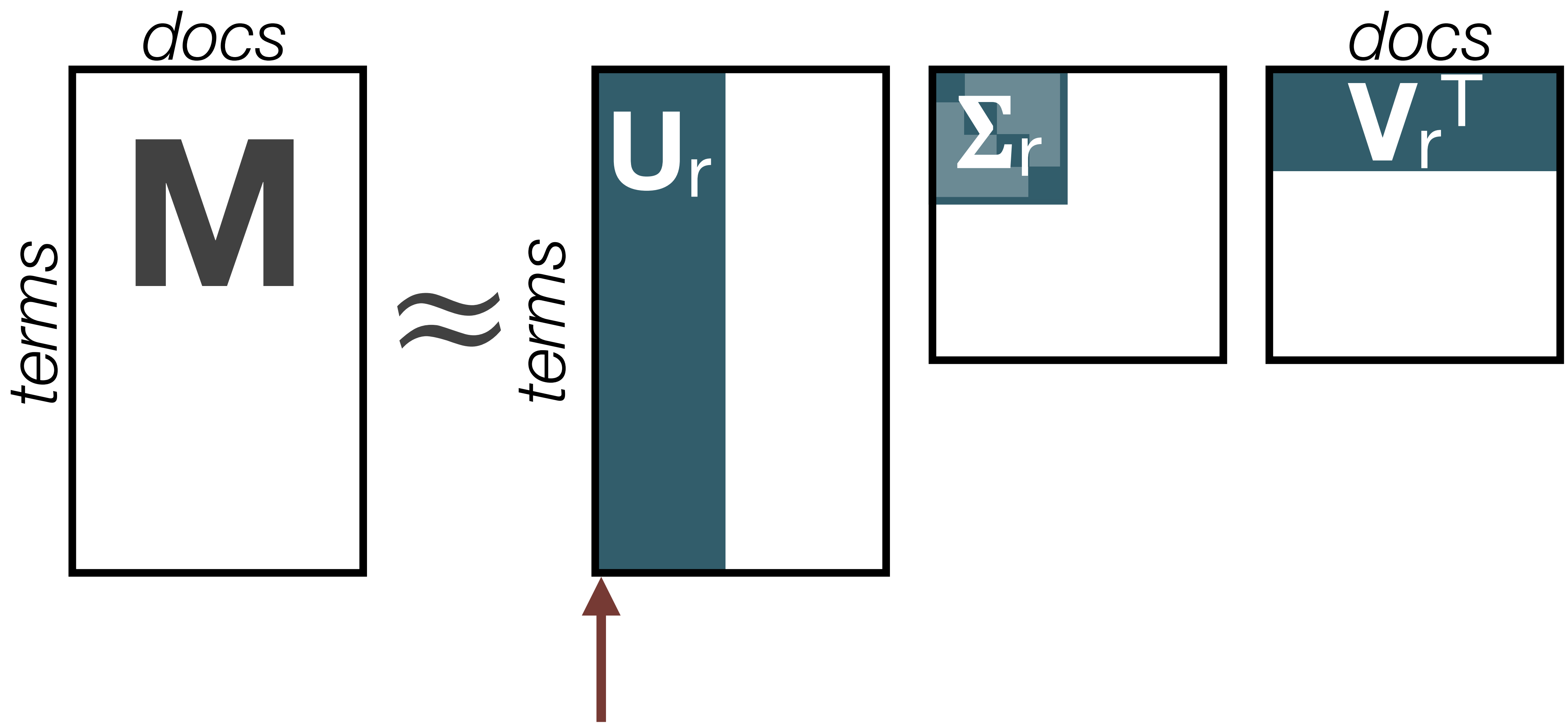
- LSA works by making a **low-rank approximation** under the following assumptions:
  - The original term-document matrix is **noisy**
    - anecdotal instances of terms are to be eliminated.
      - the approximated matrix is **de-noised**
  - The original term-document matrix is **overly sparse** relative to the "*true*" term-document matrix
    - We want to capture **synonymy**

Truncated Singular Value Decomposition Recap



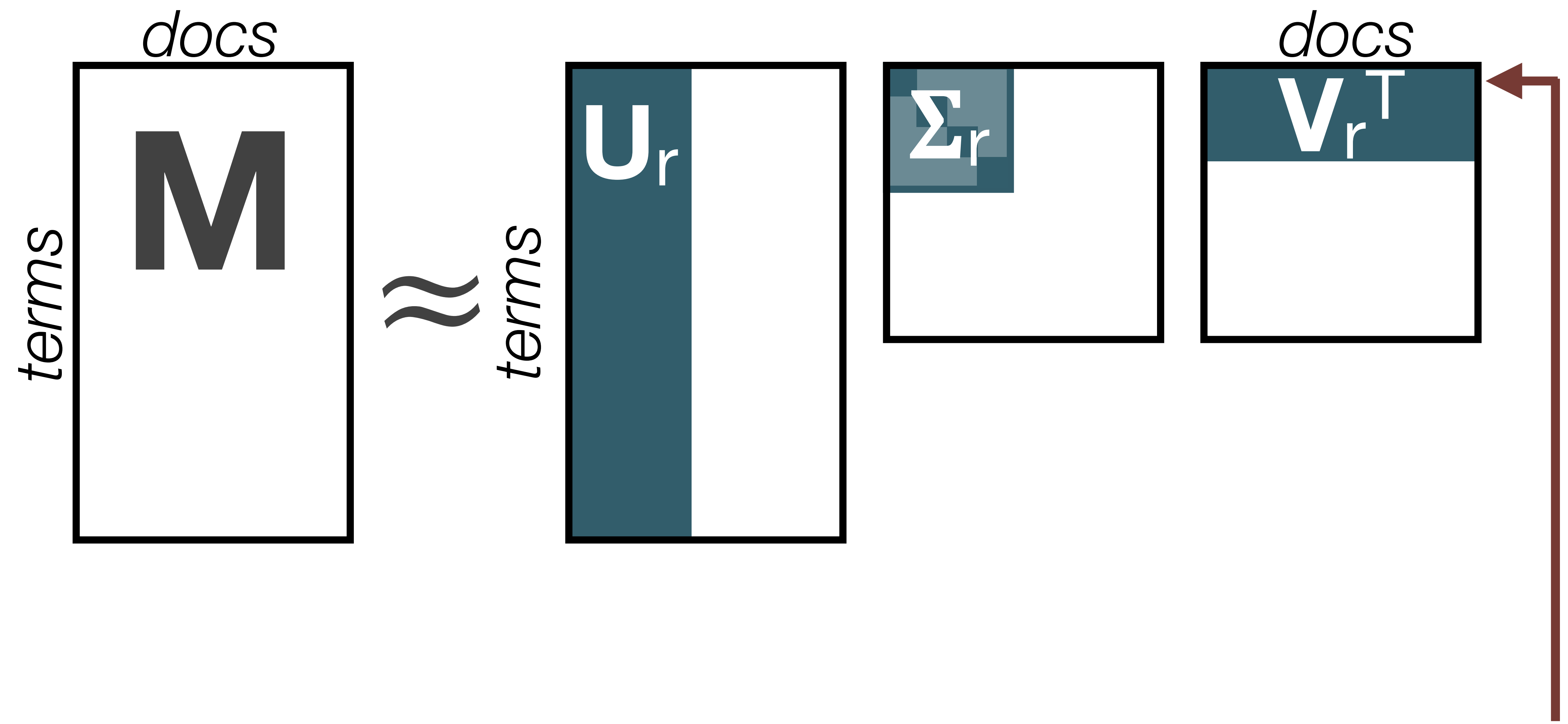
*Truncated SVD considers only the **largest** r singular values (and corresponding left & right vectors)*

Latent Semantic Analysis



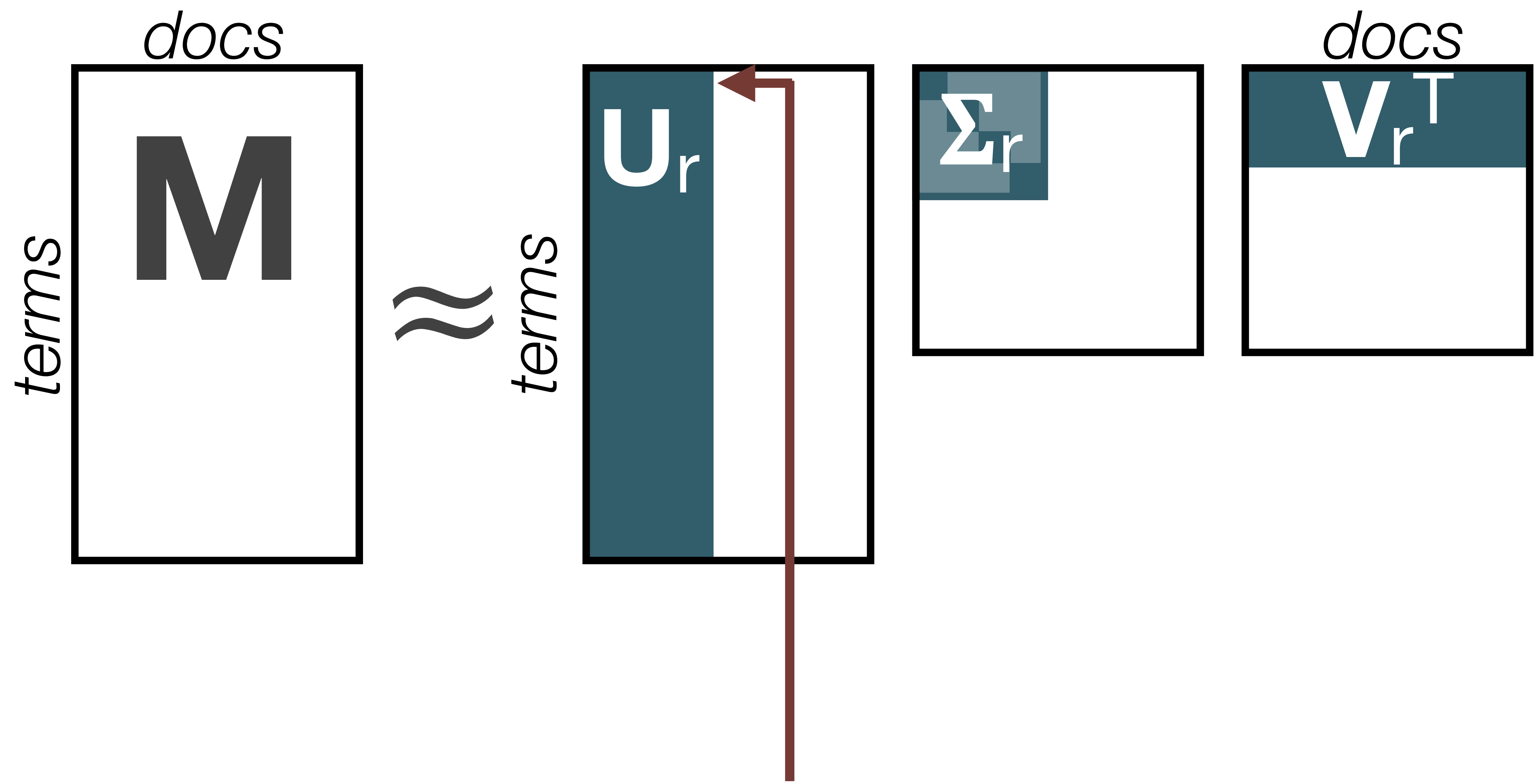
Each column corresponds to an eigenvector of $\mathbf{MM}^{T}$
(i.e. proportional to covariance or correlation of the terms)
*These are the "**concepts**"*

Latent Semantic Analysis



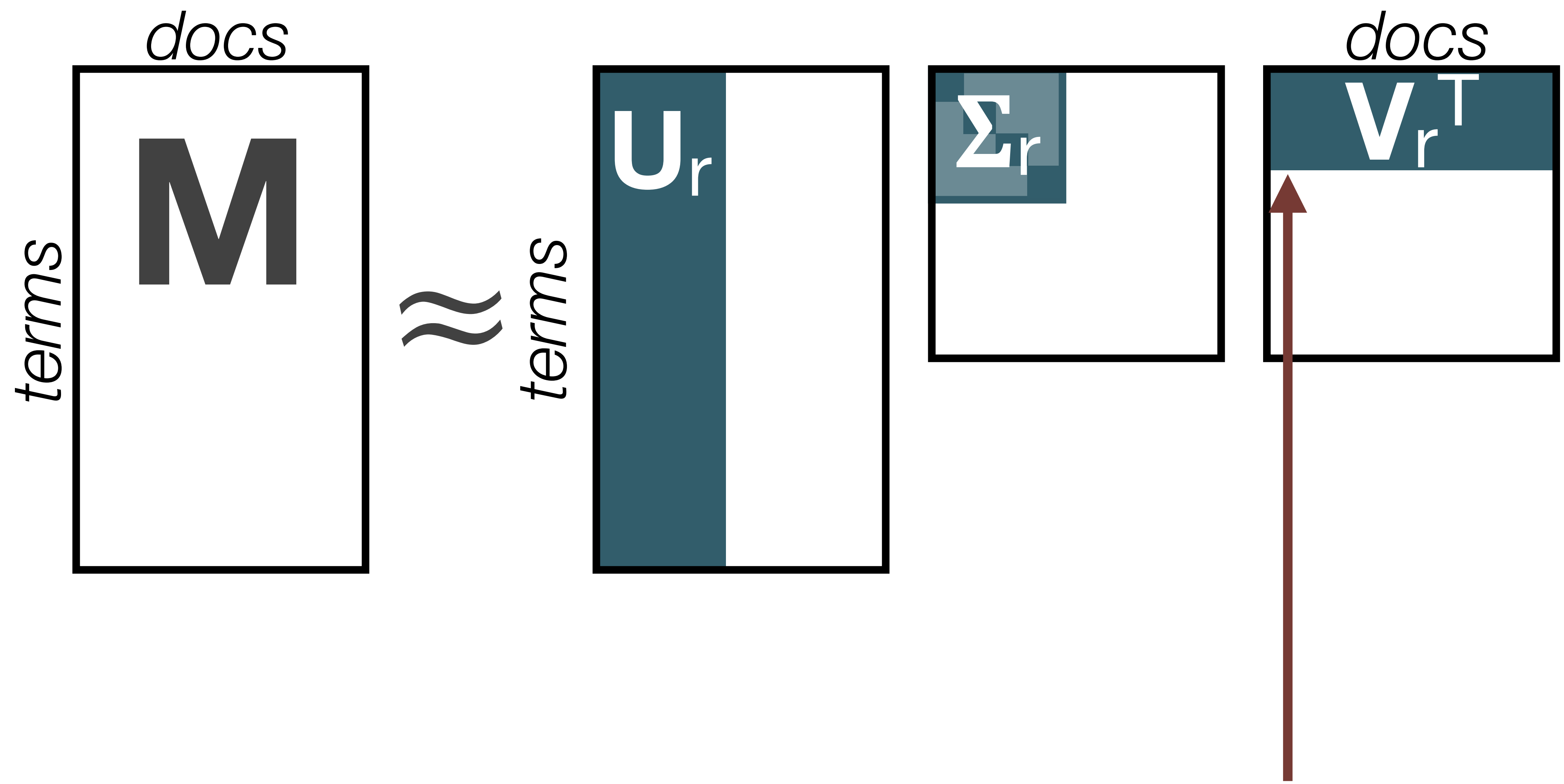
Each row corresponds to an eigenvector of $\mathbf{M}^T\mathbf{M}$
(i.e. proportional to covariance or correlation of the documents)
*These are the "**concepts**"*

Latent Semantic Analysis



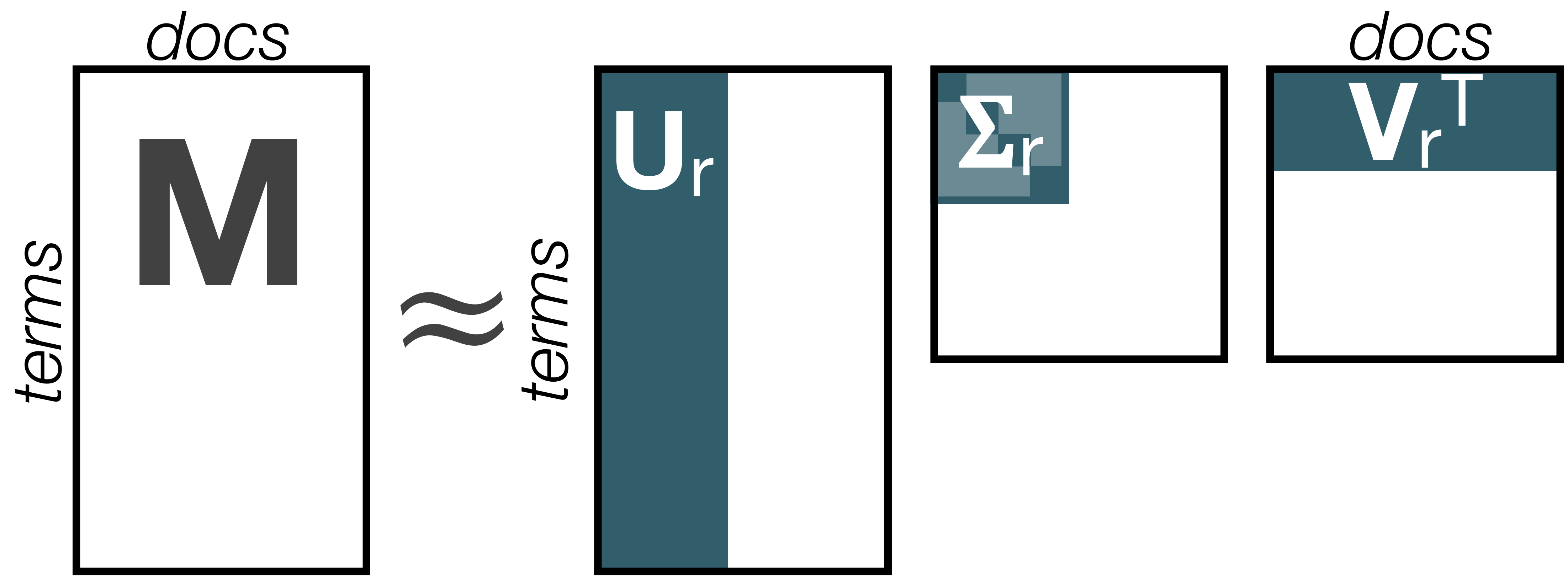
Each row corresponds to an r dimensional vector that describes a term as a vector of weights with respect to the r concepts

Latent Semantic Analysis



Each column corresponds to an r dimensional vector that describes a term as a vector of weights with respect to the r concepts

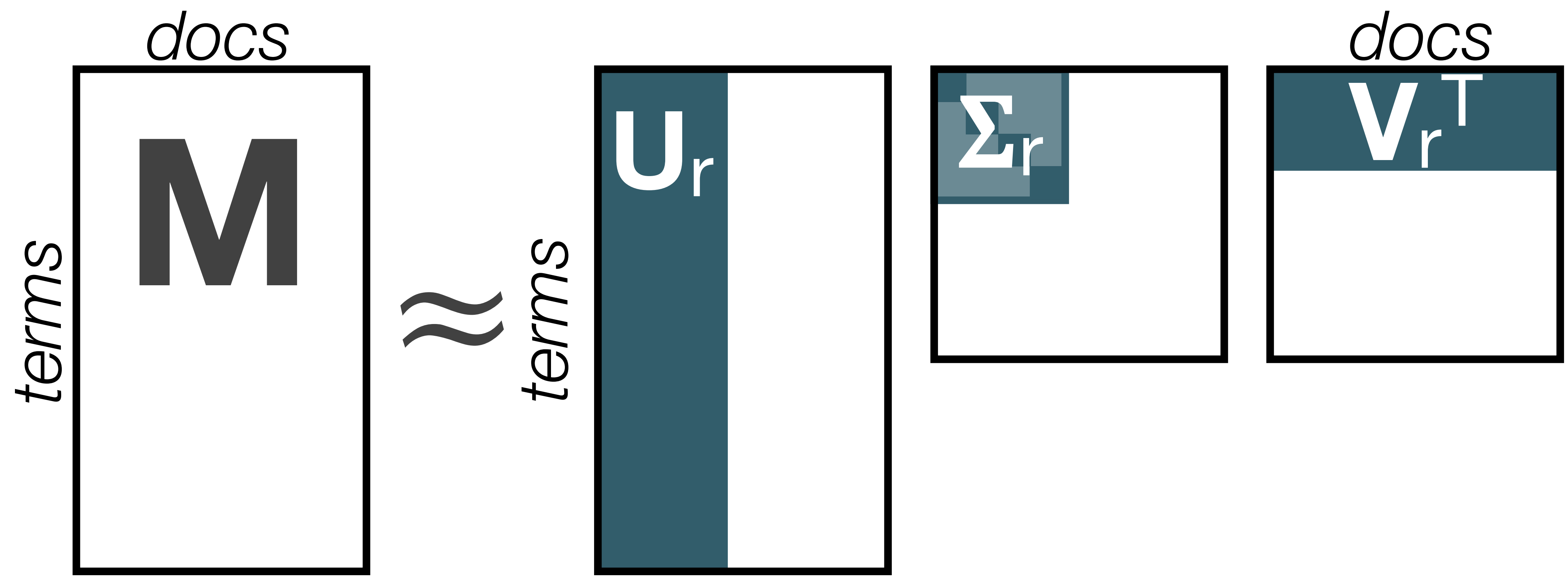
Important Note



The term-concepts and the document-concepts are not the same - they have the same dimensionality, but represent different spaces

They are intrinsically linked though, and it is possible to project one into the other

What exactly is a concept?



A linear combination of terms (or documents).
Not necessarily "comprehensible" -
*e.g. 1.3452 * **car** - 0.2828 * **bottle***

Making comparisons

- See how related documents *j* and *q* are in the low-dimensional space by comparing the vectors $\Sigma_r \mathbf{d}_j$ and $\Sigma_r \mathbf{d}_i$ where $\mathbf{d}_i$ corresponds to the *i*-th column of V_r^T
  - Typically by cosine similarity
- Ditto with terms *i* and *p* by comparing the vectors $\Sigma_r \mathbf{t}_i$ and $\Sigma_r \mathbf{t}_p$ where d_i corresponds to the *i*-th row of U_r.
- Documents and term vector representations can be clustered using traditional clustering algorithms like k-means using similarity measures like cosine.

Latent Semantic Indexing

- Given a query, view this as a mini document, and compare it to your documents in the low-dimensional space.
  - Given a query vector $\mathbf{q}$ with dimensionality equal to the number of terms, project it into the document space: $\mathbf{q}' = \Sigma_r^{-1} U_r^T \mathbf{q}$
  - Then compare $\Sigma_r \mathbf{q}'$ against the low-dimensional document vectors $\Sigma_r \mathbf{d}_j$

Limitations of LSA 1

- **The resulting dimensions might be difficult to interpret.**
  - For instance, in

    `{(car), (truck), (flower)} ↦ {(1.3452 * car + 0.2828 * truck), (flower)}`

    the

    `(1.3452 * car + 0.2828 * truck)`

    component could be interpreted as "vehicle".
  - However, it is very likely that cases close to

    `{(car), (bottle), (flower)} ↦ {(1.3452 * car + 0.2828 * bottle), (flower)}`

    will occur.
  - This leads to results which can be justified on the mathematical level, but have **no interpretable meaning** in natural language.

Limitations of LSA 2

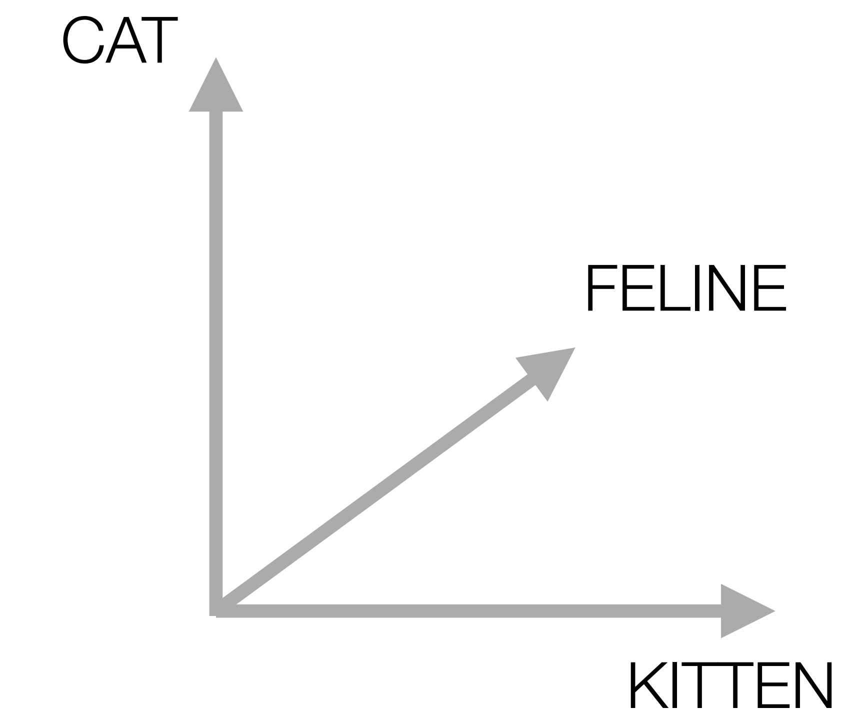
- **Polysemy isn't captured**
  - "The Chair of the Board" versus "the chair maker"
  - The vector representation of chair becomes an average of all the word's different meanings in the corpus
- **Word order is ignored**
  - (n-grams to the rescue?)
- **The probabilistic model of LSA does not match observed data**
  - LSA assumes that words and documents form a joint Gaussian model (ergodic hypothesis), while a Poisson distribution has been observed.

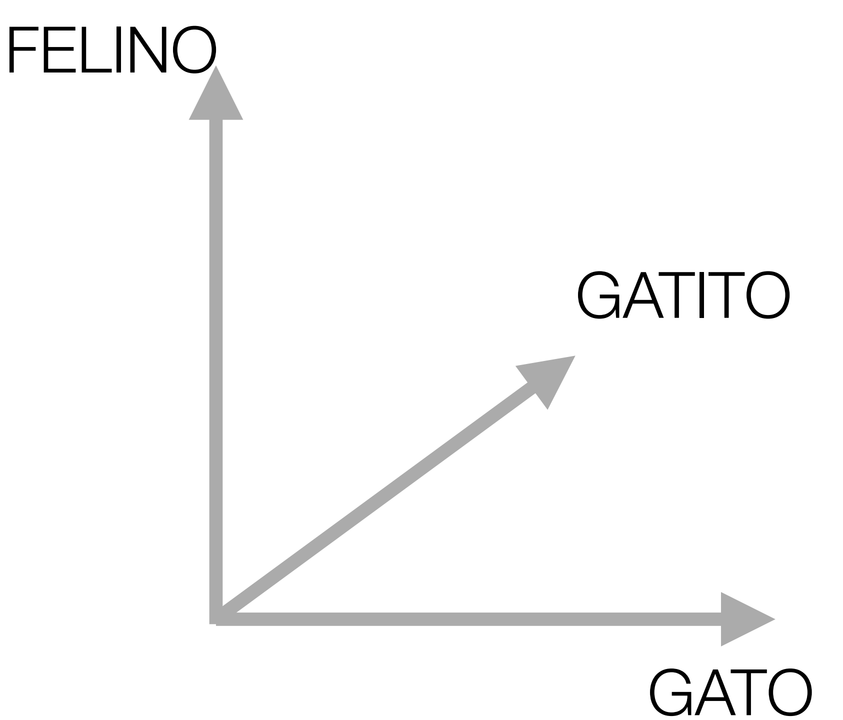
Mining semantic correspondences across feature domains

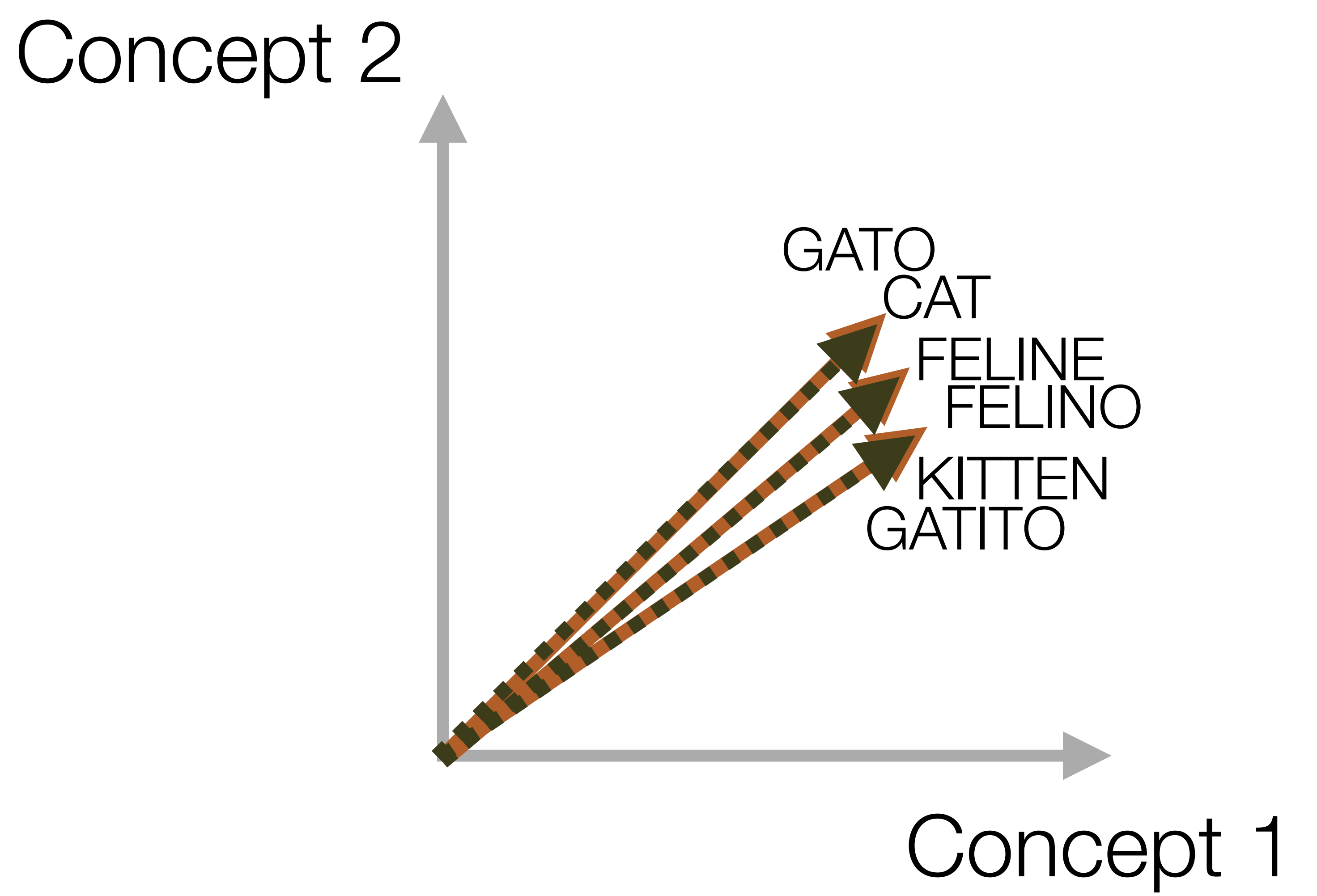
Problem Statement II: cross-lingual search

- Can think of scenarios where we might want to perform a search in one language (e.g. English) and match against documents in another (e.g. French)
  - Could index translated versions of the documents, but
    - automatic translation still has problems
    - manual translation might be expensive
  - What about other approaches?

Embedding across languages







Cross-Language LSI

- Use a bilingual (or multilingual) **training** corpus to build a single term-document matrix
  - each document vector contains terms from the original language and its translation(s)

$$
\begin{bmatrix}
 & \text{CAT} & \text{KITTEN} & \text{FELINE} & \text{FELINO} & \text{GATO} & \text{GATITO} & \ldots \\
\text{doc1} & 1 & 0 & 0 & 0 & 1 & 0 & \ldots \\
\text{doc2} & 1 & 1 & 1 & 1 & 1 & 1 & \ldots \\
\ldots & & & & & & &
\end{bmatrix}
$$

Cross-Language LSI

- Decompose with SVD as per standard LSI
- Perform queries by projecting into the lower dimensional space as before
  - but just use one language and set the rest to 0

	CAT	KITTEN	FELINE	FELINO	GATO	GATITO	…
query	1	0	0	0	1	0	…

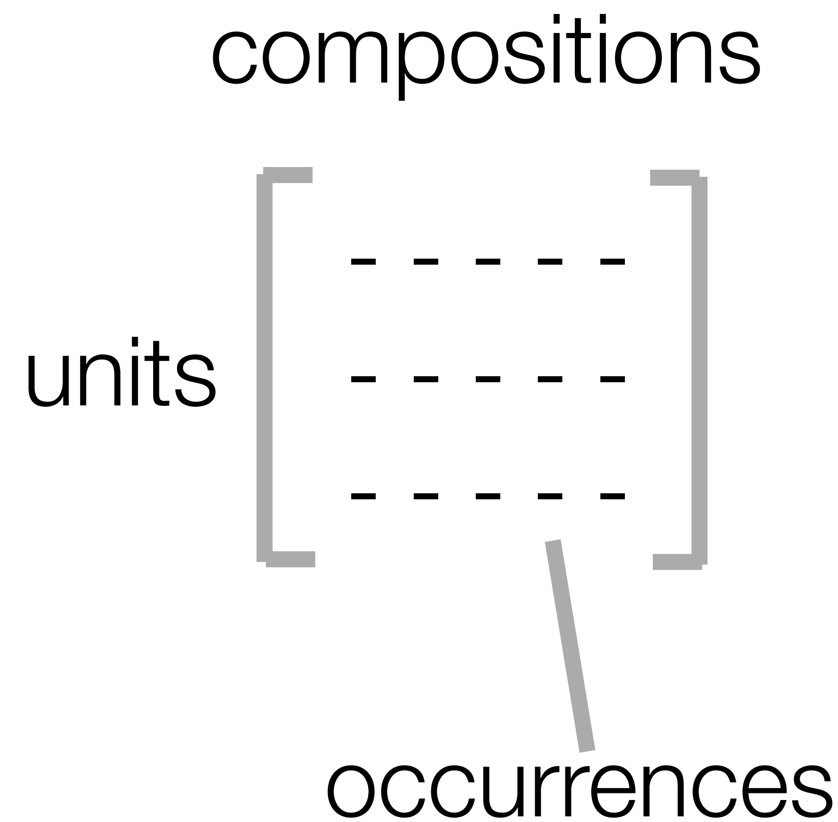
  - Obviously this still has a problem in the sense that all the indexed documents needed translation…

Cross-Language LSI

- …But, in the same way you construct queries from a single language, you can create representations for **new documents**
  - and then **append** these new vectors to the V matrix so they can be searched
  - The lower dimensional document vectors for unilingual documents should incorporate the **multilingual synonymy** captured from the training data

Multimodal LSI

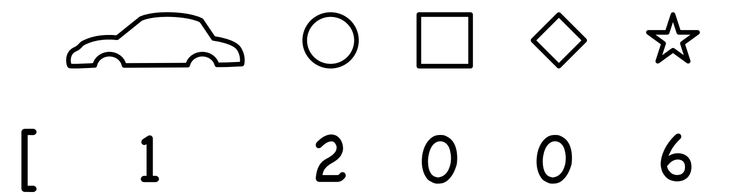
- Thus far, we've only considered BOWs from natural language
    - But there isn't anything in the mathematics of LSI that prevents us from applying it to any form of vector that records the **compositions** of occurrences in a document (or more generally a **unit**)

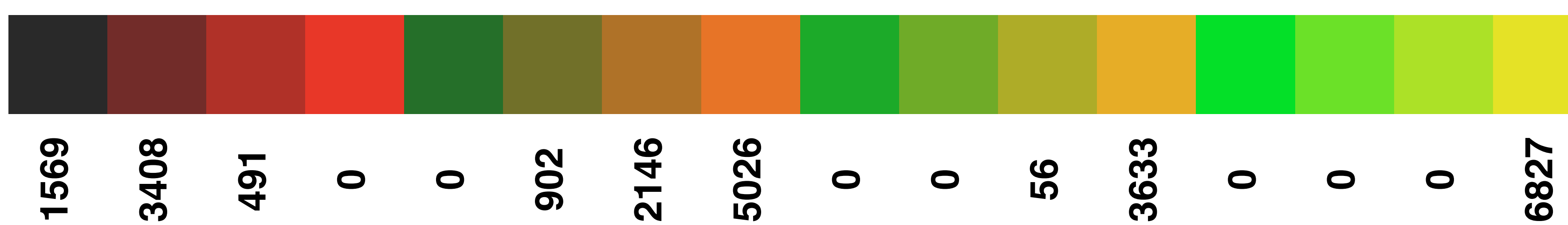


Problem Statement III: semantic image search

- Consider the problem of searching for images
  - Most of the time we want to search with words (matching against metadata)
    - But not all images have text associated with them…

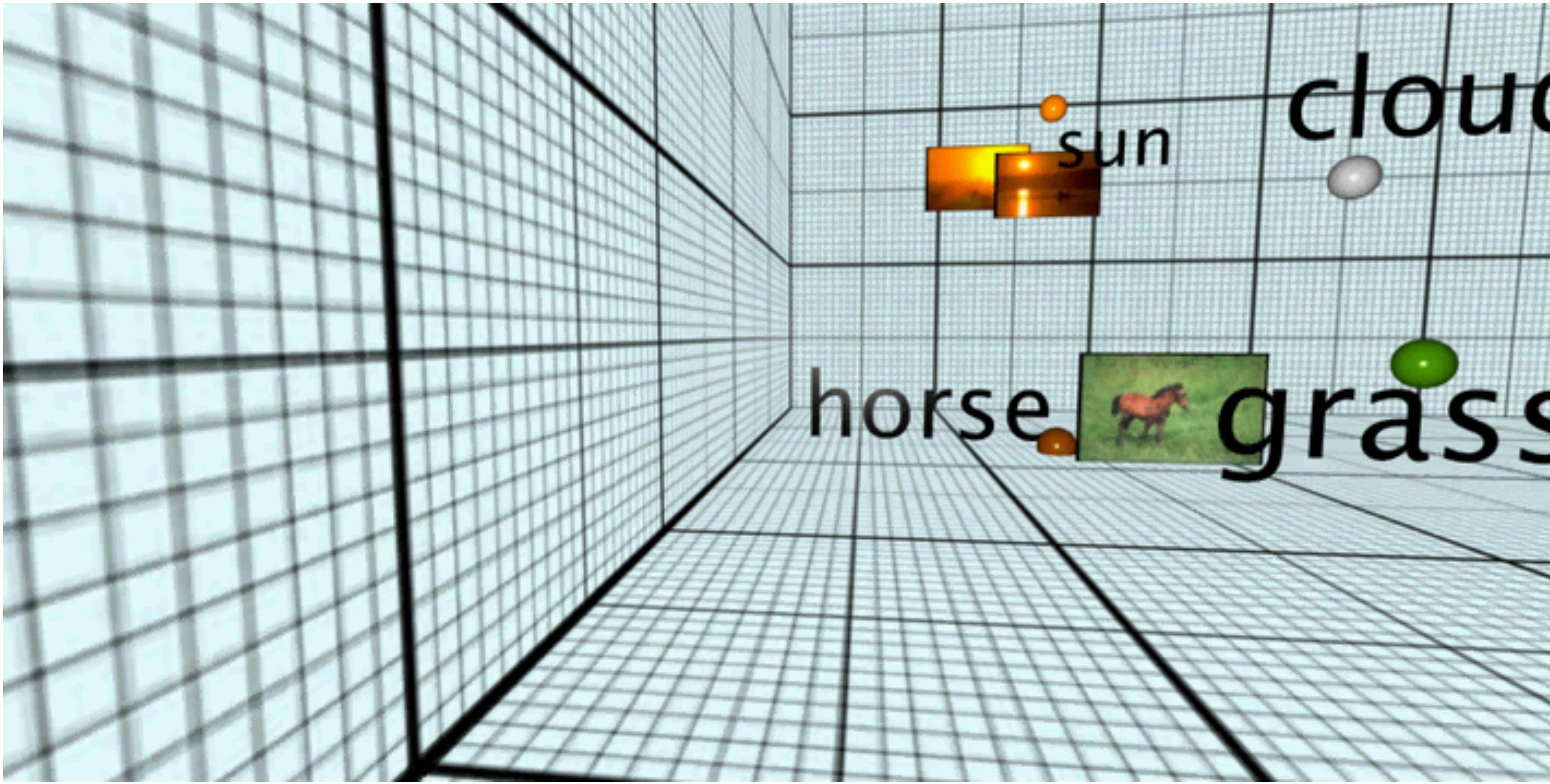
Image Representation





Conceptual Overview

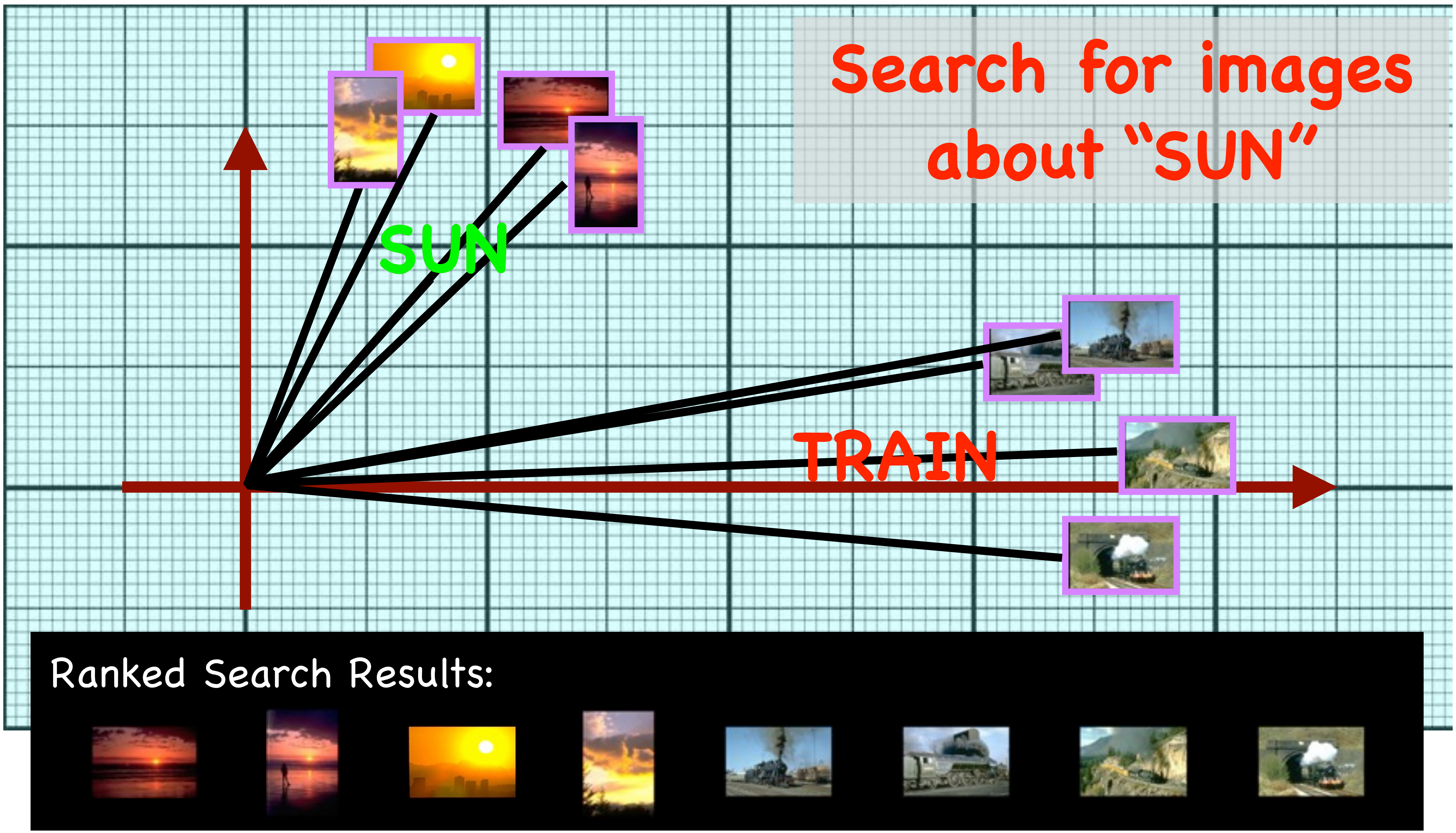
- Basic idea: Create a large multidimensional space in which images, keywords (or other metadata) and visual terms can be placed.
- In the training stage learn how keywords are related to visual terms and images.
  - Place related visual terms, images and keywords close-together within the space.
- In the projection stage unannotated images can be placed in the space based upon the visual terms they contain.
  - The placement should be such that they lie near keywords that describe them.


sun
clouc
horse
grass

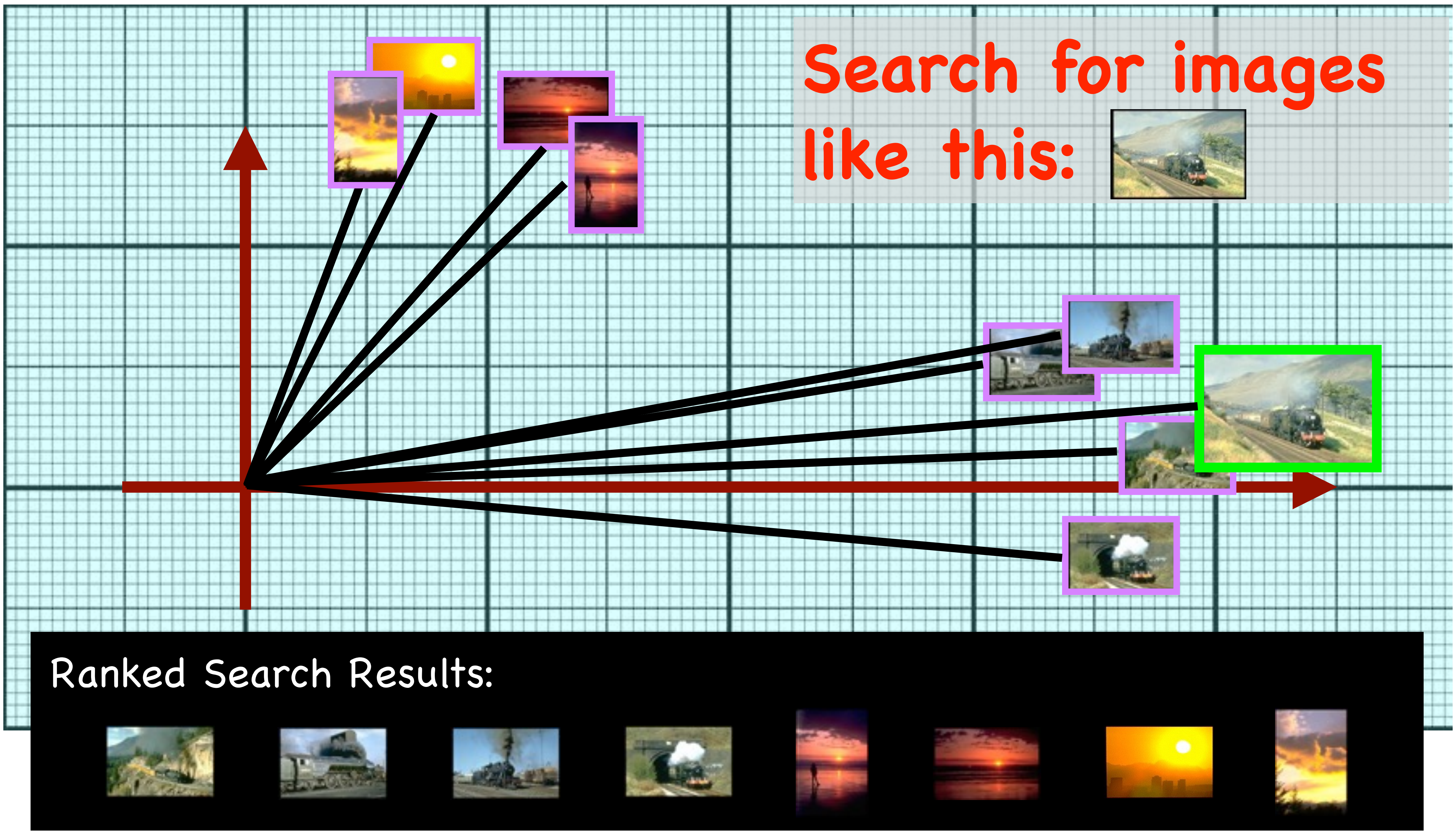
Applications of the lower dimensional space

- Finding images (both annotated and unannotated) by **similar words**.
- Finding images (both annotated and unannotated) by semantically **similar images**.
- Determining likely words for an image.
- Examining word->word and word->visual-term relationships.
  - Segmenting an image.

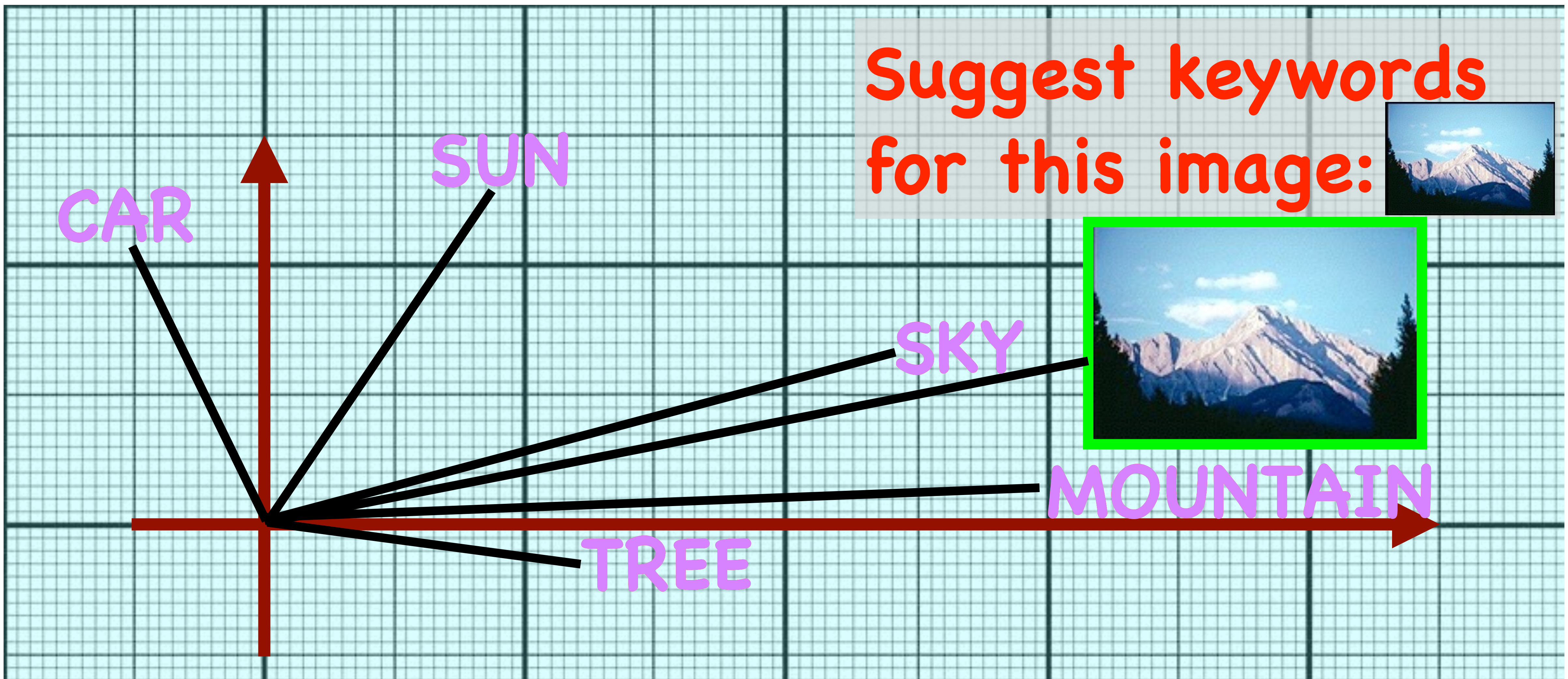
Searching by Keyword



Searching by Image



Suggesting Keywords

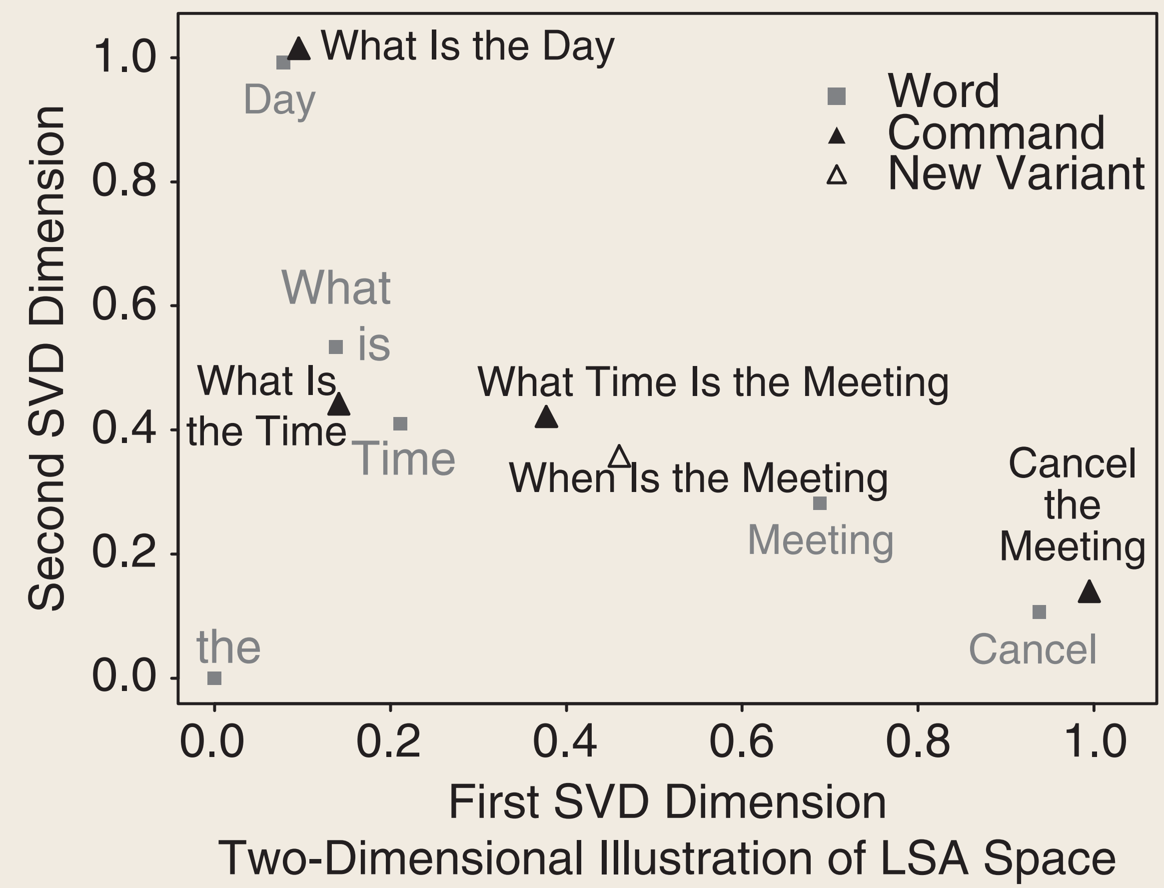


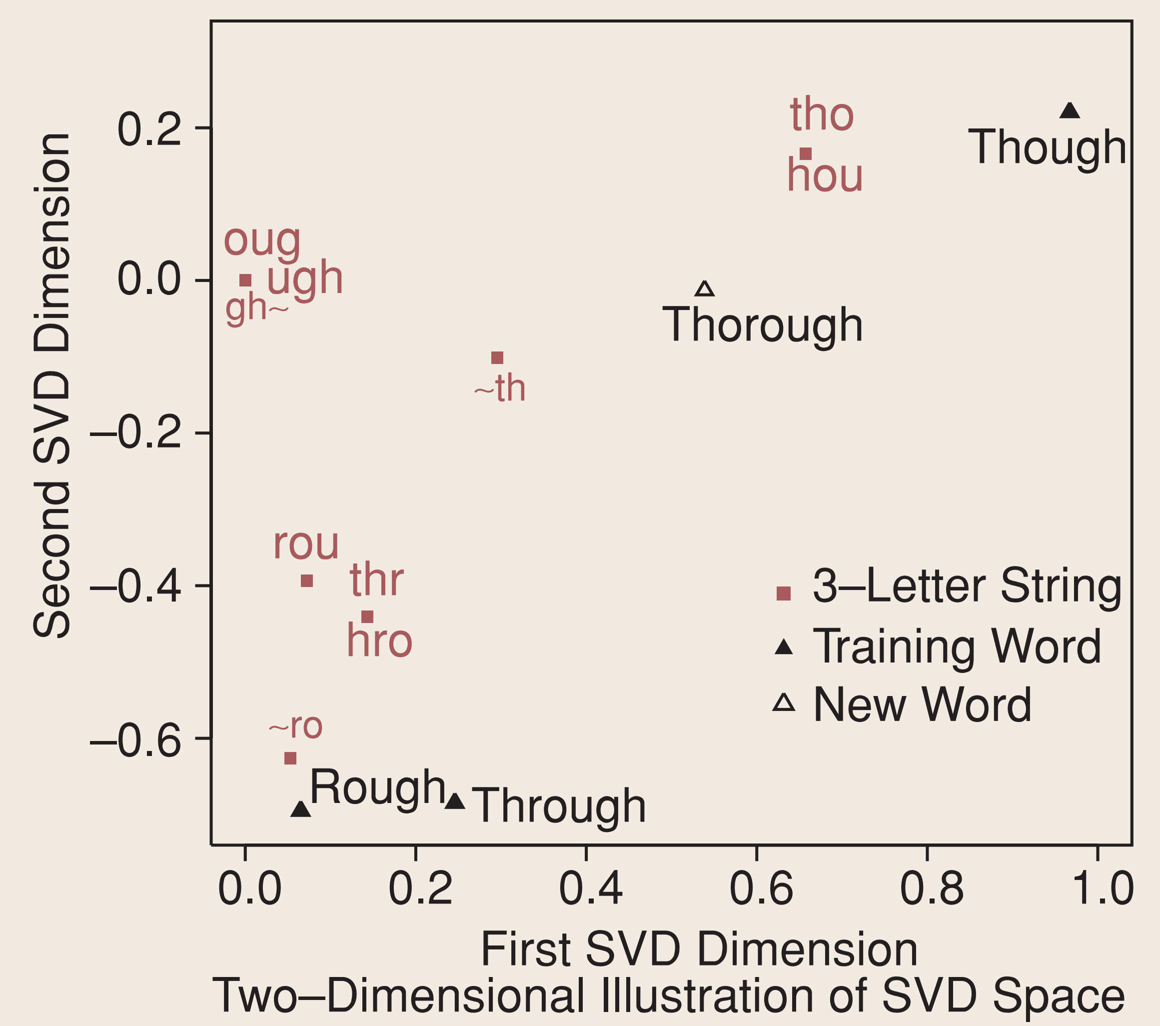
Suggested keywords:

SKY MOUNTAIN TREE SUN CAR

Other applications

- Language modelling
- Command-based speech recognition
- Spam filtering
- Pronunciation modelling
- TTS unit selection
- …





Images from: J. R. Bellegarda, "Latent semantic mapping [information retrieval]," in IEEE Signal Processing Magazine, vol. 22, no. 5, pp. 70-80, Sept. 2005.

Summary

- LSA is a powerful application of truncated SVD
- But it has a few potential problems
  - Polysemy
  - Highly abstract concepts
  - …
- In practical applications it has had some success
  - But there are newer techniques…