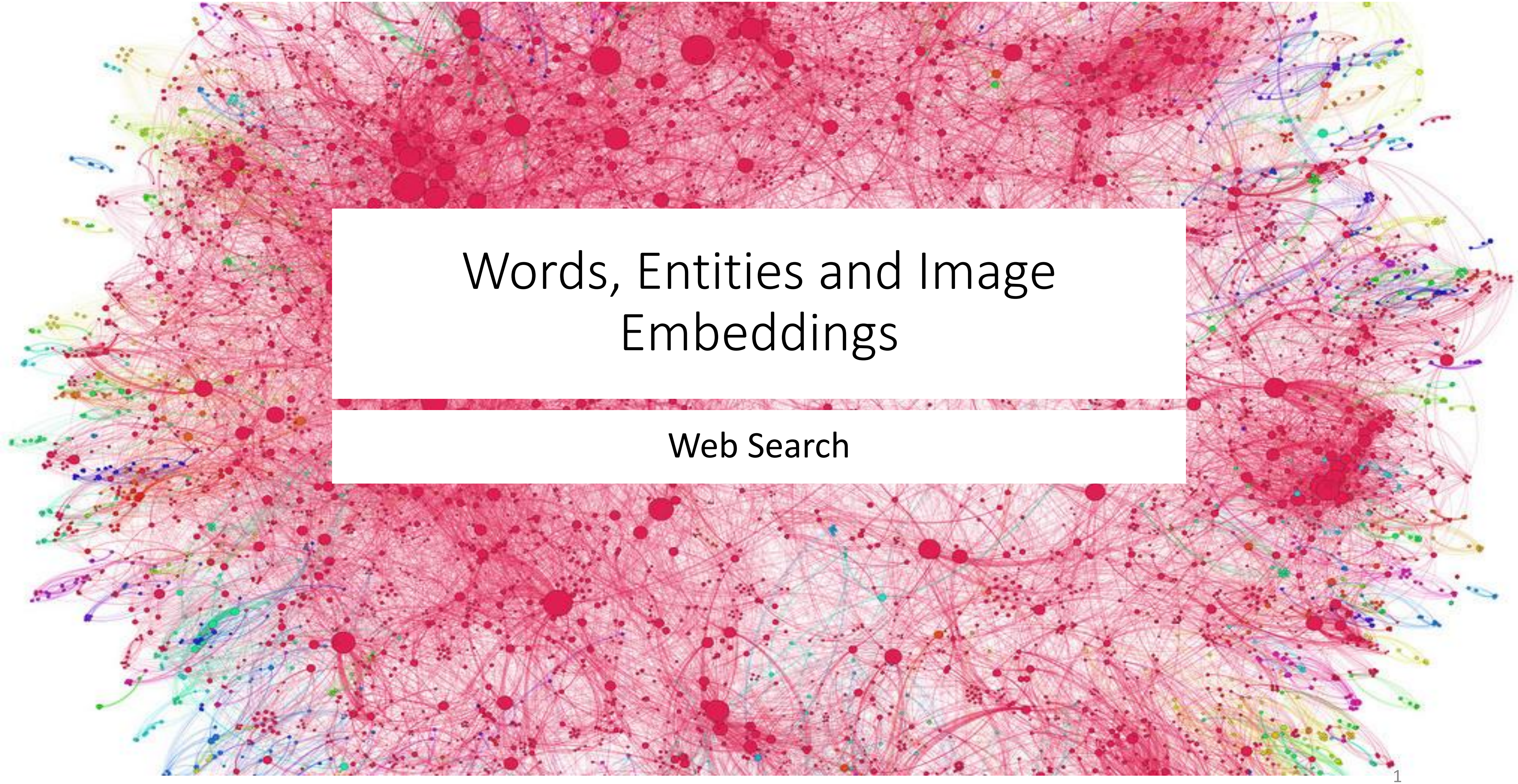


Words, Entities and Image Embeddings

Web Search

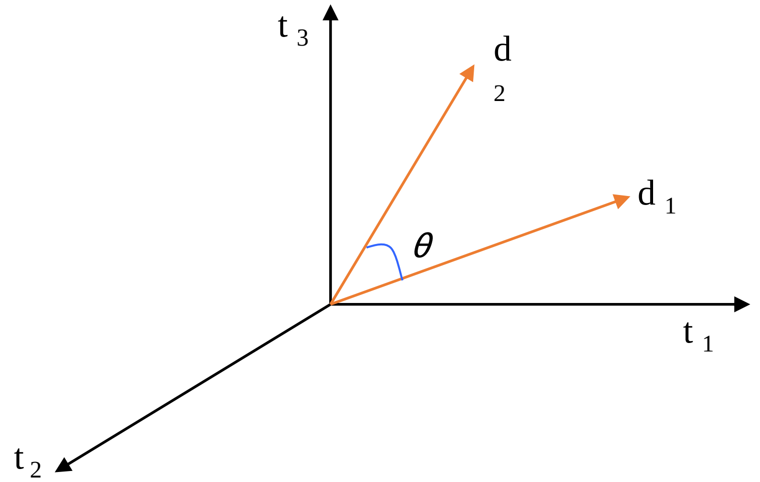
How to represent a word?

dog	1	$[1\quad 0\quad 0\quad 0\quad 0\quad 0\quad 0\quad 0\quad 0\quad 0]$
cat	2	$[0\quad 1\quad 0\quad 0\quad 0\quad 0\quad 0\quad 0\quad 0\quad 0]$
person	3	$[0\quad 0\quad 1\quad 0\quad 0\quad 0\quad 0\quad 0\quad 0\quad 0]$

Slides from Transactions on ACL: Omer Levy, Yoav Goldberg, Ido Dagan,
Improving Distributional Similarity with Lessons Learned from Word Embeddings, Bar-Ilan University, Israel
https://levyomer.wordpress.com/2015/03/30/improving-distributional-similarity-with-lessons-learned-from-word-embeddings/

Vector space model

- In the vector space model,
  each dimension corresponds to a term.



- The dimensionality V of the space corresponds to the size of the vocabulary.

- Each word is represented by a V dimensional vector, where only the dimension corresponding to that word is non-zero.

- Hence, each document is represented by the frequency of its terms.

How to represent a word?

dog	1	$[1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]$
cat	2	$[0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]$
person	3	$[0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]$

- **Problem**: distance between words using one-hot encodings always the same
- **Idea**: Instead of one-hot-encoding use a histogram of commonly co-occurring words.

Distributional Semantics

	food	walks	window	runs	mouse	invented	legs	sleeps	mirror	tail	…
dog	[5	5	0	5	0	0	5	5	0	2	...]
cat	[5	4	1	4	2	0	3	4	0	3	...]
person	[5	5	1	5	0	2	5	5	0	0	...]

This vocabulary can be extremely large

Distributional Semantics

- How similar is **pizza** to **pasta**?
- How related is **pizza** to **Italy**?

- **Representing words as vectors allows easy computation of similarity and relatedness.**

Approaches for Representing Words

Distributional Semantics (*Count*)

- Used since the 90's
- Sparse word-context PMI/PPMI matrix
- Decomposed with SVD

Word Embeddings (*Predict*)

- Inspired by deep learning
- `word2vec` *(Mikolov et al., 2013)*
- GloVe *(Pennington et al., 2014)*

Underlying Theory: **The Distributional Hypothesis** *(Harris, '54; Firth, '57)*

"Similar words occur in similar contexts"

Approaches for Representing Words

Both approaches:

- Rely on the **same linguistic theory**
- Use the **same data**
- Are **mathematically related**
  - "Neural Word Embedding as Implicit Matrix Factorization" (NIPS 2014)

- How come word embeddings are so much better?
  - "Don't Count, Predict!" (Baroni et al., ACL 2014)

- **More than meets the eye…**

Word Embeddings with Word2Vec

Algorithms

(objective + training method)

- Skip Grams + Negative Sampling
- CBOW + Hierarchical Softmax
- Noise Contrastive Estimation
- GloVe
- …

Hyperparameters

(preprocessing, smoothing, etc.)

- Subsampling
- Dynamic Context Windows
- Context Distribution Smoothing
- Adding Context Vectors
- …

What is `word2vec`?

- `word2vec` is **not** a single algorithm
- It is a **software package** for representing words as vectors, containing:
  - Two distinct models
    - CBoW
    - Skip-Gram
  - Various training methods
    - Negative Sampling
    - Hierarchical Softmax
  - A rich preprocessing pipeline
    - Dynamic Context Windows
    - Subsampling
    - Deleting Rare Words

What is `word2vec`?

- `word2vec` is **not** a single algorithm
- It is a **software package** for representing words as vectors, containing:
  - Two distinct models
    - CBoW
    - **Skip-Gram** **(SG)**
  - Various training methods
    - **Negative Sampling** **(NS)**
    - Hierarchical Softmax
  - A rich preprocessing pipeline
    - **Dynamic Context Windows** **(DCW)**
    - Subsampling
    - Deleting Rare Words

Skip-Grams with Negative Sampling (SGNS)

Marco saw a furry little cat hiding in the tree.

“word2vec Explained…”
Goldberg & Levy, arXiv 2014

Skip-Grams with Negative Sampling (SGNS)

Marco saw a furry little cat hiding in the tree.

"word2vec Explained…"
Goldberg & Levy, arXiv 2014

Skip-Grams with Negative Sampling (SGNS)

Marco saw a furry little cat hiding in the tree.

<u>**words**</u>	<u>**contexts**</u>
cat	furry
cat	little
cat	hiding
cat	in
…	…

$\left.\right\}$ D (data)

“word2vec Explained…”
Goldberg & Levy, arXiv 2014

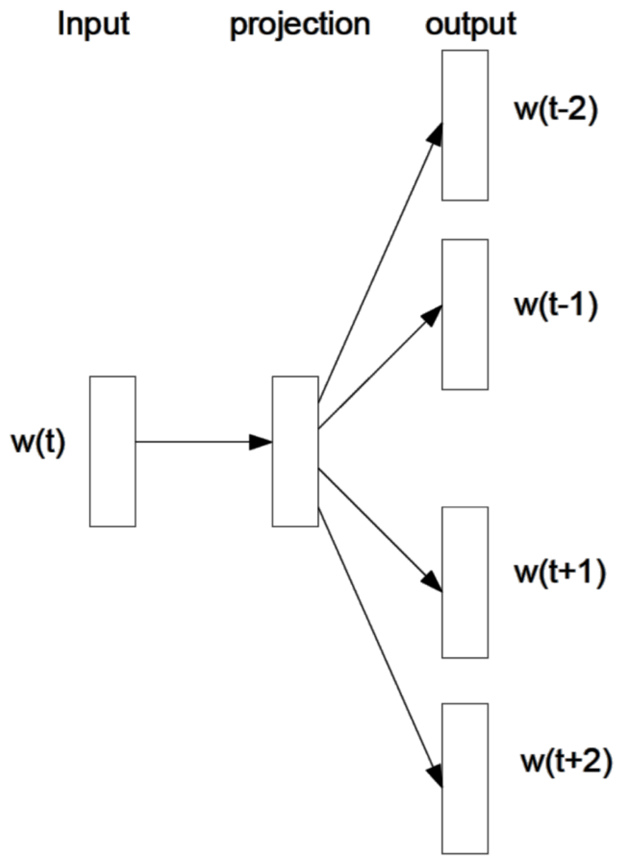
Skip-Grams with Negative Sampling (SGNS)

Marco saw a furry little cat hiding in the tree.

- Word2vec models the distribution of words and context words.

- The model will maximize the log-likelihood:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c,j\neq 0}\log p(w_{t+j}|w_t)$$



Softmax: words vs context words

- The $p(w_t|w_{t-1})$ is formalized as the softmax:

$$p(w_O|w_I) = \frac{\exp\left({v'_{w_O}}^\top v_{w_I}\right)}{\sum_{w=1}^{W} \exp\left({v'_w}^\top v_{w_I}\right)}$$

where each word is represented by a vector $v_{w_*} = [v_{w*} \quad \cdots \quad v_{w*}]$, thus rendering the argument of the softmax function:

Marco saw a furry little cat hiding in the tree.

$$v_{w_O}^T \cdot v_{w_I} = \boxed{[v_{w_{O,1}} \quad \cdots \quad v_{w_{O,n}}]} \cdot \boxed{\begin{bmatrix} v_{w_{I,1}} \\ \cdots \\ v_{w_{I,n}} \end{bmatrix}}$$

$$p(furry|cat) = \frac{\exp(\boxed{v_{furry_O}^T} \cdot \boxed{v_{cat_I}})}{\sum_{w_o} \exp(v_{w_o}^T \cdot \boxed{v_{cat_I}})}$$

Stochastic Gradient Descent

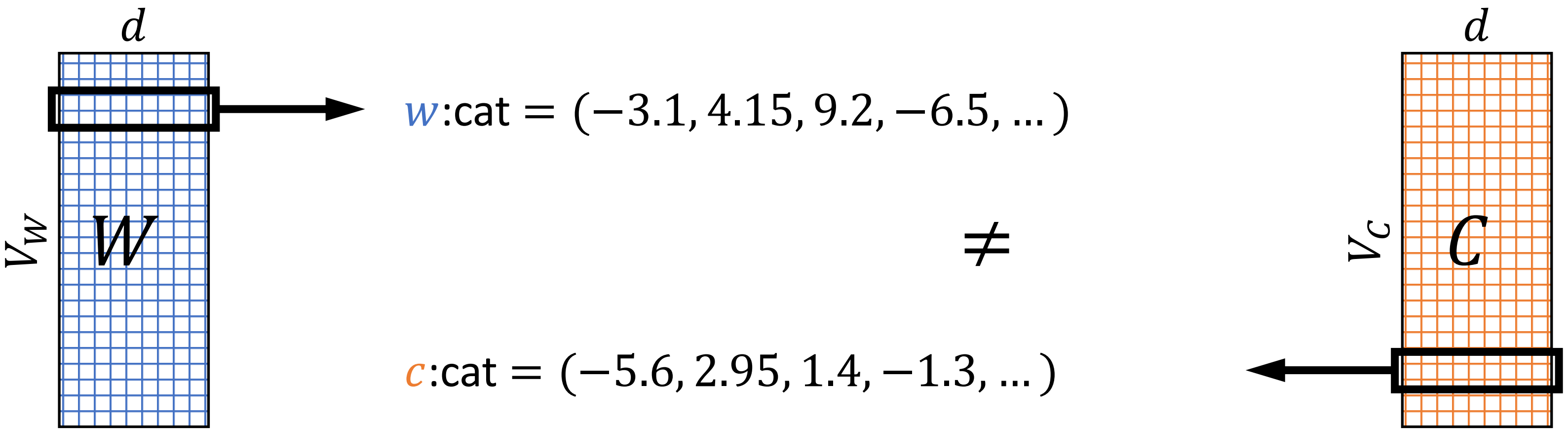
- But Corpus may have 40B tokens and windows
- You would wait a very long time before making a single update!
- **Very** bad idea for pretty much all neural nets!
- Instead: We will update parameters after each window t
  → Stochastic gradient descent (SGD)

$$v_{w_O}^{new} = v_{w_O}^{old} - \alpha \nabla_{v_{w_O}} p(w_o | w_I, D)$$

$$v_{w_I}^{new} = v_{w_I}^{old} - \alpha \nabla_{v_{w_I}} p(w_o | w_I, D)$$

Word vectors

- SGNS finds a vector $\vec{w}$ for each word w in our vocabulary V_W
- Each such vector has d latent dimensions (e.g. $d = 100$)
- Effectively, it learns a matrix W whose rows represent V_W
- **Key point:** it also learns a similar auxiliary matrix C of context vectors
- In fact, each word has two embeddings



"word2vec Explained…"
Goldberg & Levy, arXiv 2014

Positive Samples + Negative Sampling

- **Maximize:** $\sigma(\vec{w} \cdot \vec{c})$
  - c was **observed** with w

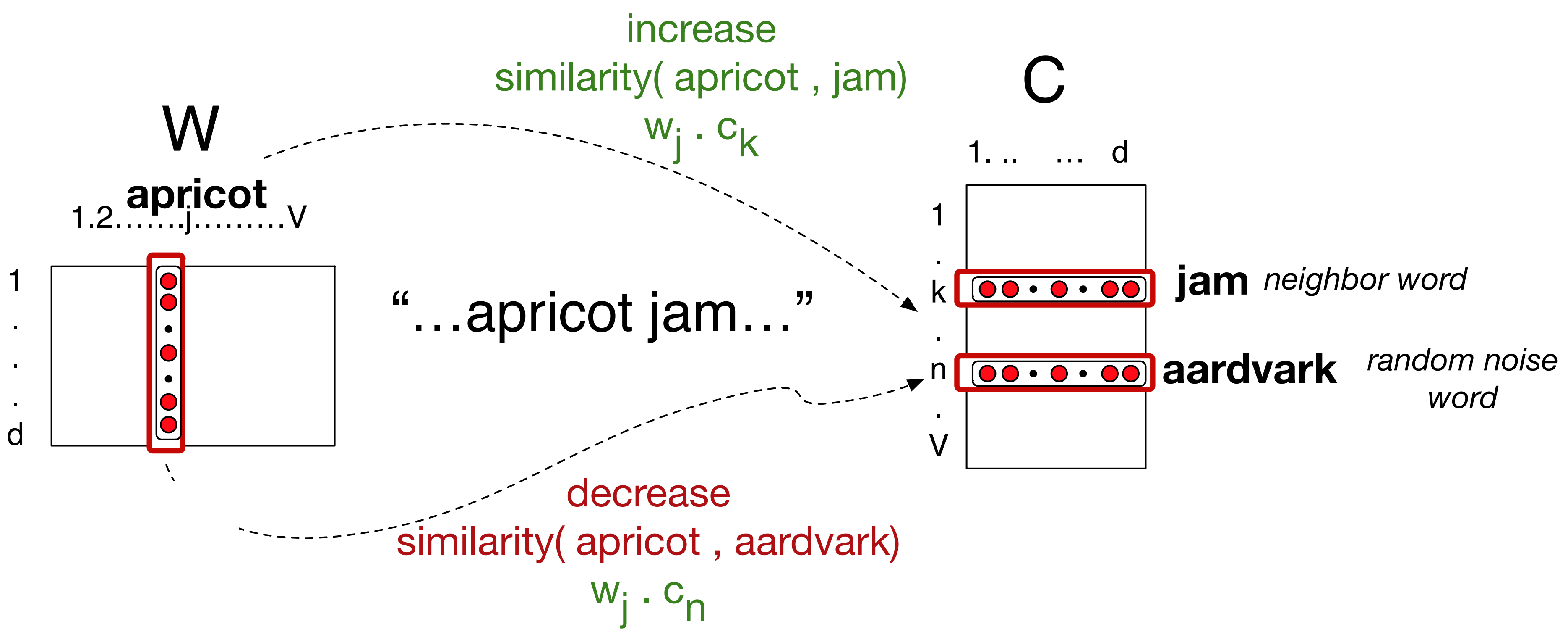
words	contexts
cat	furry
cat	little
cat	hiding
cat	in

- **Minimize:** $\sigma(\vec{w} \cdot \vec{c}')$
  - c' was **hallucinated** with w

words	contexts
cat	Australia
cat	cyber
cat	the
cat	1985

"word2vec Explained…"
Goldberg & Levy, arXiv 2014

Learning word vectors



Exercise: How would $\vec{w} \cdot \vec{c}$ be for these words?

"word2vec Explained…"
Goldberg & Levy, arXiv 2014

Implementation example

Hyperparameters

- **Preprocessing** **(word2vec)**
  - Dynamic Context Windows
  - Subsampling
  - Deleting Rare Words

- **Association Metric** **(SGNS)**
  - Shifted PMI
  - Context Distribution Smoothing

Dynamic Context Windows

Marco saw a furry little cat hiding in the tree.

Dynamic Context Windows

saw a furry little cat hiding in the tree

Dynamic Context Windows

saw a furry little cat hiding in the tree

word2vec:	$\frac{1}{4}$	$\frac{2}{4}$	$\frac{3}{4}$	$\frac{4}{4}$		$\frac{4}{4}$	$\frac{3}{4}$	$\frac{2}{4}$	$\frac{1}{4}$
GloVe:	$\frac{1}{4}$	$\frac{1}{3}$	$\frac{1}{2}$	$\frac{1}{1}$		$\frac{1}{1}$	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{1}{4}$
Aggressive:	$\frac{1}{8}$	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{1}$		$\frac{1}{1}$	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$

The Word-Space Model *(Sahlgren, 2006)*

Context Distribution Smoothing

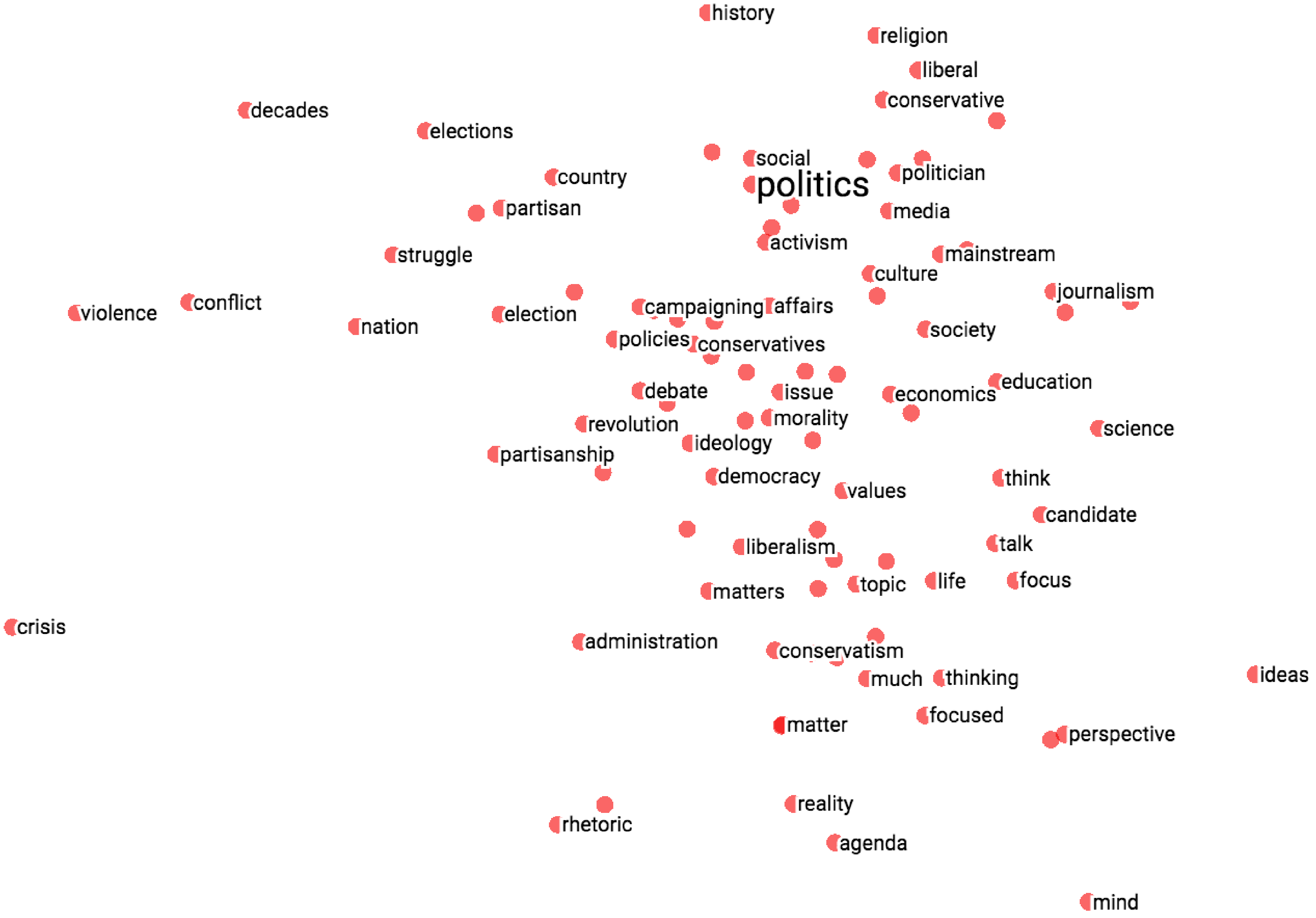
- SGNS samples $c' \sim P$ to form **negative** (w, c') examples

- Our analysis assumes P is the unigram distribution $P(c) = \frac{\#c}{\sum_{c' \in V_C} \#c'}$

- In practice, it's a **smoothed** unigram distribution

$$P^{0.75}(c) = \frac{(\#c)^{0.75}}{\sum_{c' \in V_C} (\#c')^{0.75}}$$

- This little change makes a big difference.



Slides from: Christopher Manning, Distributed word representations for IR, Stanford University, USA

Linear Relationships in word2vec

These representations are *very good* at encoding similarity and dimensions of similarity!

- Analogies testing dimensions of similarity can be solved quite well just by doing vector subtraction in the embedding space

  Syntactically

  - $x_{apple} - x_{apples} \approx x_{car} - x_{cars} \approx x_{family} - x_{families}$
  - Similarly for verb and adjective morphological forms

  Semantically (Semeval 2012 task 2)

  - $x_{shirt} - x_{clothing} \approx x_{chair} - x_{furniture}$
  - $x_{king} - x_{man} \approx x_{queen} - x_{woman}$

Slides from: Christopher Manning, Distributed word representations for IR, Stanford University, USA

Word Analogies

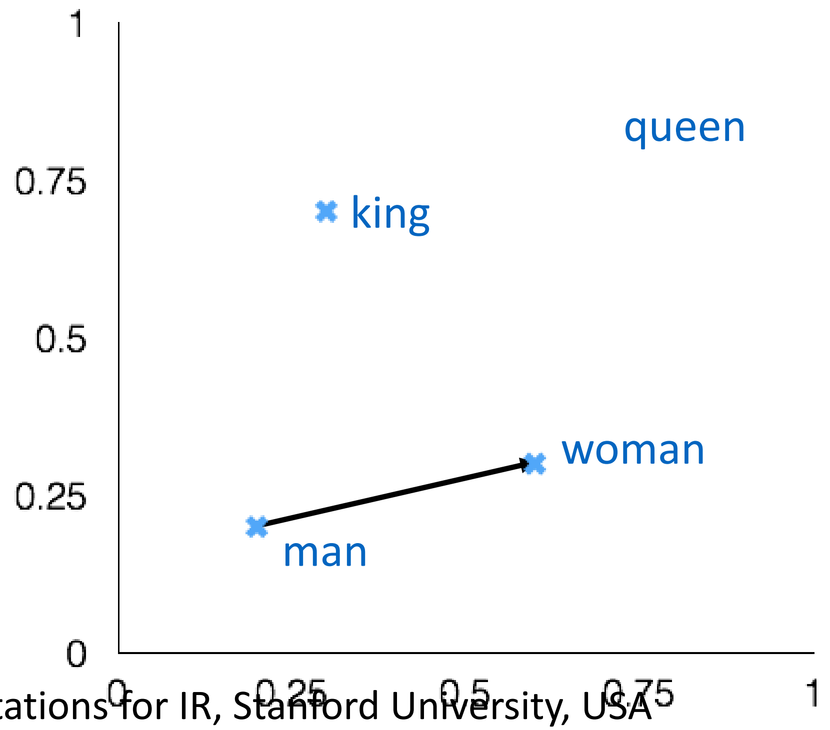
Test for linear relationships, examined by Mikolov et al.

a:b :: c:?

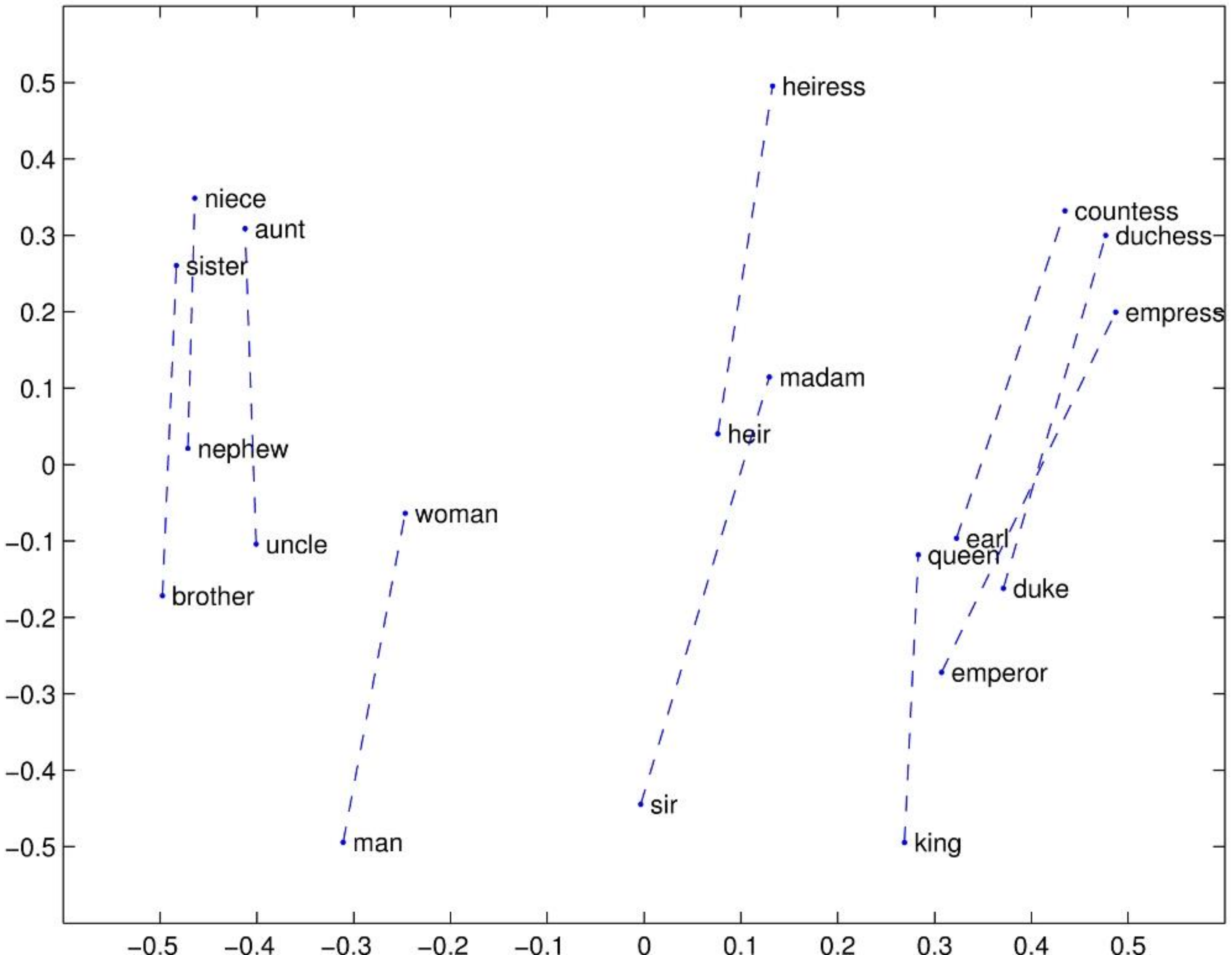
$$d = \arg\max_x \frac{(w_b - w_a + w_c)^T w_x}{||w_b - w_a + w_c||}$$

man:woman :: king:?

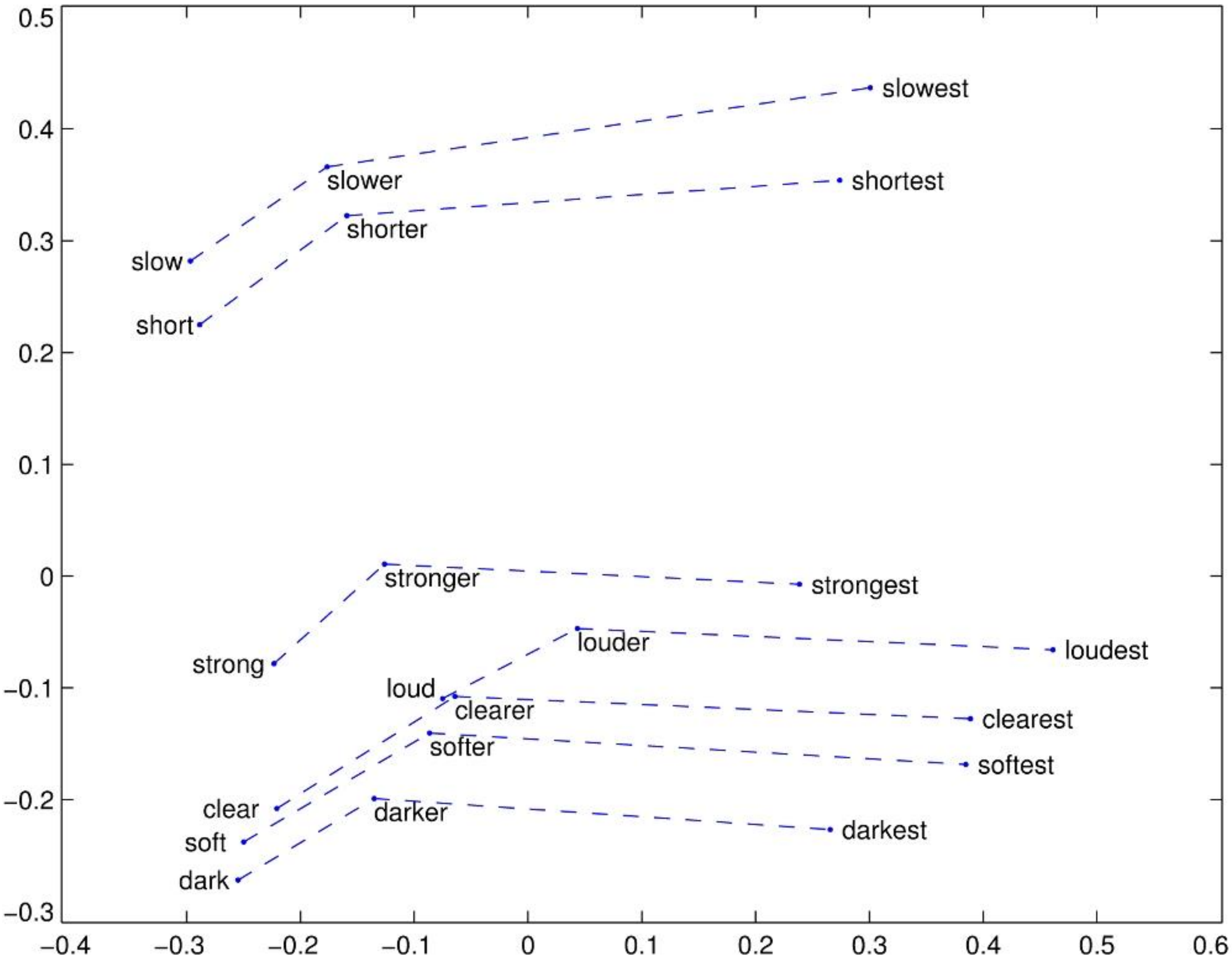
+	king	[0.30 0.70]
−	man	[0.20 0.20]
+	woman	[0.60 0.30]
	queen	[0.70 0.80]



Slides from: Christopher Manning, Distributed word representations for IR, Stanford University, USA



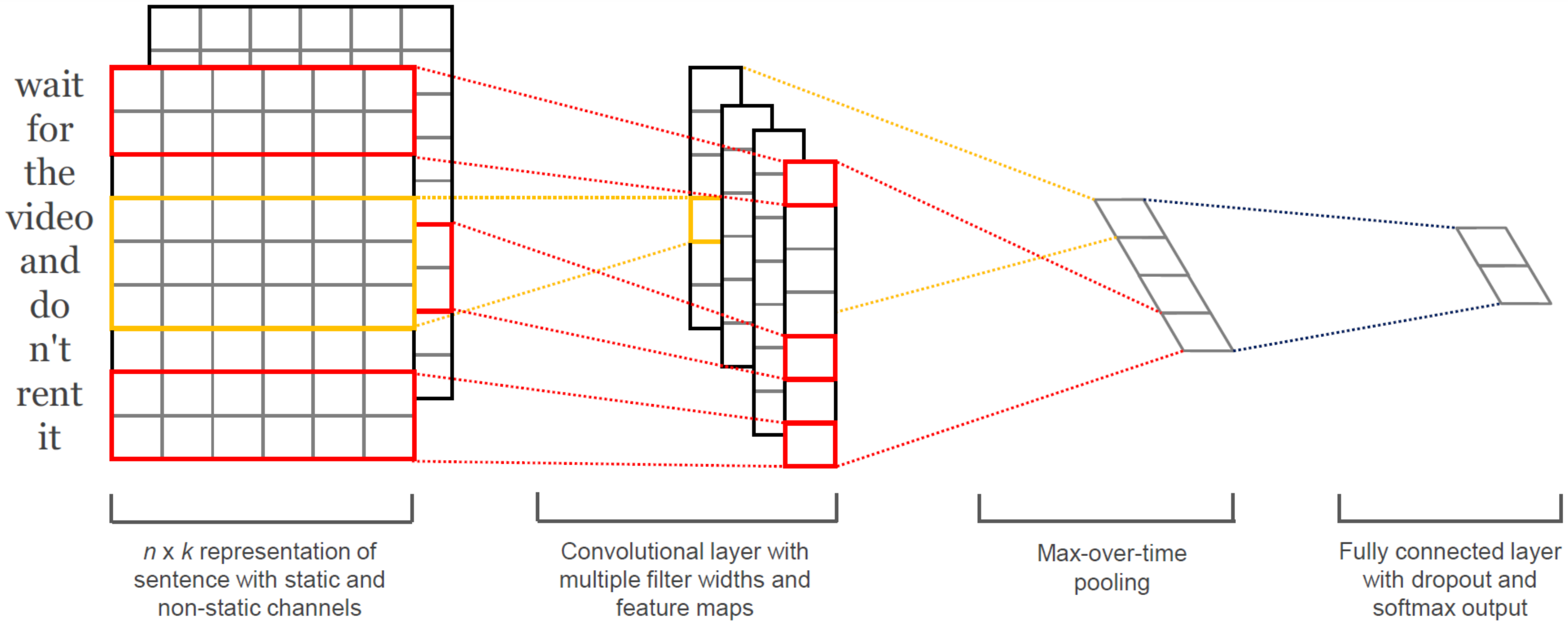
heiress
niece
aunt
sister
countess
duchess
empress
madam
heir
nephew
woman
uncle
earl
queen
duke
brother
emperor
sir
man
king
0.5
0.4
0.3
0.2
0.1
0
−0.1
−0.2
−0.3
−0.4
−0.5
−0.5
−0.4
−0.3
−0.2
−0.1
0
0.1
0.2
0.3
0.4
0.5


slowest
slower
shortest
shorter
slow
short
stronger
strongest
louder
loudest
strong
loud
clearer
clearest
softer
softest
clear
darker
darkest
soft
dark
0.5
0.4
0.3
0.2
0.1
0
−0.1
−0.2
−0.3
−0.4
−0.3
−0.2
−0.1
0
0.1
0.2
0.3
0.4
0.5
0.6

Examples

- Sentiment classification
- Word-Entity embeddings
  - Link prediction
- Word-Image embeddings
  - Zero-shot image classification
- Multimodal relation embeddings

Word embeddings and CNNs for sentence classification



Kim, Yoon. "Convolutional Neural Networks for Sentence Classification." *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2014.

Results

- Movie reviews (MR)
- Sentiment analysis (SST-1, SST-2, Sub)
- TREC Question Answering (TREC)
- Customer reviews (CR)
- Opinion polarity (MPQA)

Data	c	l	N	$\lvert V \rvert$	$\lvert V_{pre} \rvert$	*Test*
MR	2	20	10662	18765	16448	CV
SST-1	5	18	11855	17836	16262	2210
SST-2	2	19	9613	16185	14838	1821
Subj	2	23	10000	21323	17913	CV
TREC	6	10	5952	9592	9125	500
CR	2	19	3775	5340	5046	CV
MPQA	2	3	10606	6246	6083	CV

Results

Model	MR	SST-1	SST-2	Subj	TREC	CR	MPQA
CNN-rand	76.1	45.0	82.7	89.6	91.2	79.8	83.4
CNN-static	81.0	45.5	86.8	93.0	92.8	84.7	**89.6**
CNN-non-static	**81.5**	48.0	87.2	93.4	93.6	84.3	89.5
CNN-multichannel	81.1	47.4	**88.1**	93.2	92.2	**85.0**	89.4
RAE (Socher et al., 2011)	77.7	43.2	82.4	–	–	–	86.4
MV-RNN (Socher et al., 2012)	79.0	44.4	82.9	–	–	–	–
RNTN (Socher et al., 2013)	–	45.7	85.4	–	–	–	–
DCNN (Kalchbrenner et al., 2014)	–	48.5	86.8	–	93.0	–	–
Paragraph-Vec (Le and Mikolov, 2014)	–	**48.7**	87.8	–	–	–	–
CCAE (Hermann and Blunsom, 2013)	77.8	–	–	–	–	–	87.2
Sent-Parser (Dong et al., 2014)	79.5	–	–	–	–	–	86.3
NBSVM (Wang and Manning, 2012)	79.4	–	–	93.2	–	81.8	86.3
MNB (Wang and Manning, 2012)	79.0	–	–	**93.6**	–	80.0	86.3
G-Dropout (Wang and Manning, 2013)	79.0	–	–	93.4	–	82.1	86.1
F-Dropout (Wang and Manning, 2013)	79.1	–	–	**93.6**	–	81.9	86.3
Tree-CRF (Nakagawa et al., 2010)	77.3	–	–	–	–	81.4	86.1
CRF-PR (Yang and Cardie, 2014)	–	–	–	–	–	82.7	–
SVM_S (Silva et al., 2011)	–	–	–	–	**95.0**	–	–

Wikipedia embeddings

- Wikipedia is the largest source of unstructured knowledge.
- Embeddings can be learned from Wikipedia data:
  - Words
  - Links
  - Entities

Word-based skip-gram model

The neighboring words of each word are used as contexts

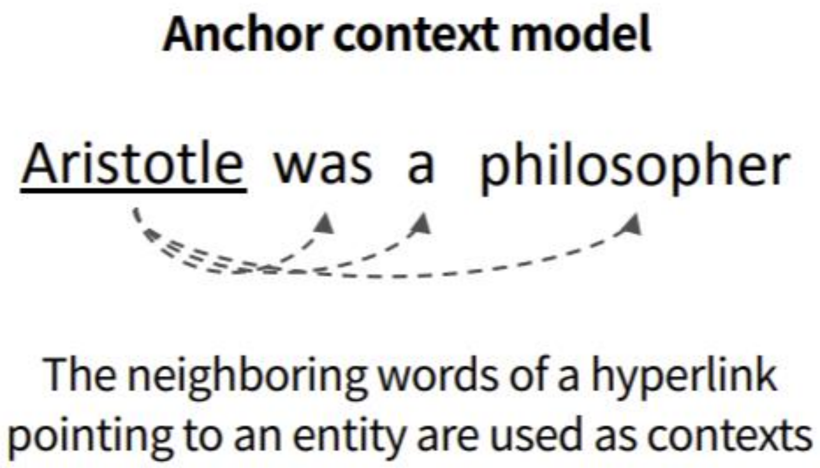
$$\mathcal{L}_w = -\sum_{i=1}^{N} \sum_{-c \le j \le c, j \ne 0} \log P(w_{i+j} | w_i)$$

Entity descriptions

- In Wikipedia a particular page can be seen as an entity (a named entity or a concept).

- The entity has description on its own page.

- Hence, we can learn which words are most associated to an entity.

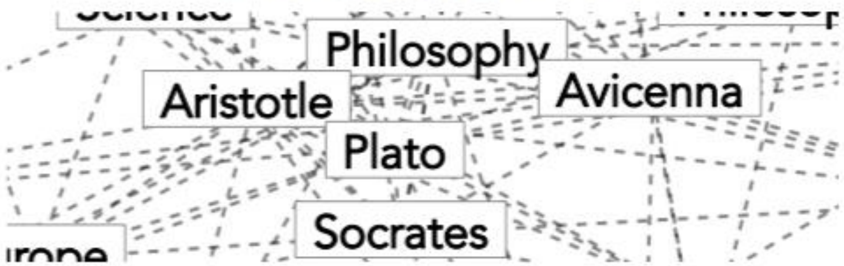


$$\mathcal{L}_a = -\sum_{(e_i,Q)\in A}\sum_{w_c\in Q}\log P(w_c|e_i)$$

Wikipedia link graph

- Finally, entities are linked among themselves creating a linked graph.
- An embedding of linked entities can be computed by predicting links between all entities.

Link graph model

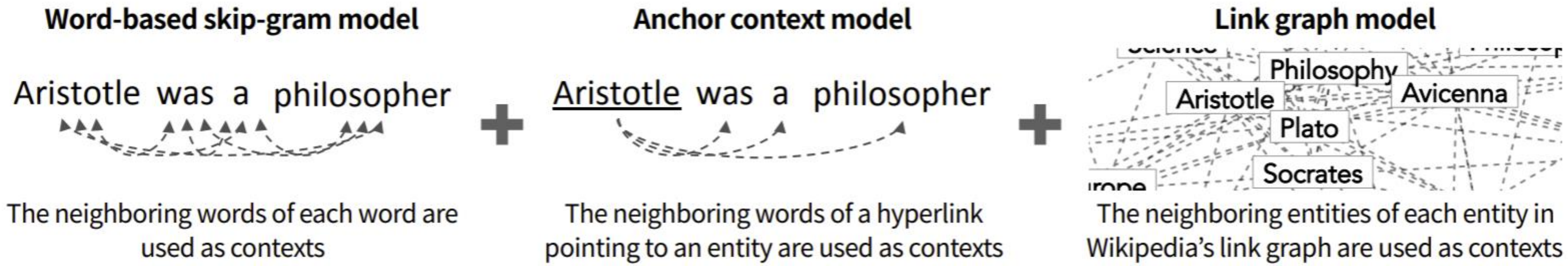


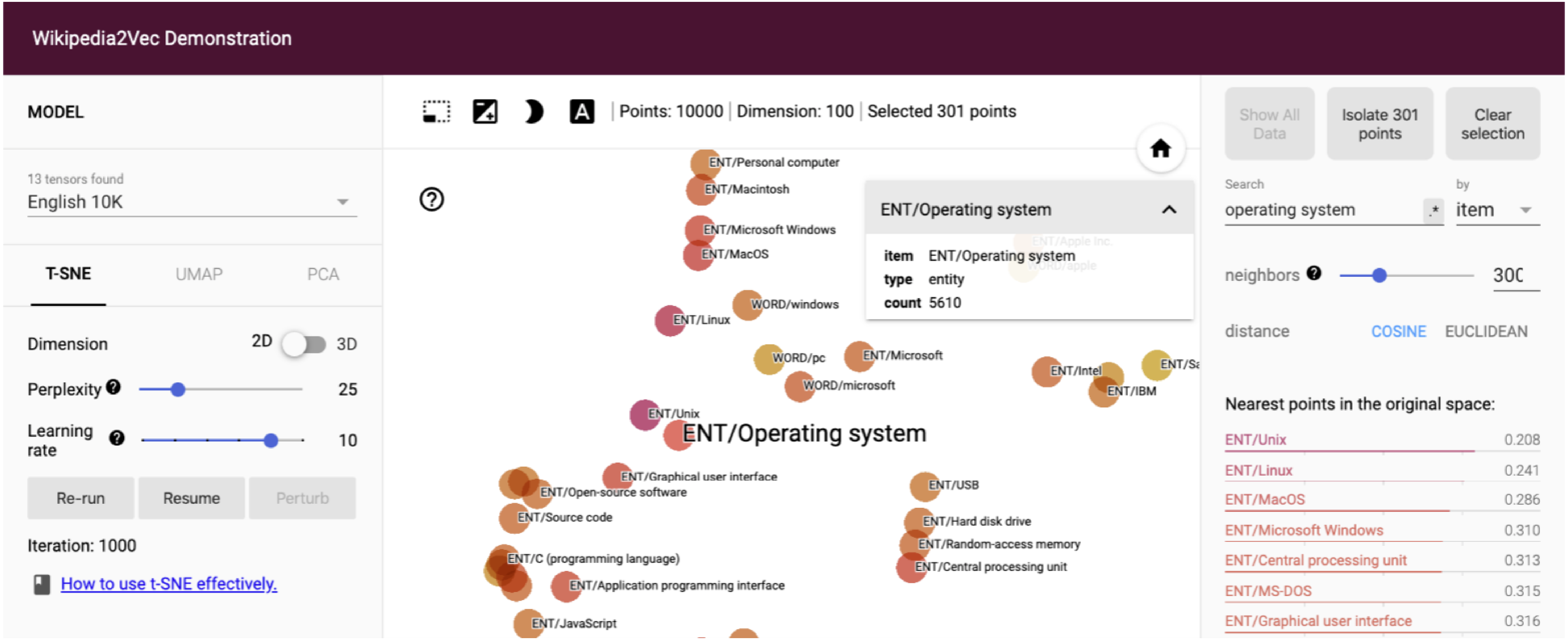
The neighboring entities of each entity in Wikipedia's link graph are used as contexts

$$\mathcal{L}_e = -\sum_{e_i \in E} \sum_{e_o \in C_{e_i}} \log P(e_o | e_i)$$

Wikipedia2vec

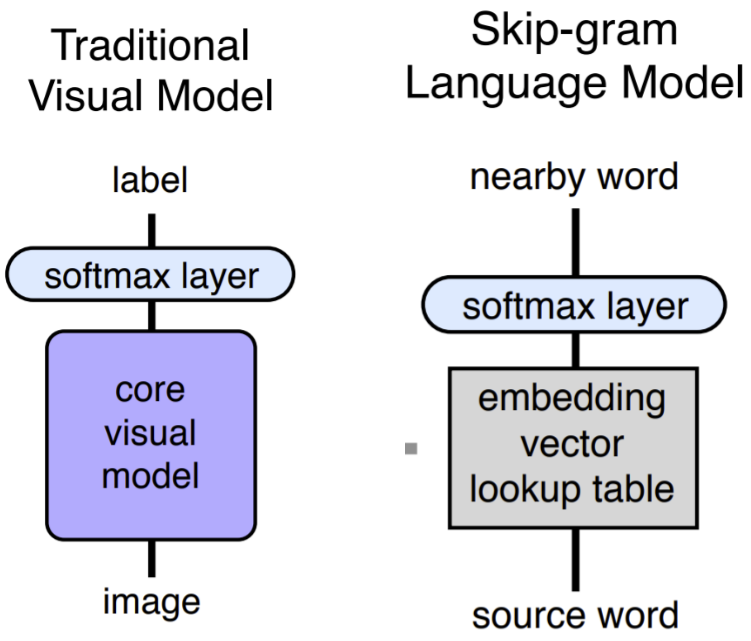
- Learning the three objectives simultaneously leads to the Wikipedia2vec embeddings.




Wikipedia2Vec Demonstration
MODEL
13 tensors found
English 10K
T-SNE
UMAP
PCA
Dimension
2D
3D
Perplexity
25
Learning rate
10
Re-run
Resume
Perturb
Iteration: 1000
How to use t-SNE effectively.
Points: 10000
Dimension: 100
Selected 301 points
ENT/Operating system
item ENT/Operating system
type entity
count 5610
ENT/Personal computer
ENT/Macintosh
ENT/Microsoft Windows
ENT/MacOS
WORD/windows
ENT/Linux
WORD/pc
ENT/Microsoft
WORD/microsoft
ENT/Intel
ENT/IBM
ENT/Unix
ENT/Operating system
ENT/Graphical user interface
ENT/Open-source software
ENT/Source code
ENT/USB
ENT/Hard disk drive
ENT/Random-access memory
ENT/Central processing unit
ENT/C (programming language)
ENT/Application programming interface
ENT/JavaScript
Show All Data
Isolate 301 points
Clear selection
Search
operating system
by
item
neighbors
distance
COSINE
EUCLIDEAN
Nearest points in the original space:
ENT/Unix 0.208
ENT/Linux 0.241
ENT/MacOS 0.286
ENT/Microsoft Windows 0.310
ENT/Central processing unit 0.313
ENT/MS-DOS 0.315
ENT/Graphical user interface 0.316

Joint word-image embeddings

- Word embeddings
  - Skip-grams
- Image embeddings
  - inner representations of VGG-16 or ResNet.

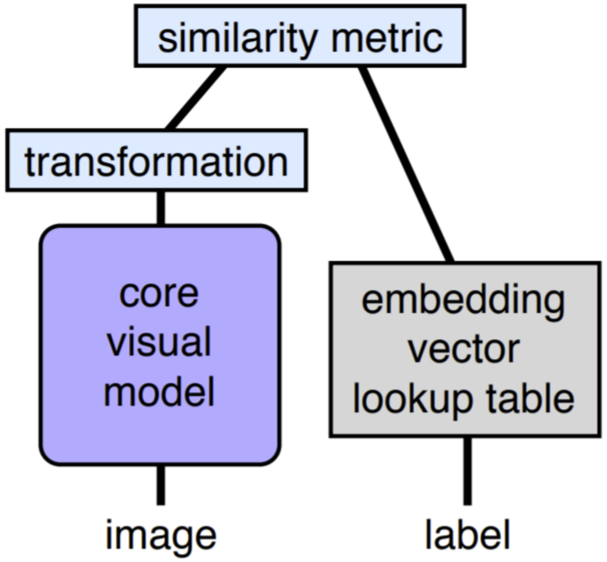


- A joint word-image model would allow to cross between the visual domain and the language domain easily.

DeViSE: Visual-Semantic Embeddings

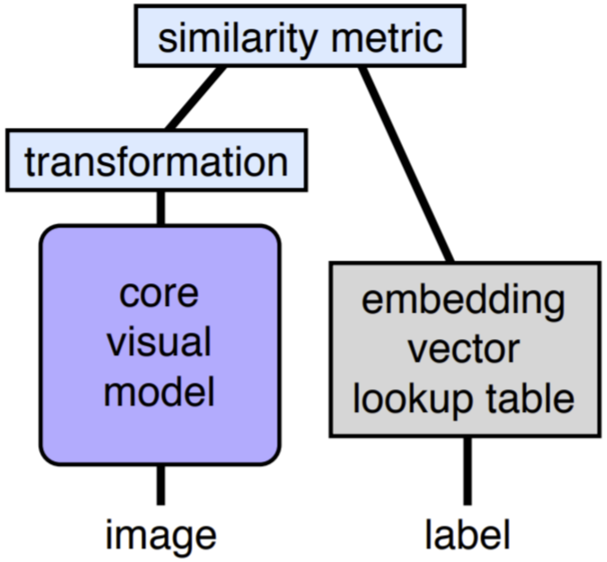
- Each **image** and its associated **words** encoded into a vector representation with a CNN model and W2V respectively.

- A transformation layer is added to the top of the CNN architecture



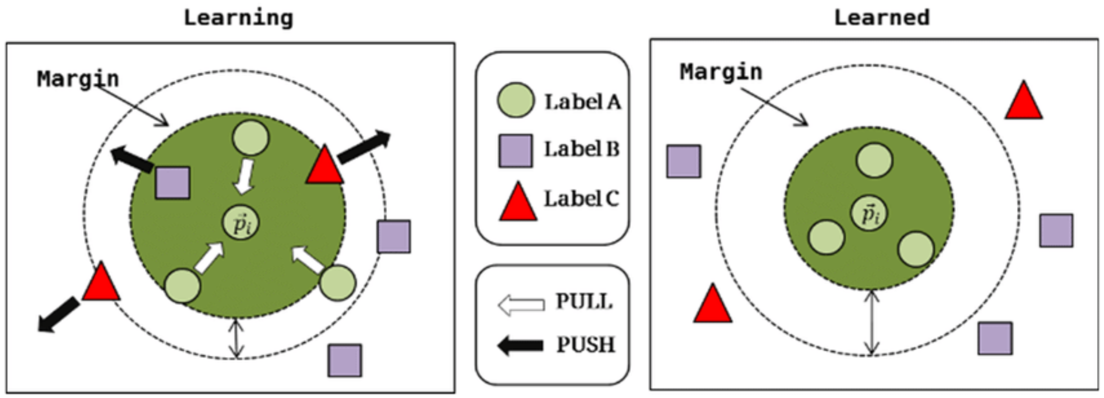
DeViSE: Visual-Semantic Embeddings

- The transformation layer is learned so that an image vector will be projected into the neighborhood of its word vectors.
- The goal is to **maximize the similarity between the image embeddings and its associated word embeddings**.



How to learn the multimodal embedding?

- The similarity(visual, word) is the 1-distance between the image embedding vector and the word embedding vector.



- The triplet loss is used to:
  - move the image embedding vector **closer** to its associated word vectors;
  - move the image embedding vector **away** from other word vectors.

$$loss(image, label) = \sum_{j \neq label} \max\{0, m - s(t_{label}, image) + s(t_j, image)\}$$

Key-insights

- We learned how to project images onto the word embedding space!
- Any image, of any known or unknown class, can now be projected onto this space!!
- This enables many other statistical reasoning tasks in the visual-linguistic embedding:
  - Zero-shot for search with unseen words
  - Visual-analogies
  - Image captioning
  - Visual QA
  - Visual-linguistic-entity embeddings

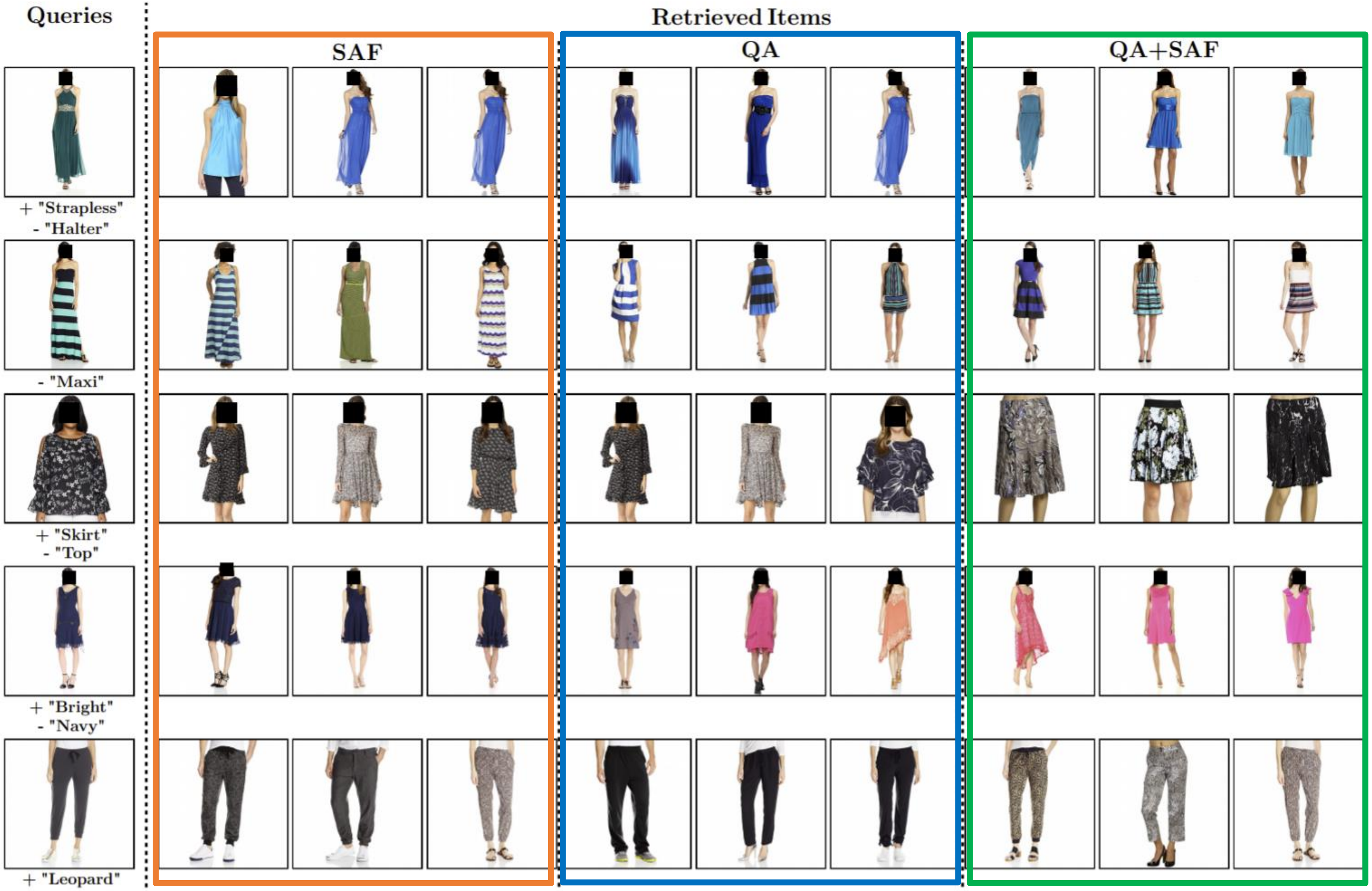
Zero-shot image predictions

	Our model	Softmax over ImageNet 1K
 **A**	eyepiece, ocular	typewriter keyboard
	Polaroid	tape player
	compound lens	reflex camera
	telephoto lens, zoom lens	CD player
	rangefinder, range finder	space bar
 **B**	oboe, hautboy, hautbois	reel
	bassoon	punching bag, punch bag, ...
	English horn, cor anglais	whistle
	hook and eye	bassoon
	hand	letter opener, paper knife, ...
 **C**	barbet	patas, hussar monkey, ...
	patas, hussar monkey, ...	proboscis monkey, Nasalis ...
	babbler, cackler	macaque
	titmouse, tit	titi, titi monkey
	bowerbird, catbird	guenon, guenon monkey

Zero-shot image predictions

		Our model	Softmax over ImageNet 1K
	D	fruit pineapple **pineapple plant, Ananas ...** sweet orange sweet orange tree, ...	pineapple, ananas coral fungus artichoke, globe artichoke sea anemone, anemone cardoon
	E	comestible, edible, ... dressing, salad dressing Sicilian pizza vegetable, veggie, veg fruit	pot, flowerpot cauliflower guacamole cucumber, cuke broccoli
	F	dune buggy, beach buggy searcher beetle, ... seeker, searcher, quester Tragelaphus eurycerus, ... bongo, bongo drum	warplane, military plane missile projectile, missile sports car, sport car submarine, pigboat, sub, ...

Relations as vector operators



Sadeh, G., Fritz, L., Shalev, G. and Oks, E., 2019. Joint visual-textual embedding for multimodal style search. *arXiv preprint arXiv:1906.06620.*

Summary

- Lots of applications wherever knowing word context or similarity helps prediction:
  - Synonym handling in search, Document aboutness, Ad serving, …
- Great tool for many Language and Vision search tasks:
  - Language models: from spelling correction to email response
  - Analogy, sentiment analysis, zero-shot predictions
- Readings (word embeddings):
  - Dan Jurafsky and James H. Martin, *Speech and Language Processing* (*3rd ed*. *draft*), Chapter 6 https://web.stanford.edu/~jurafsky/slp3/6.pdf

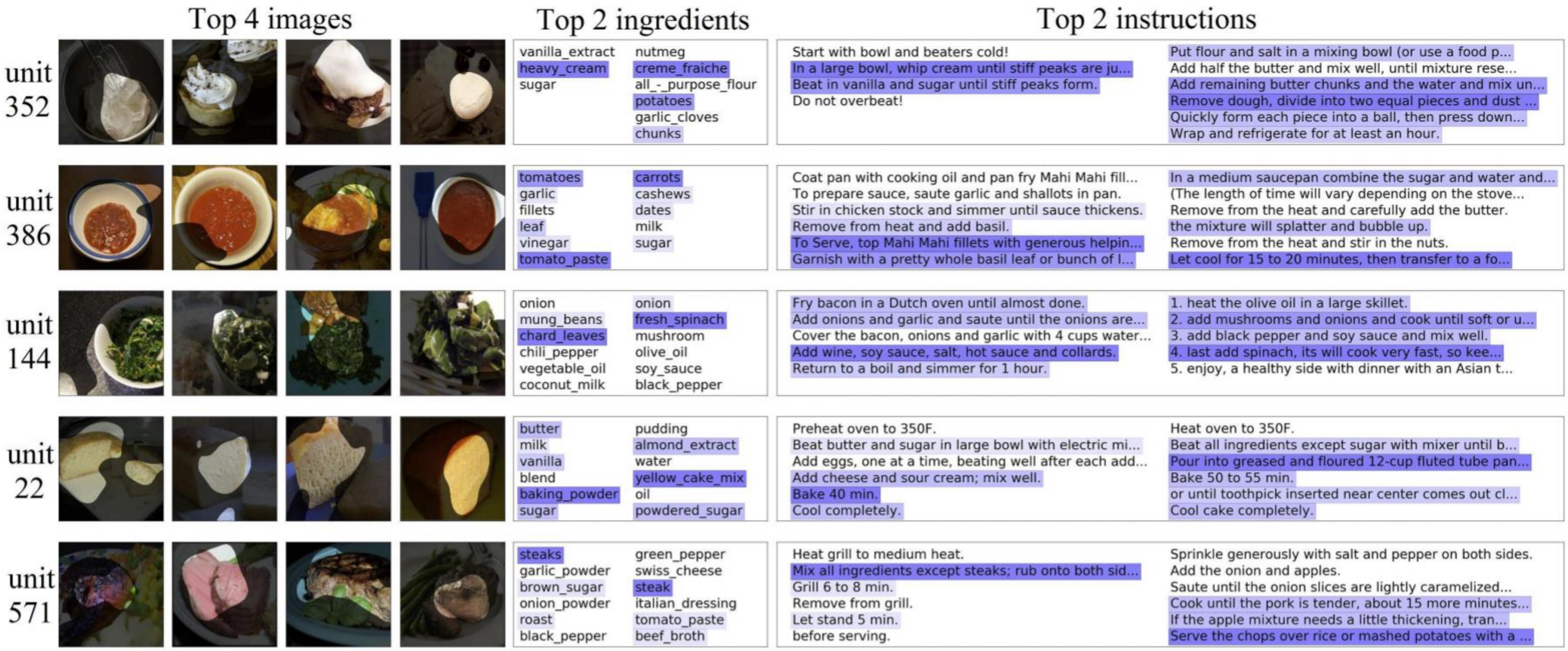
Paper references

- **Word2Vec:** Mikolov, Tomas, et al. "Distributed representations of words and phrases and their compositionality." *Advances in neural information processing systems*. 2013.
    - Omer Levy, Yoav Goldberg, Ido Dagan, Improving Distributional Similarity with Lessons Learned from Word Embeddings, Transactions of ACL, 2015

Examples:

- **CNN:** Kim, Yoon. "Convolutional Neural Networks for Sentence Classification." Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2014.
- **Wikipedia2vec:** Yamada I, Asai A, Sakuma J, Shindo H, Takeda H, Takefuji Y, Matsumoto Y. Wikipedia2vec: An efficient toolkit for learning and visualizing the embeddings of words and entities from wikipedia. arXiv preprint arXiv:1812.06280. 2018 Dec 15.
    - Yamada I, Shindo H, Takeda H, Takefuji Y. Joint Learning of the Embedding of Words and Entities for Named Entity Disambiguation. InProceedings of The 20th SIGNLL Conference on Computational Natural Language Learning 2016 Aug (pp. 250-259).
- **DeViSE:** Frome, Andrea, Gregory S. Corrado, Jonathon Shlens, Samy Bengio, Jeffrey Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. "DeViSE: A Deep Visual-Semantic Embedding Model." In NIPS. 2013.
- **Visual-Language Relations:** Sadeh, G., Fritz, L., Shalev, G. and Oks, E., 2019. Joint visual-textual embedding for multimodal style search. *arXiv preprint arXiv:1906.06620*.

Next week



Salvador, A., Hynes, N., Aytar, Y., Marin, J., Ofli, F., Weber, I. and Torralba, A., 2017. **Learning cross-modal embeddings for cooking recipes and food images**. In *Proceedings of the IEEE conference on computer vision and pattern recognition*

Marin, J., Biswas, A., Ofli, F., Hynes, N., Salvador, A., Aytar, Y., Weber, I. and Torralba, A., 2019. **Recipe1m+: A dataset for learning cross-modal embeddings for cooking recipes and food images**. *IEEE transactions on pattern analysis and machine intelligence*