University of Southampton

COMP6237 Data Mining
Lecture 10: Semantic Spaces (Finding Features I)

Zhiwu Huang

Zhiwu.Huang@soton.ac.uk

Lecturer (Assistant Professor) @ VLC of ECS

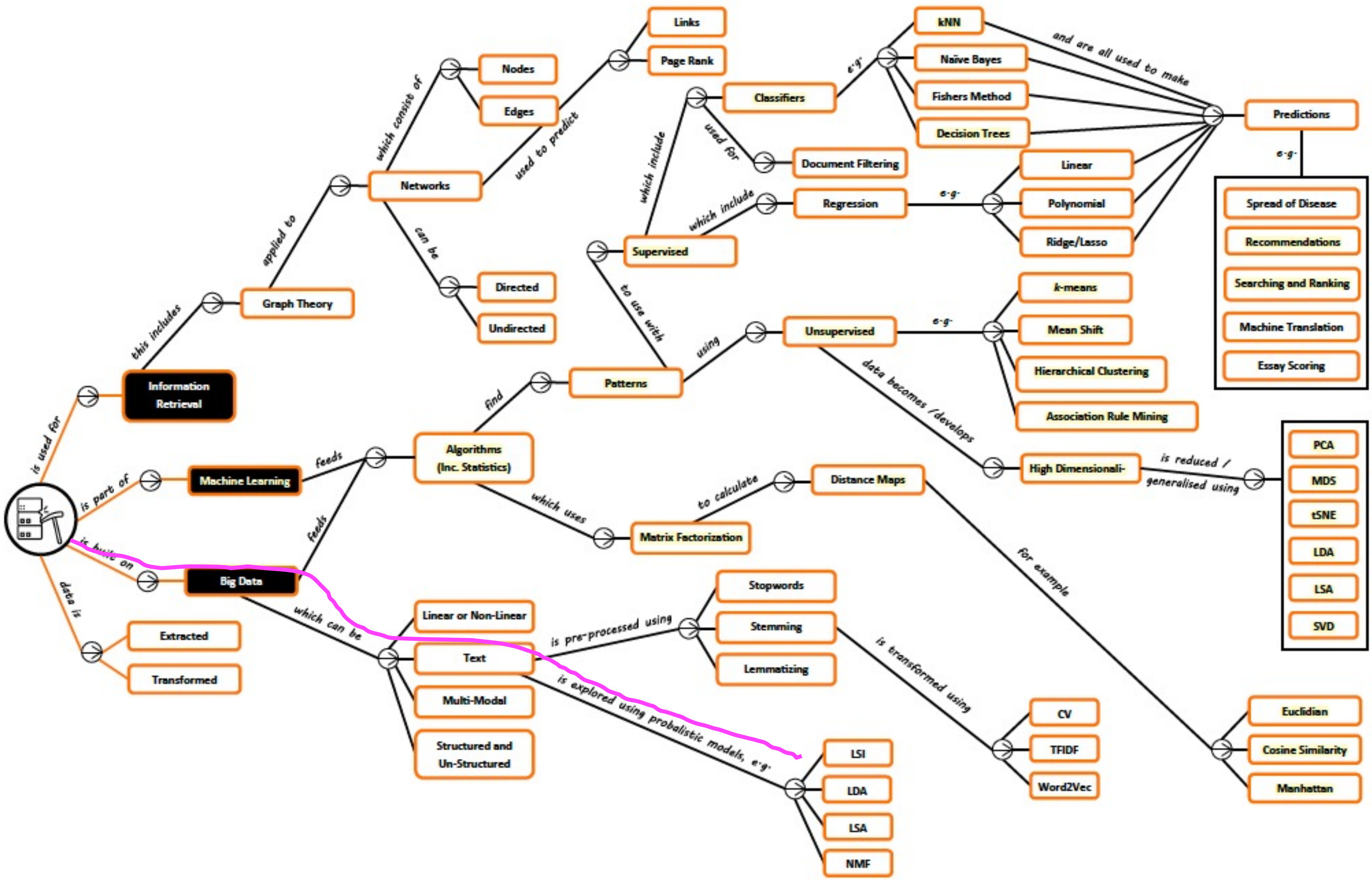
University of Southampton

Lecture slides available here:

http://comp6237.ecs.soton.ac.uk/zh.html

(Thanks to Prof. Jonathon Hare and Dr. Jo Grundy for providing the lecture materials used to develop the slides.)

Semantic Spaces – **Roadmap**



Finding Features I – **Textbook**

CHAPTER 10

Finding Independent Features

Most of the chapters so far have focused primarily on *supervised* classifiers, except Chapter 3, which was about *unsupervised* techniques called *clustering*. This chapter will look at ways to extract the important underlying features from sets of data that are not labeled with specific outcomes. Like clustering, these methods do not seek to make predictions as much as they try to characterize the data and tell you interesting things about it.

- Programming Collective Intelligence: Building Smart Web 2.0 Applications *T. Segaran*.

Semantic Spaces – **Overview (1/4)**

Distributional Semantics - Hypothesis:

Words that have similar distributions have similar meanings

"Words that occur in similar contexts have similar meanings"
Wittgenstein 1953

"A word is characterised by the company it keeps" Firth 1958

We can exploit this to uncover *hidden meanings*

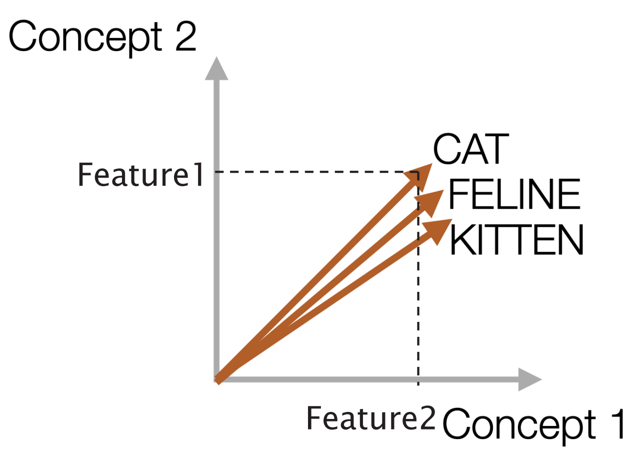
Credit: Jo Grundy

Semantic Spaces – **Overview (2/4)**

Semantic Spaces:

- represent word meanings as vectors that keep track of the words distributional history
- focus on semantic similarity
- similarity measured using geometrical methods

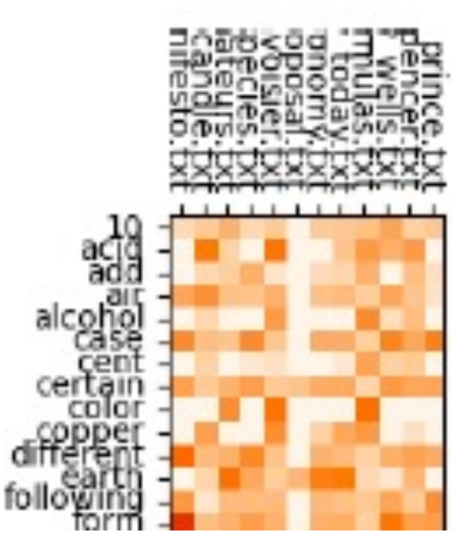
e.g. Cosine similarity between PC and Windows = 0.77
Cosine similarity between PC and window = 0.13
In Japanese, A. Utsumi, IEEE SMC 2010



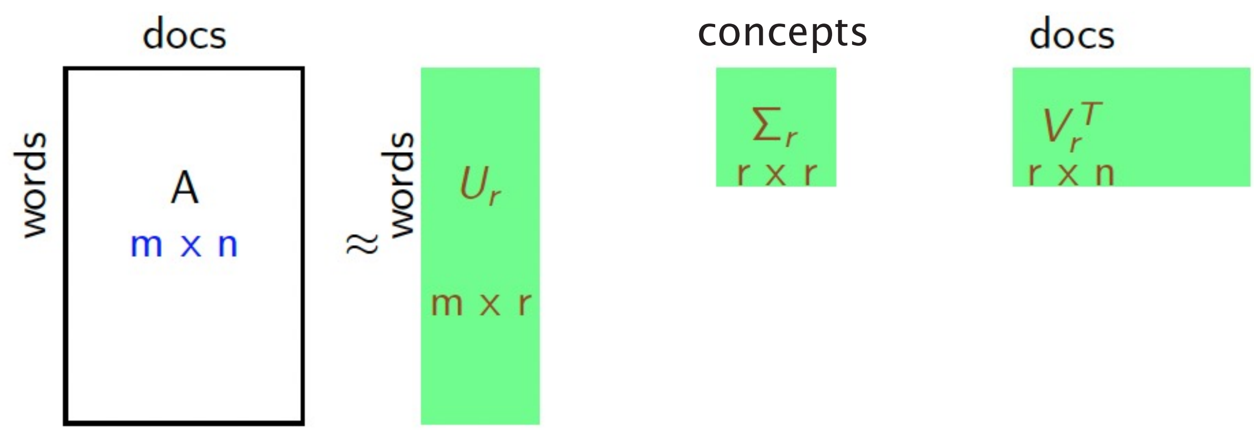
Credit: Jo Grundy, Jon Hare

Semantic Spaces – **Overview (3/4)**

Latent Semantic Analysis (LSA) using Bag of Words (BoW) & truncated SVD



Bag of Words (BoW)



Each row of V_r corresponds to an eigenvector of $A^T A$

- This means it is proportional to the covariance or correlation between the documents
- These are the *concepts*

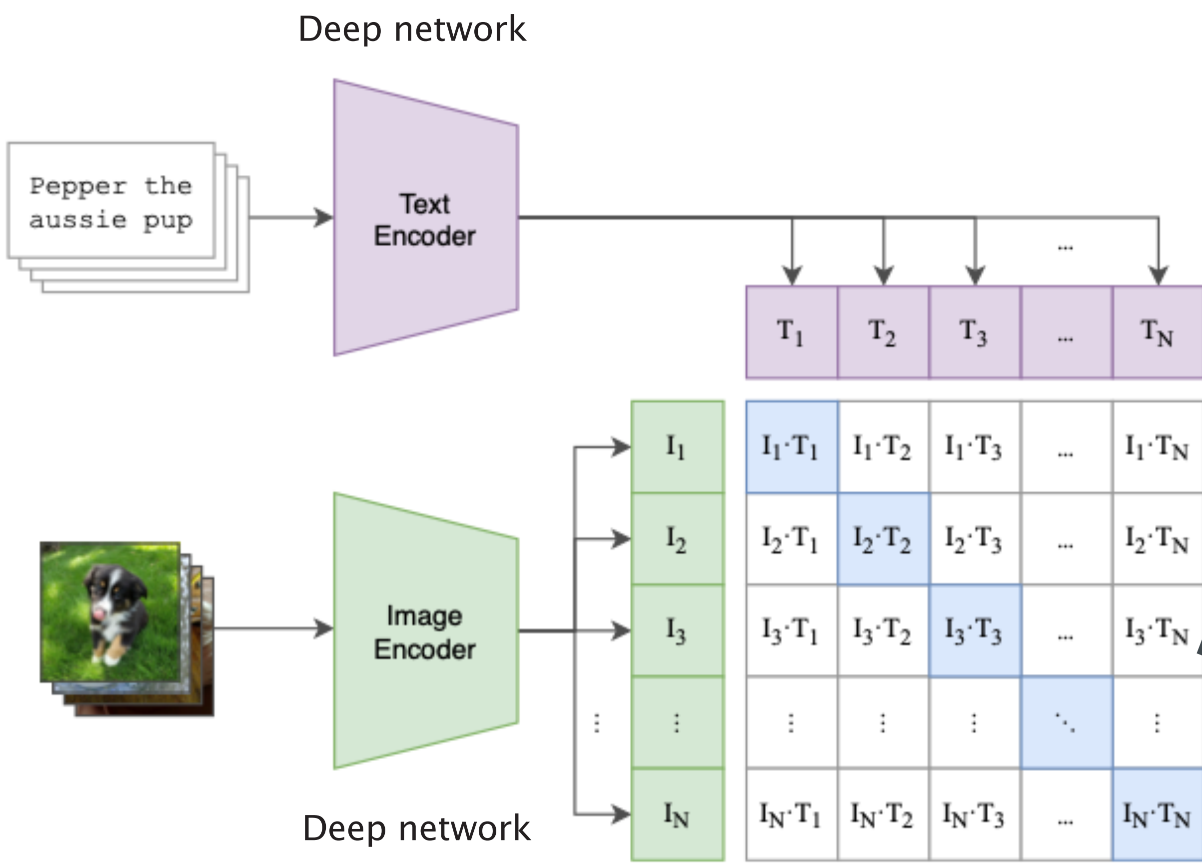
Features

- Each row of U_r describes a term as a vector of weights with respect to r *concepts*
- Each column of V_r describes a document as a vector of weights with respect to r concepts

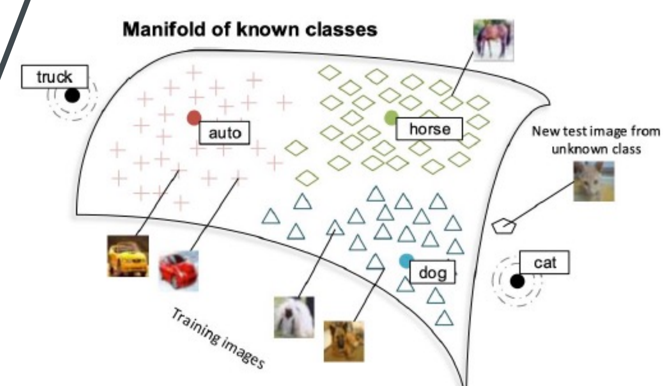
Credit: Jo Grundy

Semantic Spaces – **Overview (4/4)**

Contrastive Language-Image Pre-training (CLIP) uses an abundantly available source of supervision: the text paired with images found across the internet



I and T along diagonal direction are from the same semantic (paired image-text)



Socher et al. 2013

Radford et al. 2021, Learning Transferable Visual Models From Natural Language Supervision

Semantic Spaces – **Learning Outcomes**

- **LO1**: Demonstrate an understanding of techniques for finding independent semantic features, such as: (exam)
  - ❖ Comprehending the core concepts of Latent Semantic Analysis (LSA) and apply LSA on a dataset
  - ❖ Understanding the key pipeline of Contrastive Language-Image Pre-Training (CLIP)
  - ❖ Discussing the advantages and disadvantages of algorithms like LSA and CLIP
- **LO2**: Implement the learned algorithms for independent semantic feature learning (coursework)

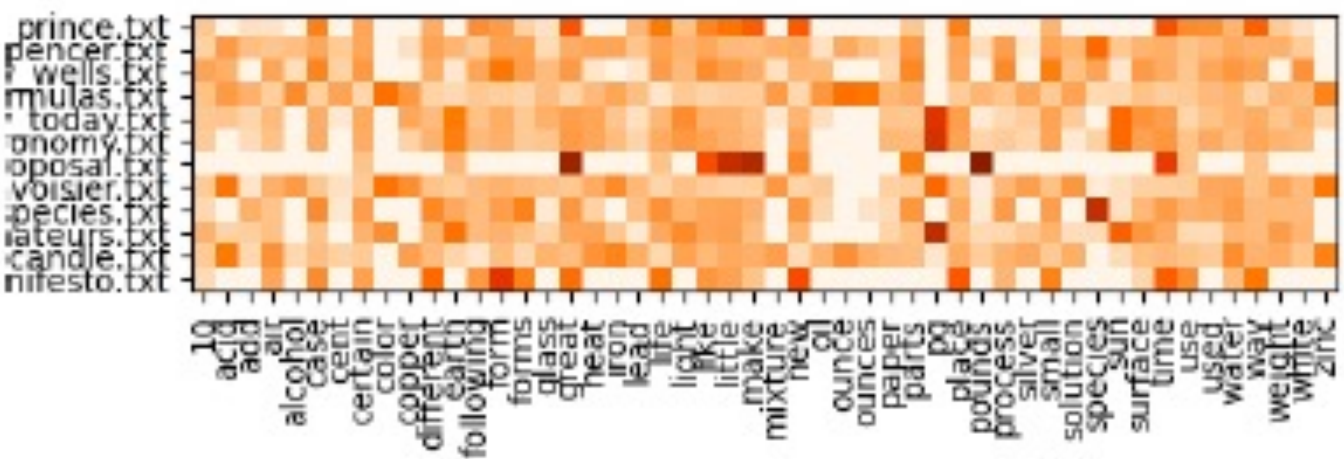
Assessment hints: Multi-choice Questions (single answer: concepts, calculation etc)

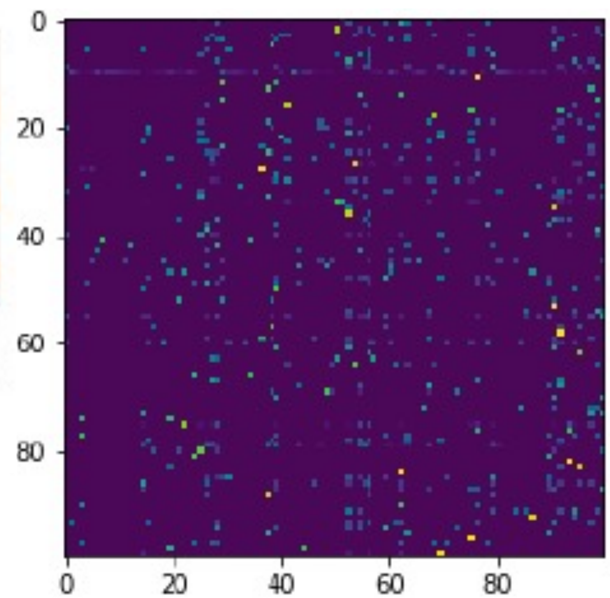
- *Textbook Exercises: textbooks (Programming + Mining)*
- *Other Exercises: https://www-users.cse.umn.edu/~kumar001/dmbook/sol.pdf*
- *ChatGPT or other AI-based techs*

Semantic Spaces – Latent Semantic Analysis

Matrix Construction:
Consider a term-document matrix which describes occurrences of terms in documents

- Sparse
- Weighted (e.g. TF.IDF)





Credit: Jo Grundy

Semantic Spaces – Latent Semantic Analysis

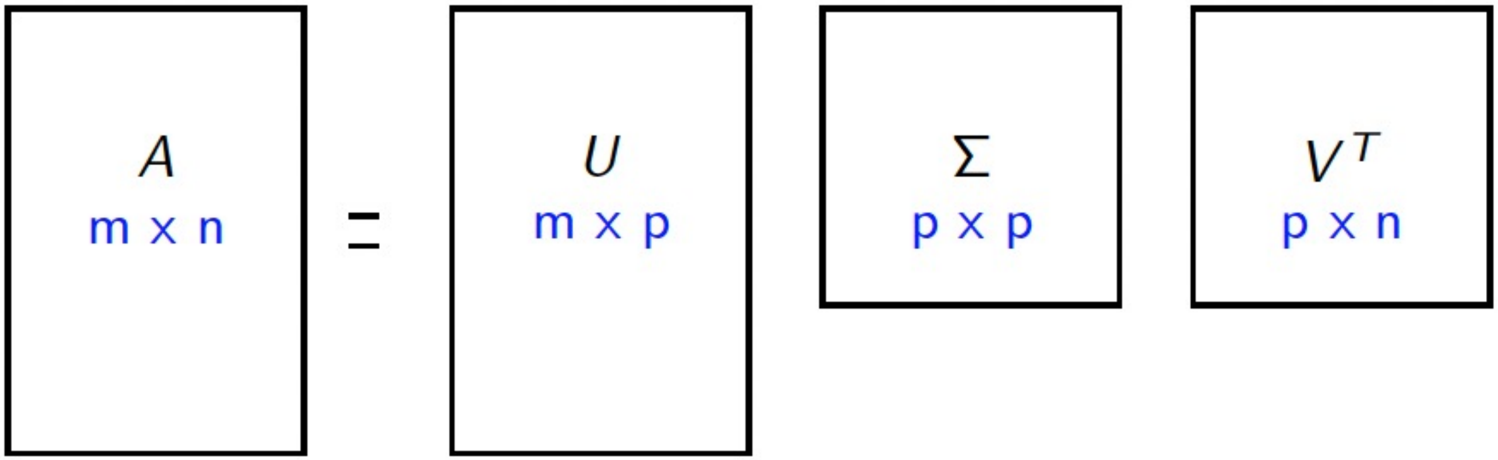
Latent Semantic Analysis (LSA) makes a low-rank approximation
It assumes the term-document matrix:

- is noisy, and should be de-noised
- is more sparse than it should be

Credit: Jo Grundy

Semantic Spaces – Recap SVD

$$A = U\Sigma V^T$$



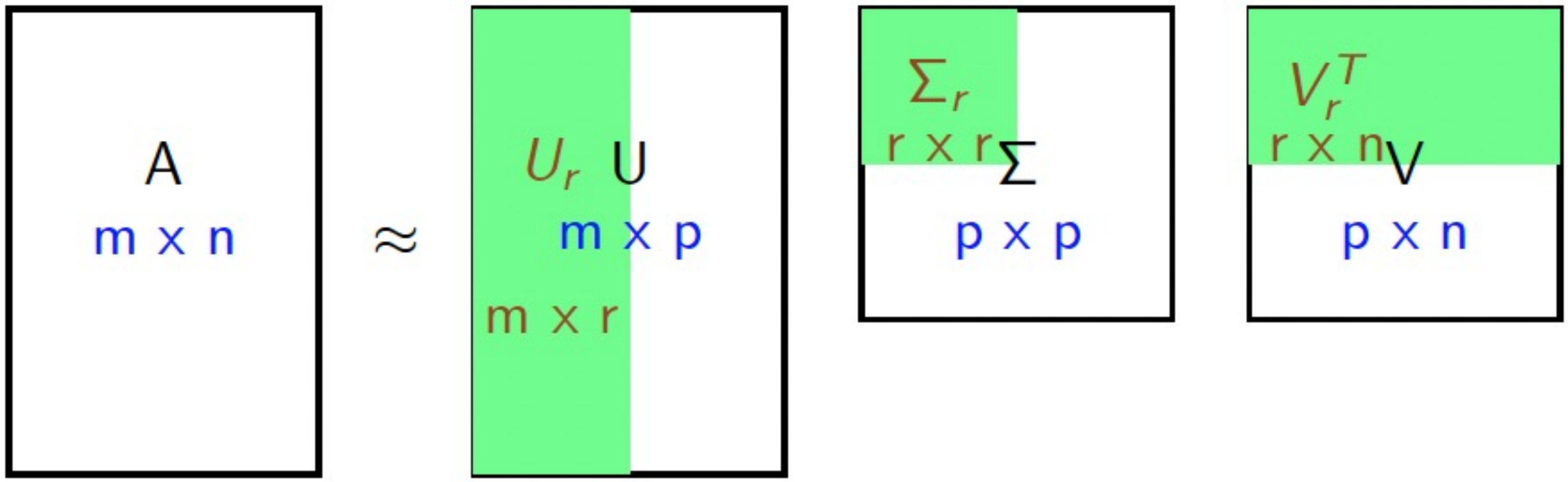
Where p is rank of matrix A

U called *left singular vectors*, contains the eigenvectors of AA^T,
V called *right singular vectors*, contains the eigenvectors of A^TA
Σ contains square roots of eigenvalues of AA^T and A^TA

If A is matrix of mean centred featurevectors, V contains principal components of the covariance matrix

Credit: Jo Grundy

Semantic Spaces – Recap Truncated SVD



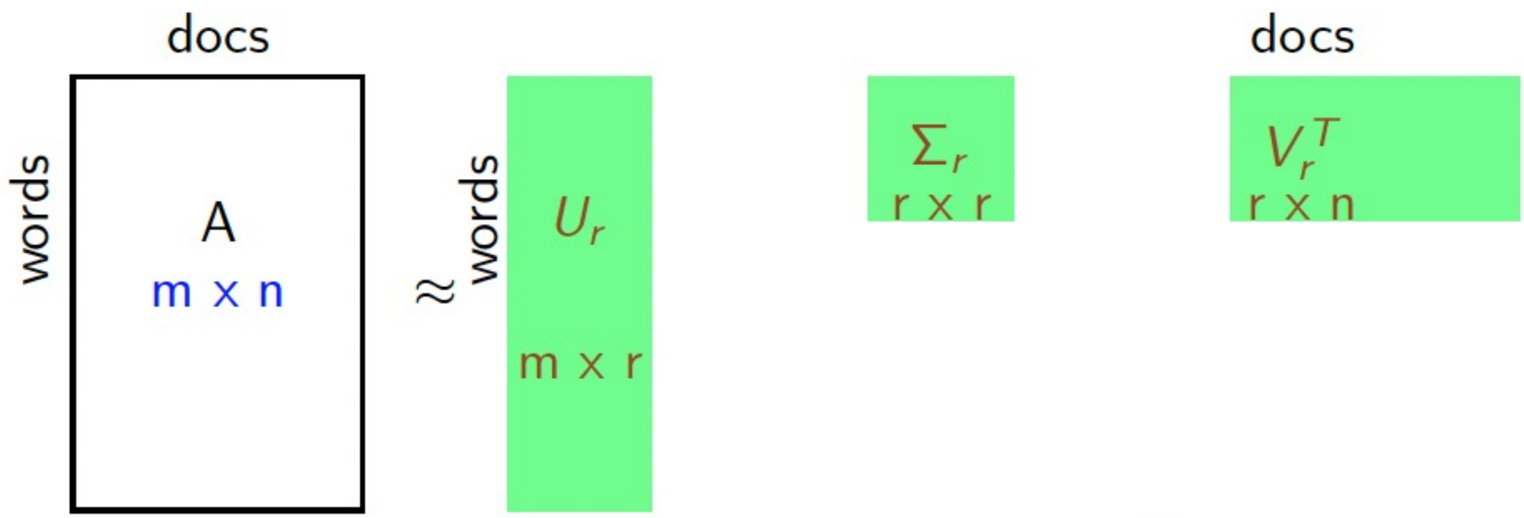
Uses only the largest r singular values (and corresponding left and right vectors)

This can give a *low rank approximation* of A, $\tilde{A} = U_r \Sigma_r V_r$

This has the effect of minimising the Frobenius norm of the difference between A and $\tilde{A}$

Credit: Jo Grundy

Semantic Spaces – Recap Truncated SVD



Each row of V_r corresponds to an eigenvector of A^TA

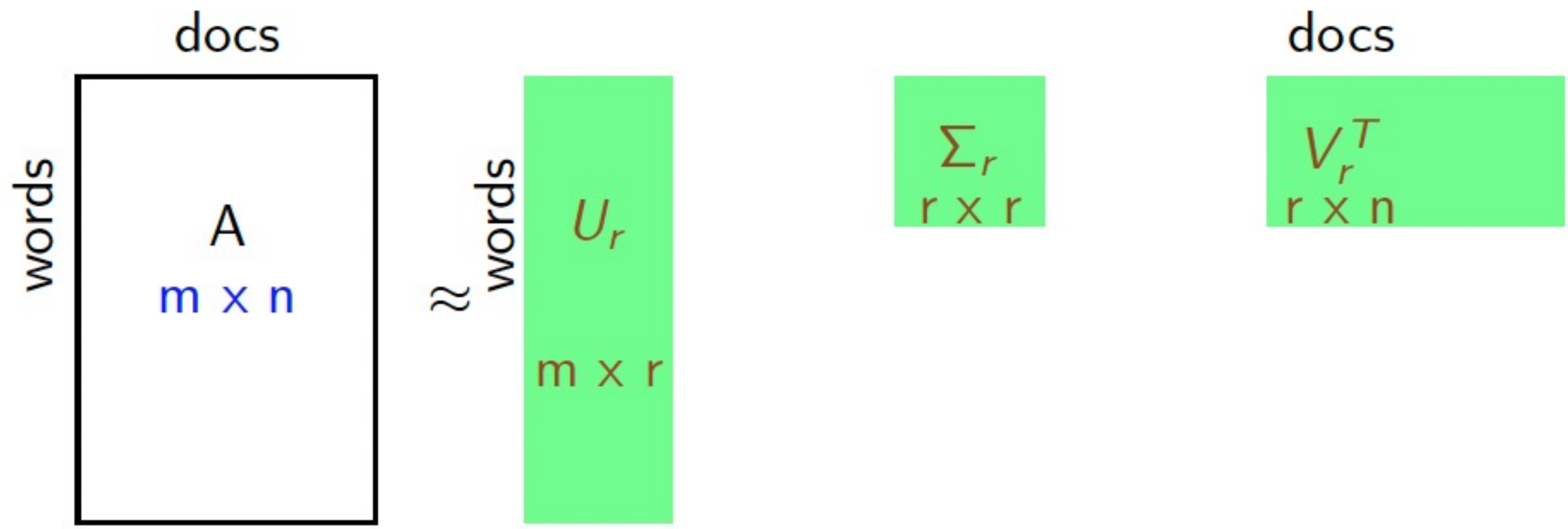
- This means it is proportional to the covariance or correlation between the documents
- These are the *concepts*

Each row of U_r describes a term as a vector of weights with respect to r *concepts*

Each column of V_r describes a document as a vector of weights with respect to r concepts

Credit: Jo Grundy

Semantic Spaces – LSA



Term concepts and document concepts have the same dimensionality, but represent different spaces.

Credit: Jo Grundy

Semantic Spaces – LSA

Example:
a set of strings:

m1	"Human machine interface for ABC computer applications"
m2	"A survey of user opinion of computer system response time"
m3	"The EPS user interface management system"
m4	"System and human system engineering testing of EPS"
m5	"Relation of user perceived response time to error measurement"
g1	"The generation of random, binary, ordered trees"
g2	"The intersection graph of paths in trees"
g3	"Graph minors IV: Widths of trees and well-quasi-ordering"
g4	"Graph minors: A survey"

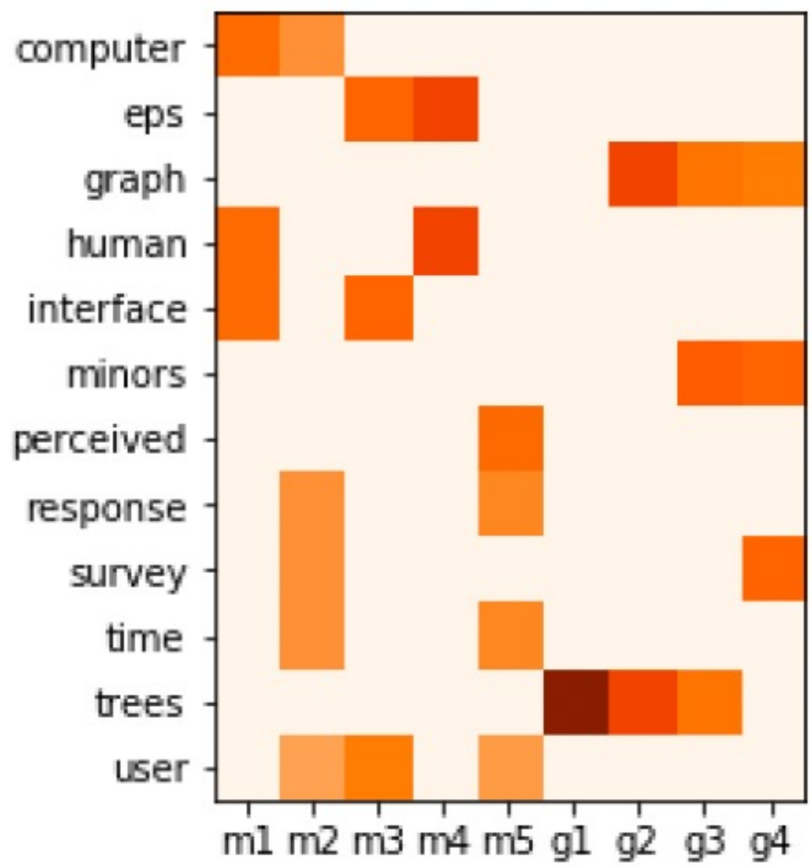
`http://lsa.colorado.edu/papers/dp1.LSAintro.pdf`

Credit: Jo Grundy

Semantic Spaces – LSA

calculate TF.IDF

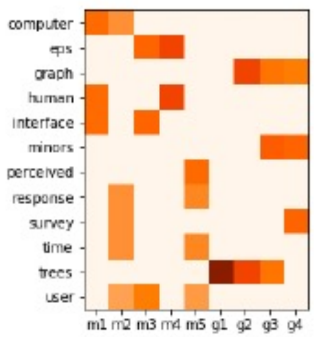
0.58	0.46	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.6	0.71	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.71	0.55	0.52
0.58	0.	0.	0.71	0.	0.	0.	0.	0.
0.58	0.	0.6	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.63	0.6
0.	0.	0.	0.	0.58	0.	0.	0.	0.
0.	0.46	0.	0.	0.49	0.	0.	0.	0.
0.	0.46	0.	0.	0.	0.	0.	0.	0.6
0.	0.46	0.	0.	0.49	0.	0.	0.	0.
0.	0.	0.	0.	0.	1.	0.71	0.55	0.
0.	0.4	0.52	0.	0.43	0.	0.	0.	0.



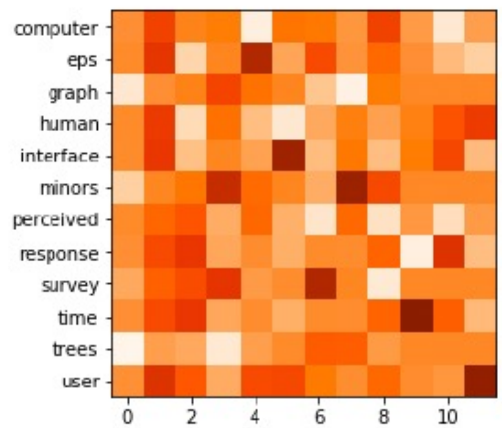
Credit: Jo Grundy

Semantic Spaces – LSA

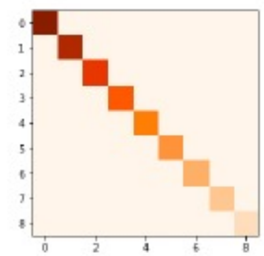
SVD: $A = U\Sigma V^T$



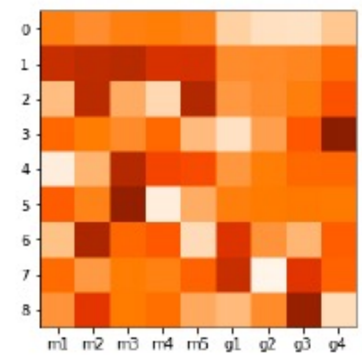
A



U



Σ



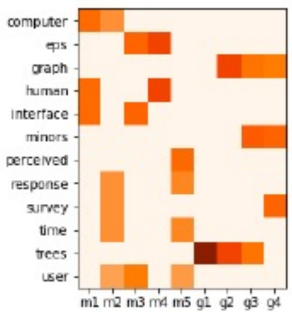
V

We then reduce the dimensionality by choosing only the first few eigenvalues (in Σ) and the corresponding columns in U and V.

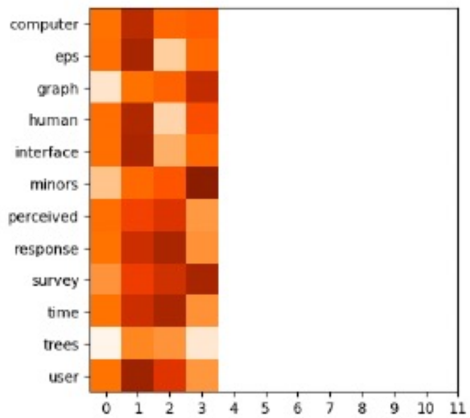
Credit: Jo Grundy

Semantic Spaces – LSA

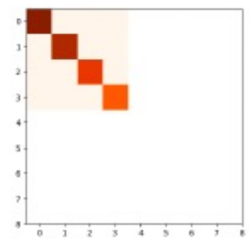
SVD: $A \approx U_r \Sigma_r V_r^T \quad r = 4$



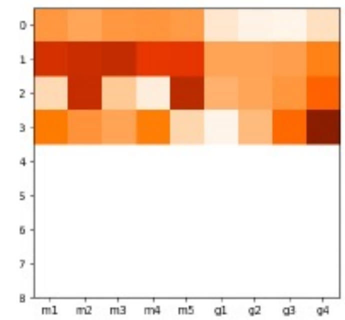
A



U_r



Σ_r



V_r

Each row of U_r describes a word as a vector of weights with respect to r *concepts*

Each column of V_r describes the title as a vector of weights with respect to r concepts

Credit: Jo Grundy

Semantic Spaces – LSA

What do these 'concepts' mean?

U_r gives us the weighting of the words for each concept

We can show that weighting for each concept:

	different	different	different	different
	copper	copper	copper	copper
	color	color	color	color
	certain	certain	certain	certain
	cent	cent	cent	cent
	case	case	case	case
	alcohol	alcohol	alcohol	alcohol
	air	air	air	air
	add	add	add	add
	acid	acid	acid	acid
	10	10	10	10
Topic:	0	1	2	3

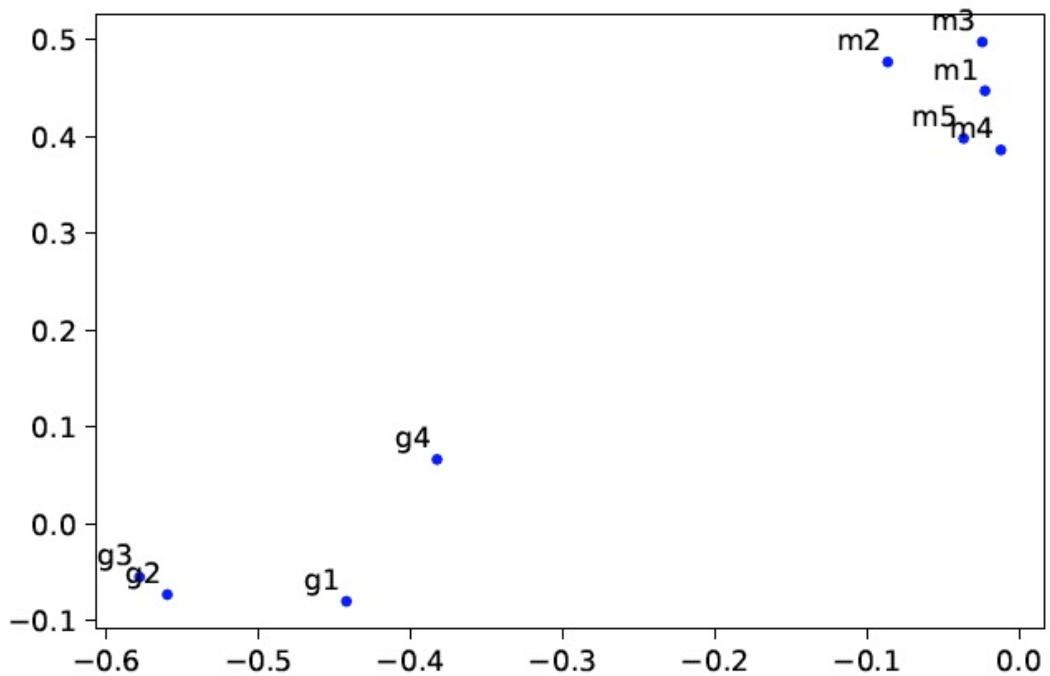
This shows us perhaps that the concepts are not always particularly meaningful.

Credit: Jo Grundy

Semantic Spaces – LSA

Cosine similarity of document vectors can be compared ($r = 2$)

- Vectors for "m1" and "m2" give cosine similarity $= 0.93$
- Vectors for "g1" and "g2" give cosine similarity $= 0.83$
- Vectors for "g1" and "m1" give cosine similarity $= 0.18$



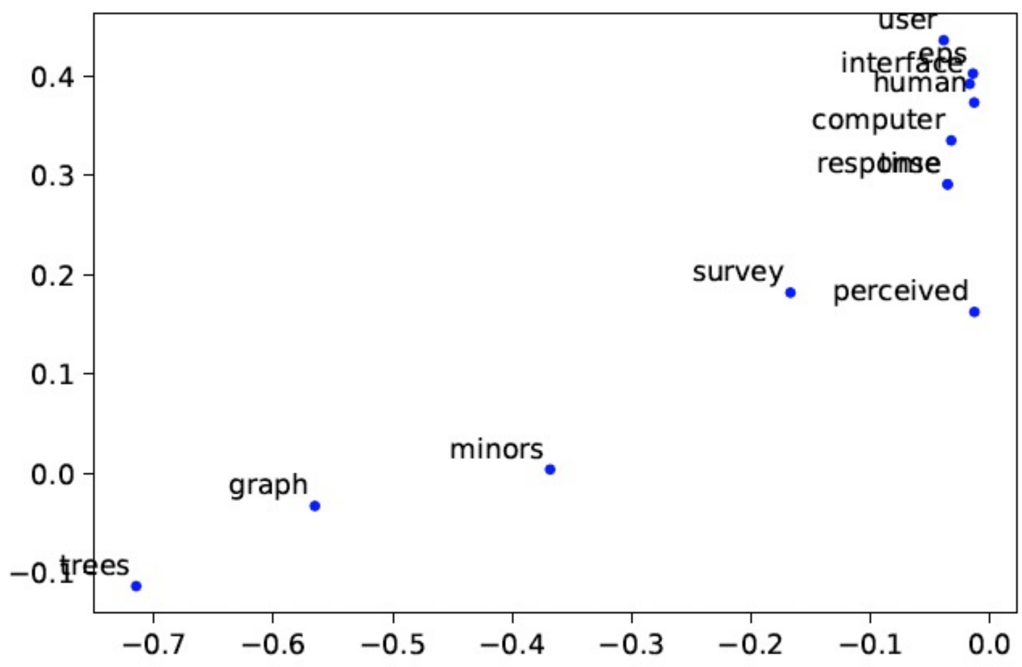
Clustering algorithms can be used on the vectors

Credit: Jo Grundy

Semantic Spaces – LSA

Cosine similarity of word vectors can be compared ($r = 2$)

- Vectors for "human" and "interface" give cos similarity $= 0.95$
- Vectors for "human" and "user" give cos similarity $= 0.11$
- Vectors for "graph" and "minor" give cos similarity $= 0.90$



Clustering algorithms can be used on the vectors

Credit: Jo Grundy

Semantic Spaces – LSI

Latent Semantic Indexing (LSI) LSA can be used for document retrieval

- Given Query: view as query vector $\boldsymbol{q}$
- Project $\boldsymbol{q}$ in to document space
- Compare with document vectors, find closest

Results work mathematically
However, results may not be easy to interpret in terms of natural language.

Credit: Jo Grundy

Semantic Spaces – LSA

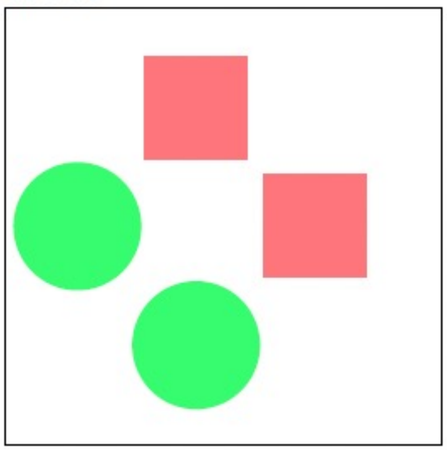
Problems?

- Polysemious words - with multiple meanings - aren't captured
  - The vector representation averages all meanings of the word
  - e.g. 'fit' is an adjective and a verb
- Word order is ignored (use n-grams?) *An n-gram is a sequence of n words*
- LSA assumes words and documents form a joint Gaussian distribution, however a Poisson distribution is observed

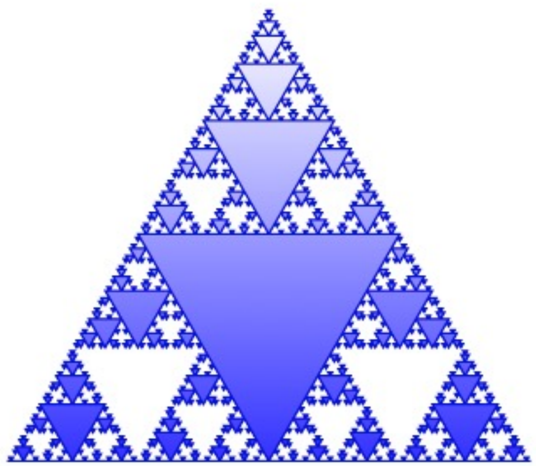
Credit: Jo Grundy

Semantic Spaces – LSA

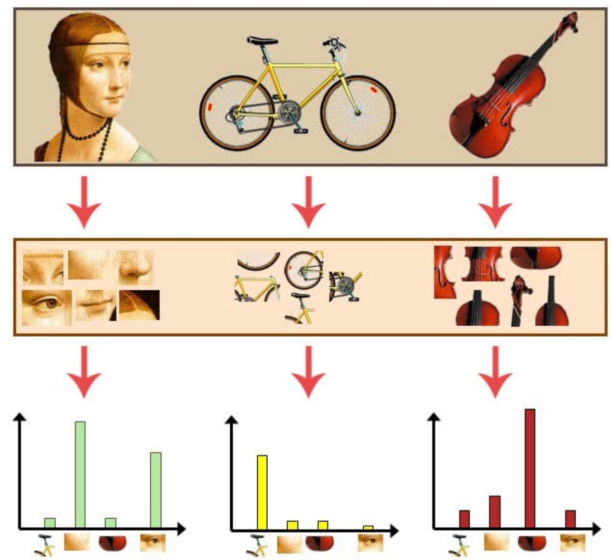
So far: *bag of words* (BOW) from natural language.
However the maths should work for compositions of occurrences in any unit.
For example, in image search we might want to search for other images with circles.
The image could be encoded with the number of different shapes it has.



○	△	□	×	*
2	0	3	0	0



○	△	□	×	*
0	∞	0	0	0



Histogram of visual words

Credit: Jo Grundy, Jon Hare

Semantic Spaces – LSA

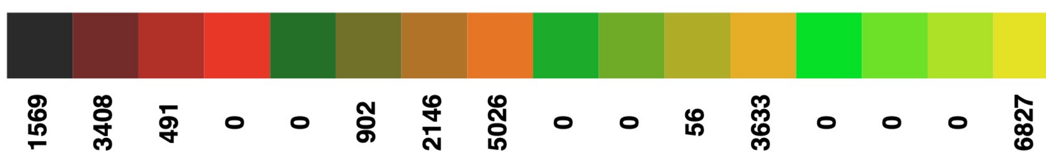
Need to make a large multidimensional space in which images, keywords and visual terms can be placed

In training:

- Learn how images and keywords are related
- Place images and keywords close together in the space

Unannotated images can be placed in the space based on the visual terms they contain

- Images can be placed based on their visual terms in the space
- They should lie near the keywords that describe them

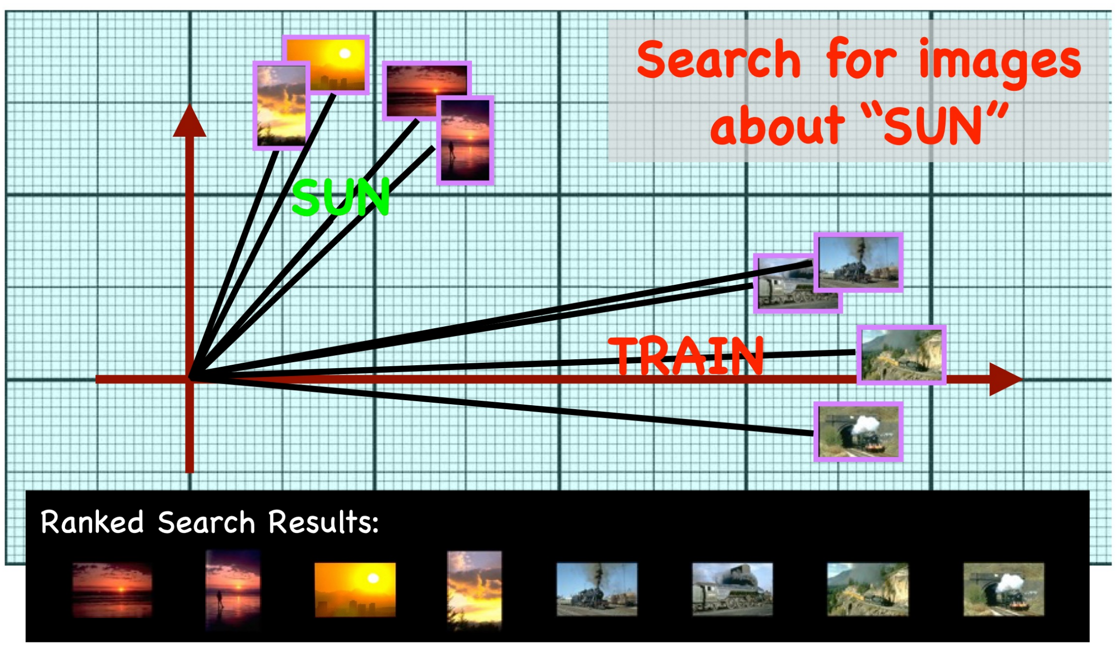


Credit: Jo Grundy, Jon Hare

Semantic Spaces – LSA

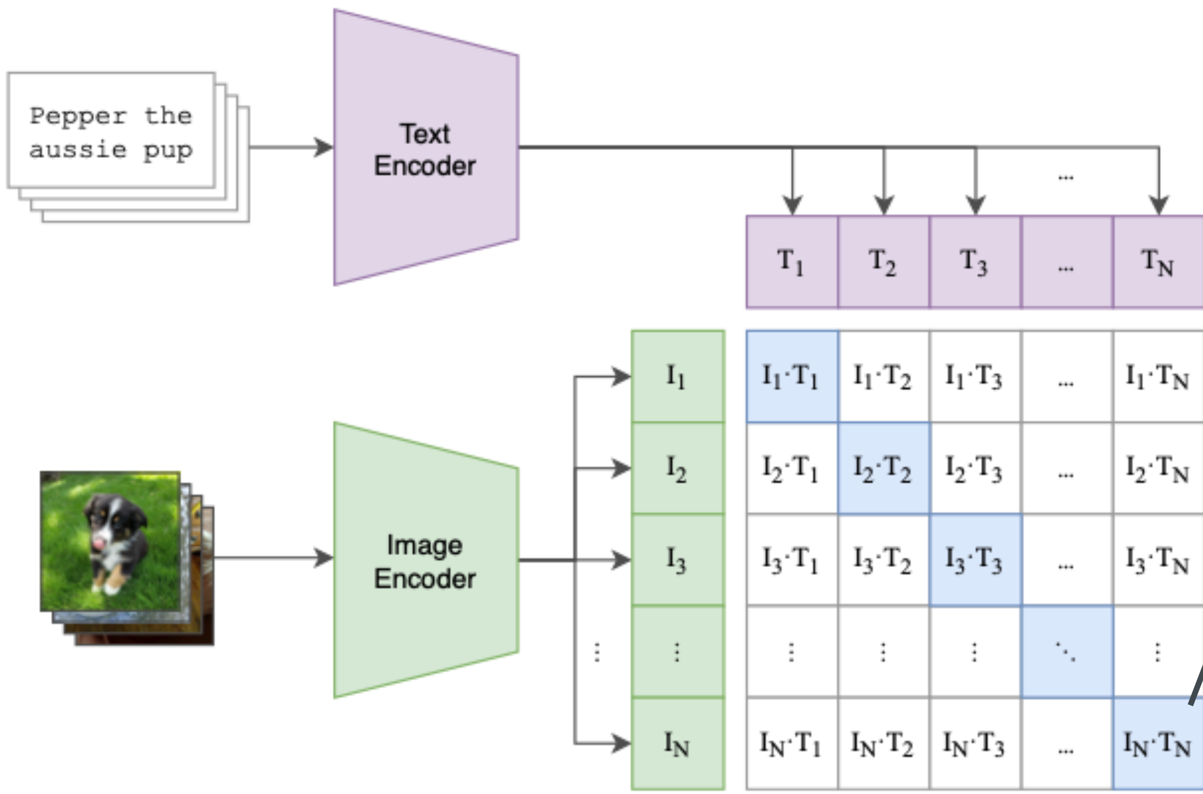
This lower dimensional space can be used to:

- Find Images using similar words
- Find images with similar images
- Return possible key words for an image
- Find relationships between words, and between words and visual terms

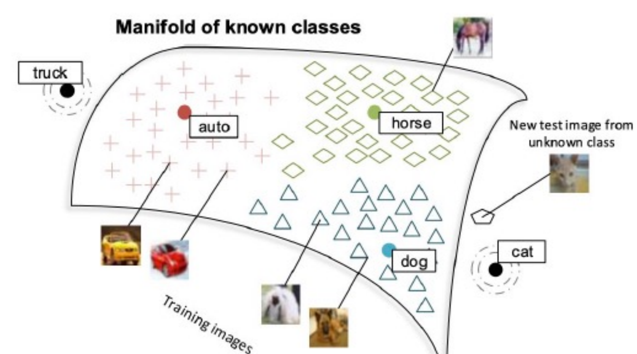


Credit: Jo Grundy, Jon Hare

Semantic Spaces – Contrastive Language-Image Pre-training (**CLIP**)



I and T along diagonal direction are from the same semantic (paired image-text)



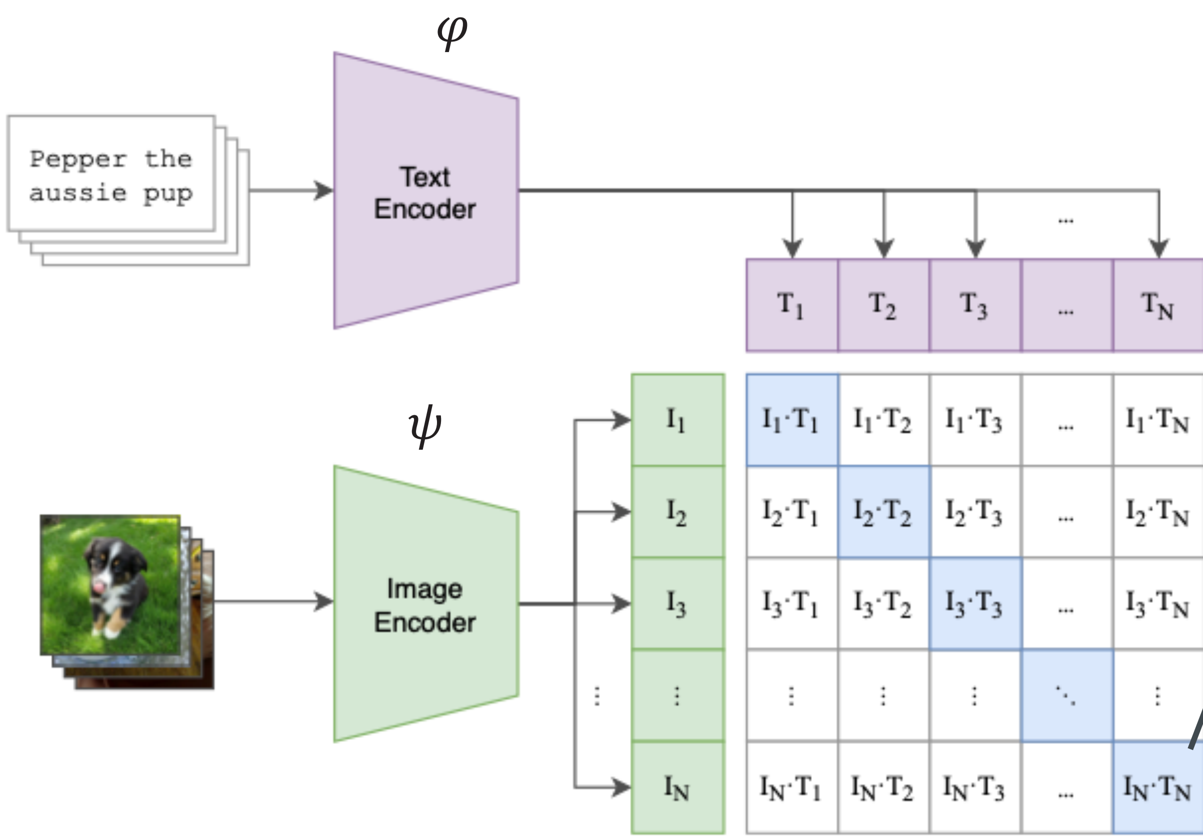
Socher et al. 2013

CLIP uses an abundantly available source of supervision: the text paired with images found across the internet

Radford et al. 2021, Learning Transferable Visual Models From Natural Language Supervision

Semantic Spaces – Contrastive Language-Image Pre-training (**CLIP**)

Training phase



I and T along diagonal direction are from the same semantic (paired image-text)

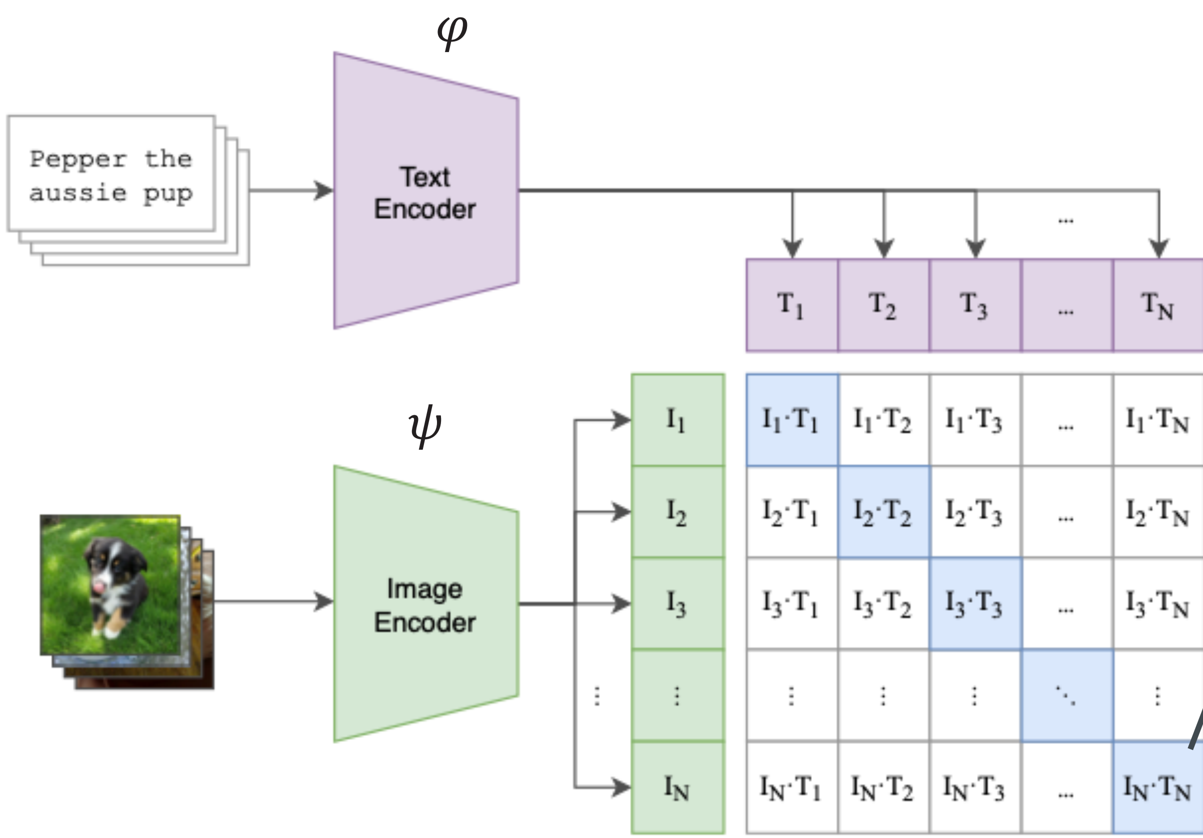
Radford et al., 2021

Text encoder: find the latent features of text with a non-linear mapping $\varphi(X)$, where X indicates the text raw features

Image encoder: find the latent features of text with a non-linear mapping $\psi(Y)$, where Y indicates the image raw features

Semantic Spaces – Contrastive Language-Image Pre-training (**CLIP**)

Training phase

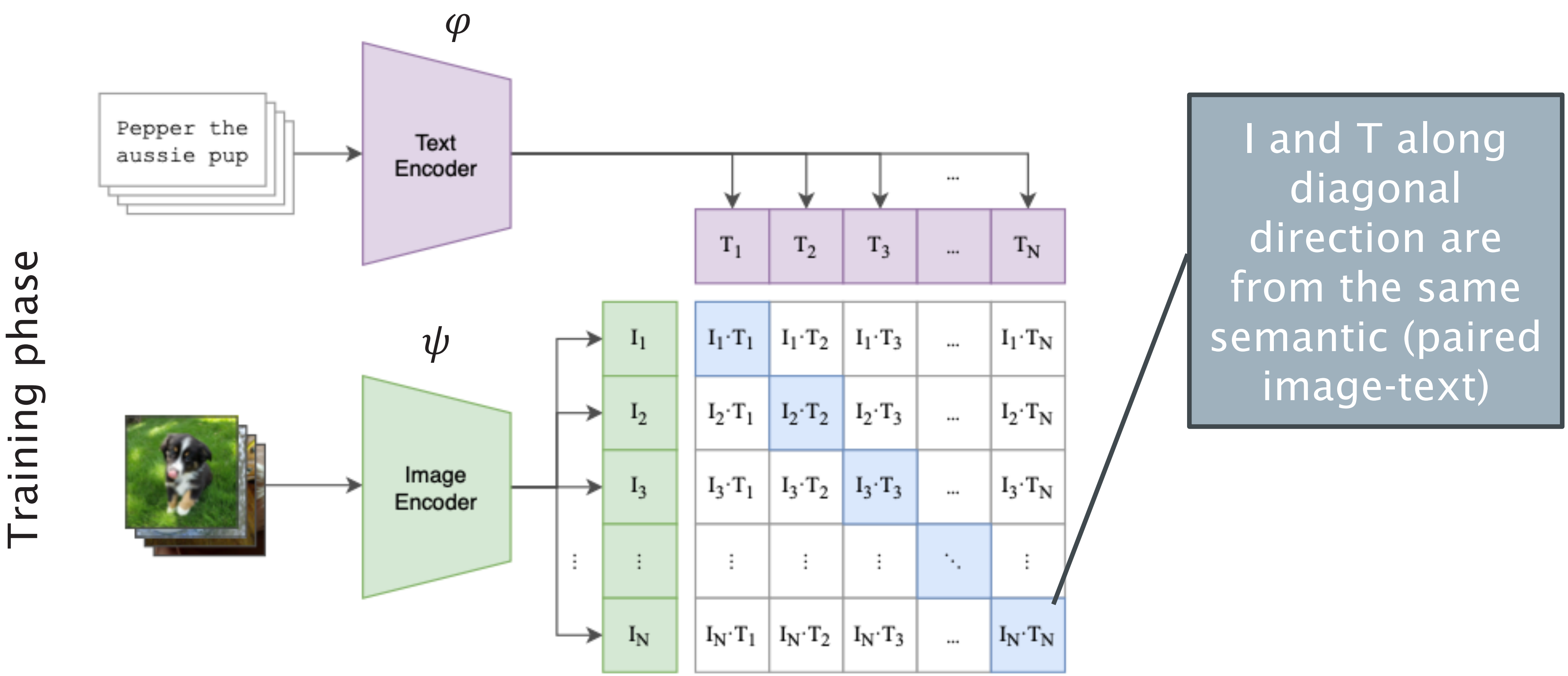


I and T along diagonal direction are from the same semantic (paired image-text)

Radford et al., 2021

Similarity: $A = \varphi(X)\psi(Y)^T$, where X, Y indicate the text and image raw features, φ and ψ are the mapping functions of text/image encoder

Semantic Spaces – Contrastive Language-Image Pre-training (**CLIP**)



Radford et al., 2021

Supervision for training: $\min \ell(A, G)$, where A, G are the similarity matrix and ground truth, and ℓ is the cross-entropy loss (the lower the closer distance from A to G

Semantic Spaces – Contrastive Language-Image Pre-training (**CLIP**)

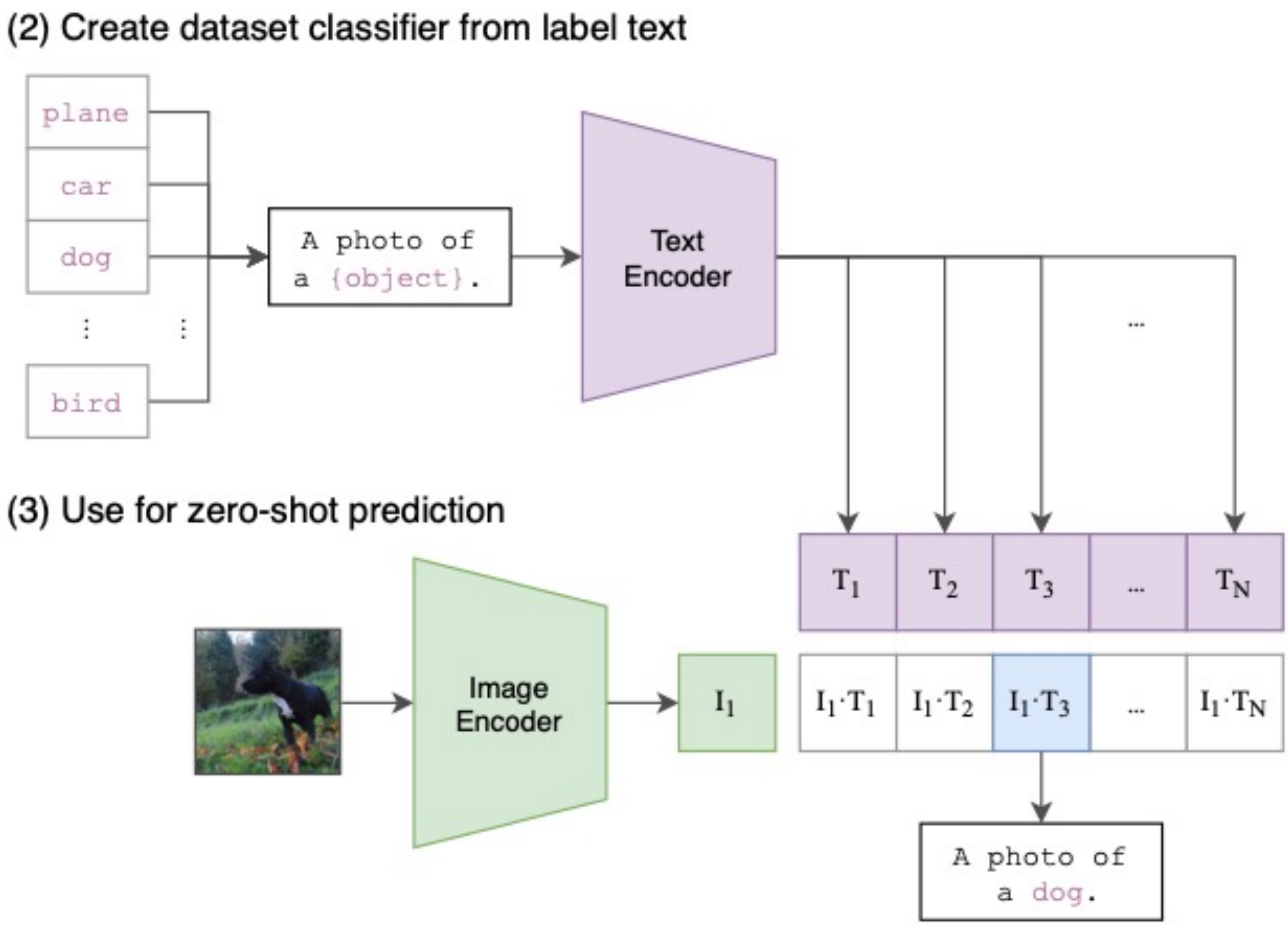
Numpy-like pseudocode for the core of an implementation of CLIP

```
# image_encoder - ResNet or Vision Transformer
# text_encoder  - CBOW or Text Transformer
# I[n, h, w, c] - minibatch of aligned images
# T[n, l]       - minibatch of aligned texts
# W_i[d_i, d_e] - learned proj of image to embed
# W_t[d_t, d_e] - learned proj of text to embed
# t             - learned temperature parameter

# extract feature representations of each modality
I_f = image_encoder(I) #[n, d_i]
T_f = text_encoder(T)  #[n, d_t]

# joint multimodal embedding [n, d_e]
I_e = l2_normalize(np.dot(I_f, W_i), axis=1)
T_e = l2_normalize(np.dot(T_f, W_t), axis=1)

# scaled pairwise cosine similarities [n, n]
logits = np.dot(I_e, T_e.T) * np.exp(t)

# symmetric loss function
labels = np.arange(n)
loss_i = cross_entropy_loss(logits, labels, axis=0)
loss_t = cross_entropy_loss(logits, labels, axis=1)
loss   = (loss_i + loss_t)/2
```

Radford et al. 2021, Learning Transferable Visual Models From Natural Language Supervision

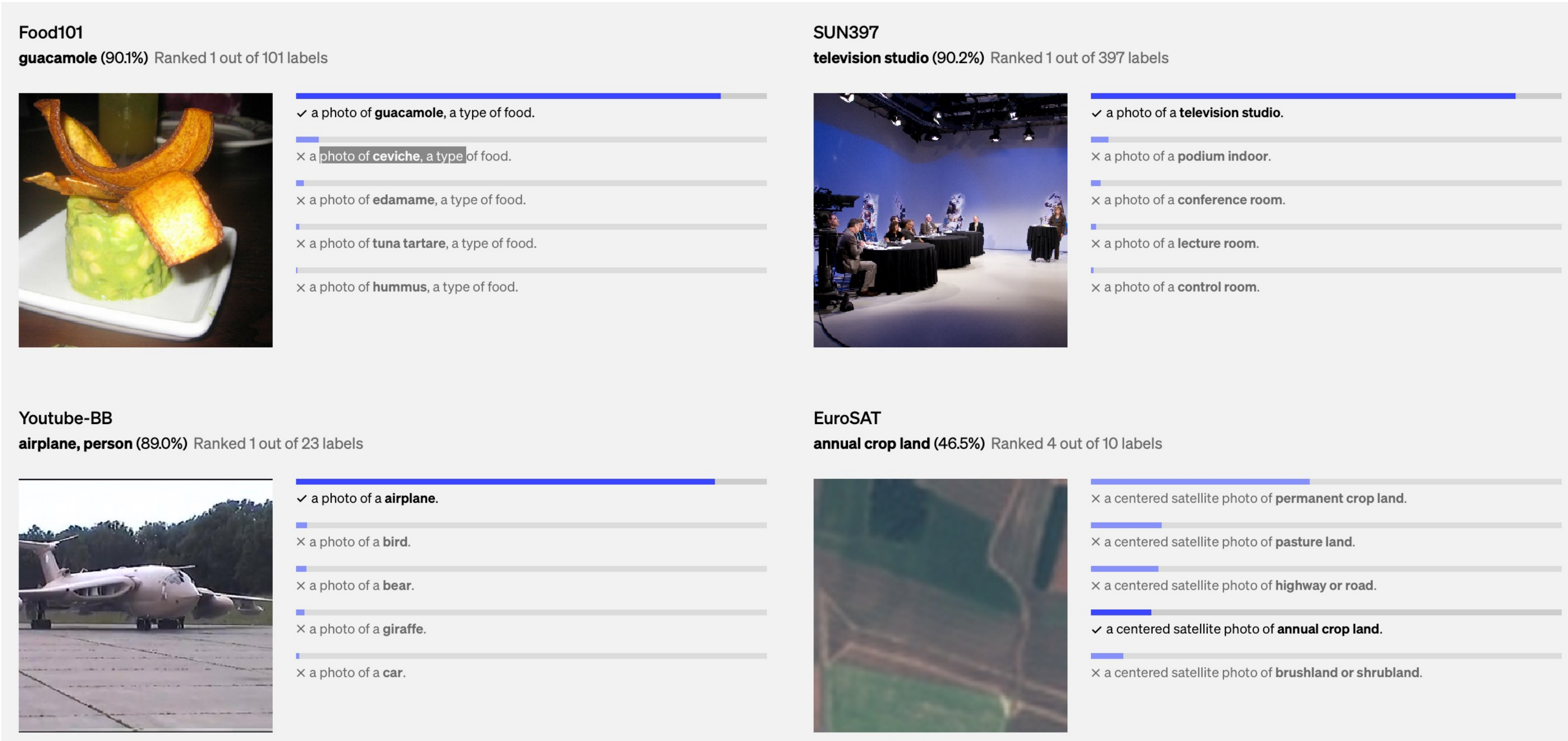
Semantic Spaces – Contrastive Language-Image Pre-training (**CLIP**)

Test phase



Radford et al., 2021

Semantic Spaces – Contrastive Language-Image Pre-training (**CLIP**)



https://openai.com/research/clip

Semantic Spaces – Summary

- **Latent Semantic Analysis (LSA)**
  - Shallow Learning Approach: BoW & truncated SVD, Simple and efficient
  - Feasible extensions for Multimodal LSA: learns from both image and text data
  - Abstract Concepts: represented with linear mixtures of words
  - Unconstrained Weights: Weights could be negative, impacting interpretation
  - Limited Semantic Interpretation: Topics may lack semantic meaning

- **Contrastive Language-Image Pre-training (CLIP)**
  - Deep Learning Approach: Deep Networks, data- and compute-hungry
  - Multimodal Understanding: Learns from both image and text data
  - Abstract Concepts: Captures complex semantic relationships
  - Challenges in Interpretability: deep learning nature may hinder interpretability