ANLP Lecture 21
Distributional Semantics

Shay Cohen
(Based on slides by Henry Thompson and Sharon Goldwater)

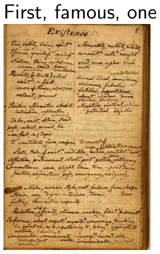
1 November 2019

Example Question (5)

- Question
  What is a good way to remove wine stains?
- Text available to the machine
  Salt is a great way to eliminate wine stains
- What is hard?
  - words may be related in other ways, including **similarity** and **gradation**
  - how to know if words have similar meanings?

Can we just use a thesaurus?

- A **thesaurus** is a synonym (and sometimes antonym) dictionary
  - Organised by a hierarchy of meaning classes
  - First, famous, one for English by Roget published in 1852



First edition entry for **Existence**: *Ens, entity, being, existence, essence...*
- WordNet is a super-thesaurus in digital form
- The next slide shows paired entries
  - One from the original English version
  - One from a Chinese version

07200527-n (12)	answer 回答	the speech act of replying to a question
06746005-n (56)	answer, reply, response 答复, 回答	a statement (either spoken or written) that is made to reply to a question or request or criticism or accusation
00636279-v (7) V2	answer 解决, 回答	give the correct answer or solution to
00815686-v (123) V1, V2	answer, reply, respond 答应, 答复, 回, 答覆 , 响应, 回答	react verbally

Extract from Open Multilingual Wordnet 1.2 from results of searching for *answer* in English and Chinese (simplified).

Problems with thesauri/Wordnet

Not every language has a thesaurus
Even for the ones that we do have, many words and phrases will be missing
So, let's try to compute similarity automatically

- Context is the key

Meaning from context(s)

- Consider the example from J&M (quoted from earlier sources):

  a bottle of *tezgüino* is on the table
  everybody likes *tezgüino*
  tezgüino makes you drunk
  we make *tezgüino* out of corn

Distributional hypothesis

- Perhaps we can infer meaning just by looking at the contexts a word occurs in
- Perhaps meaning IS the contexts a word occurs in (!)
- Either way, similar contexts imply similar meanings:
  - This idea is known as the **distributional hypothesis**

"Distribution": a polysemous word

- Probability distribution: a function from outcomes to real numbers
- Linguistic distribution: the set of contexts that a particular item (here, word) occurs in
  - Sometimes displayed in **Keyword In Context** (KWIC) format:

category error was partly the	answer	to the uncouth question, since
Leg was governor, and the	answer	was "one Leg", and the
But Greg knew he would	answer	his questions about anyone local
Trent didn't bother to	answer.	
not provide the sort of	answer	we want, we can always
we dismiss (5) with the	answer	"Yes we do"! Regarding
The	answer	is simple – speed up your
and so he'd always	answer	back and say I want
doing anything else is one	answer	often suggested.

Taken at random from the British National Corpus

Distributional semantics: basic idea

- Represent each word w_i as a vector of its contexts
  - distributional semantic models also called **vector-space models**
- Ex: each dimension is a context word; = 1 if it co-occurs with w_i, otherwise 0.

	pet	bone	fur	run	brown	screen	mouse	fetch
$w_1 =$	1	1	1	1	1	0	0	1
$w_2 =$	1	0	1	0	1	0	1	0
$w_3 =$	0	0	0	1	0	1	1	0

- Note: real vectors would be far more sparse

Questions to consider

- What defines "context"? (What are the dimensions, what counts as co-occurrence?)
- How to weight the context words (Boolean? counts? other?)
- How to measure similarity between vectors?

Defining the context

- Usually ignore **stopwords** (function words and other very frequent/uninformative words)
- Usually use a large window around the target word (e.g., 100 words, maybe even whole document)
- Can use just cooccurrence within window, or may require more (e.g., dependency relation from parser)
- Note: all of these for *semantic* similarity
  - For *syntactic* similarity, use a small window (1-3 words) and track *only* frequent words

How to weight the context words

- Binary indicators not very informative
- Presumably more frequent co-occurrences matter more
- But, is frequency good enough?
  - Frequent words are expected to have high counts in the context vector
  - Regardless of whether they occur more often with this word than with others

Collocations

- We want to know which words occur *unusually* often in the context of w: more than we'd expect by chance?
- Put another way, what **collocations** include w?

Mutual information

- One way: use **pointwise mutual information** (PMI):

$$\mathrm{PMI}(x, y) = \log_2 \frac{P(x, y)}{P(x)P(y)} \quad \begin{array}{l} \Leftarrow \text{ Observed probability of seeing words } x \text{ and } y \text{ together} \\ \Leftarrow \text{ Predicted probability of same, if } x \text{ and } y \text{ are independent} \end{array}$$

- PMI tells us how much more/less likely the cooccurrence is than if the words were independent

$= 0$	independent	as predicted
> 0	friends	occur together *more* than predicted
< 0	enemies	occur together *less* than predicted

A problem with PMI

- In practice, PMI is computed with counts (using MLE)
- Result: it is over-sensitive to the chance co-occurrence of infrequent words
- See next slide: ex. PMIs from bigrams with 1 count in 1st 1000 documents of NY Times corpus
  - About $633,000$ words, compared to $14,310,000$ in the whole corpus

Example PMIs (Manning & Schütze, 1999, p181)

I_{1000}	w^1	w^2	w^1w^2	Bigram
16.95	5	1	1	Schwartz eschews
15.02	1	19	1	fewest visits
13.78	5	9	1	FIND GARDEN
12.00	5	31	1	Indonesian pieces
9.82	26	27	1	Reds survived
9.21	13	82	1	marijuana growing
7.37	24	159	1	doubt whether
6.68	687	9	1	new converts
6.00	661	15	1	like offensive
3.81	159	283	1	must think

These values are are 2–4 binary orders of magnitude higher than the corresponding estimates based on the whole corpus

Alternatives to PMI for finding collocations

- There are a **lot**, all ways of measuring statistical (in)dependence
  - Student *t*-test
  - Pearson's χ^2 statistic
  - Dice coefficient
  - likelihood ratio test (Dunning, 1993)
  - Lin association measure (Lin, 1998)
  - and many more...
- Of those listed here, the Dunning LR test is probably the most reliable for low counts
- However, which works best may depend on particular application/evaluation

Improving PMI

Rather than using a different method, PMI itself can be modified to better handle low frequencies

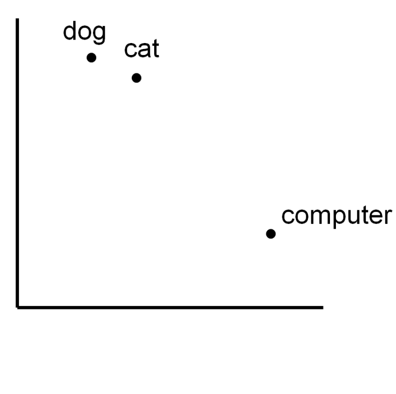
- Use **positive PMI** (PPMI): change all negative PMI values to 0
  - Because for infrequent words, not enough data to accurately determine negative PMI values
- Introduce smoothing in PMI computation
  - See J&M (3rd ed.) Ch 6.7 for a particularly effective method discussed by Levy, Goldberg and Dagan 2015

How to measure similarity

- So, let's assume we have context vectors for two words $\vec{v}$ and $\vec{w}$
- Each contains PMI (or PPMI) values for all context words
- One way to think of these vectors: as points in high-dimensional space
  - That is, we **embed** words in this space
  - So the vectors are also called **word embeddings**

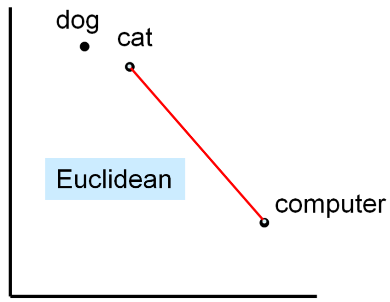
Vector space representation

- Example, in 2-dimensional space: `cat` $= (v_1, v_2)$, `computer` $= (w_1, w_2)$



Euclidean distance

- We could measure (dis)similarity using Euclidean distance: $\left(\sum_i (v_i - w_i)^2\right)^{1/2}$



- But doesn't work well if even one dimension has an extreme value

Dot product

- Another possibility: take the dot product of $\vec{v}$ and $\vec{w}$:

$$\begin{aligned} \text{sim}_{\text{DP}}(\vec{v}, \vec{w}) &= \vec{v} \cdot \vec{w} \\ &= \sum_i v_i w_i \end{aligned}$$

  - Gives a large value if there are many cases where v_i and w_i are both large: vectors have similar counts for context words

Normalized dot product

- Some vectors are longer than others (have higher values):
  [5, 2.3, 0, 0.2, 2.1] vs. [0.1, 0.3, 1, 0.4, 0.1]
  - If vector is context word counts, these will be *frequent* words
  - If vector is PMI values, these are likely to be *infrequent* words
- Dot product is generally larger for longer vectors, regardless of similarity

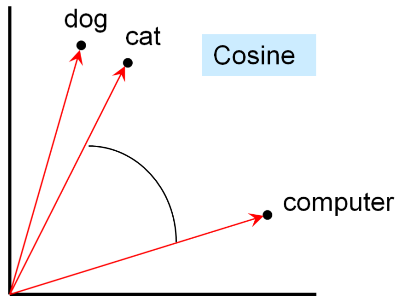
Normalized dot product

- Some vectors are longer than others (have higher values):
  [5, 2.3, 0, 0.2, 2.1] vs. [0.1, 0.3, 1, 0.4, 0.1]
  - If vector is context word counts, these will be *frequent* words
  - If vector is PMI values, these are likely to be *infrequent* words
- Dot product is generally larger for longer vectors, regardless of similarity
- To correct for this, we **normalize**: divide by the length of each vector:

$$\text{sim}_{\text{NDP}}(\vec{v}, \vec{w}) = (\vec{v} \cdot \vec{w}) / (|\vec{v}||\vec{w}|)$$

Normalized dot product = cosine

- The normalized dot product is just the cosine of the angle between vectors



- Ranges from -1 (vectors pointing opposite directions) to 1 (same direction)

Other similarity measures

- Again, many alternatives
  - Jaccard measure
  - Dice measure
  - Jenson-Shannon divergence
  - etc.
- Again, may depend on particular application/evaluation

Evaluation

- Extrinsic may involve IR, QA, automatic essay marking, ...
- Intrinsic is often a comparison to psycholinguistic data
  - Relatedness judgments
  - Word association

Relatedness judgments

- Participants are asked, e.g.: on a scale of 1-10, how related are the following concepts?

  LEMON FLOWER
- Usually given some examples initially to set the scale , e.g.
  - LEMON-TRUTH = 1
  - LEMON-ORANGE = 10
- But still a funny task, and answers depend a lot on how the question is asked ('related' vs. 'similar' vs. other terms)

Word association

- Participants see/hear a word, say the first word that comes to mind
- Data collected from lots of people provides probabilities of each answer:

	ORANGE	0.16
	SOUR	0.11
	TREE	0.09
LEMON ⇒	YELLOW	0.08
	TEA	0.07
	JUICE	0.05
	PEEL	0.04
	BITTER	0.03
	...	

Example data from the Edinburgh Associative Thesaurus: `http://www.eat.rl.ac.uk/`

Comparing to human data

- Human judgments provide a ranked list of related words/associations for each word w
- Computer system provides a ranked list of most similar words to w
- Compute the Spearman rank correlation between the lists (how well do the rankings match?)
- Often report on several data sets, as their details differ

Learning a more compact space

- So far, our vectors have length V, the size of the vocabulary
- Do we really need this many dimensions?
- Can we represent words in a smaller dimensional space that preserves the similarity relationships of the larger space?

Learning a more compact space

- So far, our vectors have length V, the size of the vocabulary
- Do we really need this many dimensions?
- Can we represent words in a smaller dimensional space that preserves the similarity relationships of the larger space?

We'll talk about these ideas next week