Explicit world-knowledge and distributional semantic representations

ESSLLI 2017 Day 3: Distributional semantics

Asad Sayeed and Alessandra Zarcone

University of Gothenburg, Saarland University

Remember our bird?

Well, let's get back to talking about lexical features.

Jorge Luis Borges

An Argentinian philosopher and fiction writer. One of his stories mentions 'a certain Chinese Encyclopedia', the *Celestial Emporium of Benevolent knowledge.* It contains a classifcation of animals.

- those that belong to the emperor
- embalmed ones
- those that are trained
- suckling pigs
- mermaids
- fabulous ones
- stray dogs

Jorge Luis Borges

. . . actually, it goes on.

- those that are included in the present classification
- those that tremble as if they are mad
- innumerable ones
- those drawn with a very fine camelhair brush
- others
- those that have just broken a flower vase
- those that from a long way off look like flies

Categories

The point is, lexical semantic features are a kind of categorization.

- Categories aren't stable: total subjectivity for our purposes.
- The goal is:
  - develop heuristic that allows us to separate informative ones from uninformative.
  - apply that heuristic to produce as many useful categories as possible, so that "flaws" in some categories are balanced out by other categories.

What words are

So far we've taken a somewhat formal approach to lexical semantics.
But natural language processing people tend to not be formal linguists, and "do" language on intuitions:

- We use dictionaries in real life for a reason.
- We need to make fine-grained distinctions, draw connections, and so on.
- Humans make judgements about similarities.
  - You know that "motorcycle" can be used in most, but not all contexts that "car" can be used.
  - English-German bilinguals know that "pride" and "Stolz" are quite similar.

Define "chair"

From dictionary.com (just the noun version):

- A seat, especially for one person, usually having four legs for support and a rest for the back and often having rests for the arms.
- Something that serves as a chair or supports like a chair: "two men clasped hands to make a chair for their injured companion".
- A position of authority, as of a judge, professor, etc.
- The person occupying a seat of office, especially the chairperson of a meeting: "the speaker addressed the chair"
- (in an orchestra) the position of a player, assigned by rank; desk: "first clarinet chair".
- "the chair", Informal. electric chair.

Words in terms of other words

That doesn't seem very helpful, but it gives us a place to start.
Define "chair" in terms of features:

- +one-person, +four-legs, +support, +backrest, +armrest
- +authority
- +occupies-chair
- +orchestra
- +execution

A place to start. . . doing what?

Part 1: distributional hypothesis: underlying intuitions

Underlying intuitions

Distributional hypothesis [Harris, 1954; Firth, 1957; Miller and Charles, 1991]

Two linguistic units are more semantically similar the more similar their context of occurrence are.

Basic idea: words occurring in similar contexts are semantically related

- structuralist concept of paradigmatic relations going back to de Saussure.
- linking distributional facts and semantic meaning became popular later.

Distributional semantic models (DSMs)

. . . a.k.a. vector space models

- represent linguistic units as corpus-extracted vectors
- dimensions are (a function of) co-occurence frequencies with other linguistic units.

DSMs differ with regard to:

- definition of context (content word within window, text regions, documents)
- their way of representing the distributional facts
- the aspects of meaning they are meant to represent
- similarity metrics

So back to our chair

Define "chair" in terms of features:

- +one-person, +four-legs, +support, +backrest, +armrest
- +authority
- +occupies-chair
- +orchestra
- +execution

Lexicon entries are only useful if there are other lexical items, so. . .

Define “cockpit”

Let’s go to dictionary.com again. I get as features:

- +enclosed, +airplane, +controls, +panel, +seats
- +instrumentation, +automobile
- +pit, +cockfights
- +conflict

Very little overlaps with “chair”.
But now we have a basis to compare them.

Representation

A first attempt: encode features as 1 or 0

	chair	cockpit
one-person	1	0
backrest	1	0?
four-legs	1	0
support	1	0?
armrest	1	0?
authority	1	0?
enclosed	0	1
airplane	0	1
seats	0?	1
...		

Similarity

- What we've just defined is a vector space.
- Dimension = feature. So far it's a low-dimensional space.
- How can we measure the similarity between them? Common answer: cosine similarity.

Cosine similarity

$$\text{sim}(\mathbf{A}, \mathbf{B}) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|}$$

- So what would the similarity of "chair" and "cockpit" be in our space? Probably zero!

Words in terms of other words

We need a new data source. Collect it from a real corpus. Let's try Google.

	chair	cockpit
one-person		
backrest		
four-legs		
support		
armrest		
authority		
enclosed		
airplane		
seats		
...		

Now it's not so bad: we can get a non-zero similarity. Yay?

I mean, Google is a corpus, right?

Well, it's an entire model. "Modeling assumptions" when using Google:

- Relevant counts are document counts, as opposed to strict mentions.
- Google's search algorithm chooses documents that are relatively representative.
- Google's semantic analysis doesn't introduce too much bias.

But it's still pretty good for examples and specific uses. Which ones?

Distributional semantic models (DSMs)

DSMs are appealing because they

- offer a straightforward way to represent meaning and compare representations.
- have a "cognitive vocation" – strong version of distributional hypothesis [Lenci, 2008] claims that:
  - distributional behaviour of word in context is direct correlate of its semantic content at cognitive level
  - context of occurrence provides an insight into the organization of the mental lexicon

Where do we get DSMs from?

At some point we have to count stuff.

- It turns out that the concept of "counting" can get terribly complicated.
- Design decisions involved in counting word contexts:
  - Application: not all meanings are relevant to all applications.
  - Cognitive theory: what do you think people remember when they learn?
  - Choice, size of corpus: does genre matter?
  - Do literal counts matter, or do we need to adjust?

Kinds of DSMs

DSMs can be divided into two overall categories:

- Count-based
  - Corpus counts (however chosen) are either taken "literally" or adjusted by an information statistic: pointwise mutual information, local mutual information, tf/idf, etc.
- Prediction-based
  - Counts are readjusted by applying machine learning techniques to "compress" the data.
  - Word contexts no longer necessarily human-comprehensible.

Part 2: corpus-based approaches to building distributional models

Words in terms of other words

- In fact, rather than using dictionary definitions of explicit features, cut out the middle man.
- "Learn" a vector for each word by counting corpus context. Ways of learning:
  - Simple co-occurrence counts based on a window.
    - The vocabulary basically becomes the feature space.
  - More complex counts, such as part-of-speech tags, bits of parse trees.
- Sometimes raw counts aren't what you need: smoothing, reweighting.

Words in terms of other words

These are "count" vectors. What are the problems with doing it this way?

- Sparsity: many words just never appear with other words.
- Dimensionality: especially if you use fancy features (syntax, etc), you get million dimensional spaces.

What we need? Dimensionality reduction, or some other way to start from a compressed space.

- Sharing dimensions helps generalization.
- Nevertheless, there's value in count vectors (for things that require explicit linguistic knowledge)

So now... "predict" vectors...

The power of dimensionality reduction

The "vector spaces" of count-based DSMs are very high-dimensional.

- Any "respectable" DSM will have a dimensionality that is lexicon scale (at least!).
- But since the dimensions are labelled by linguistic features, those features have interrelationships.
- What we need: a way to "compress" feature dimensions so that relationships are revealed.

Dimensionality reduction == clustering features that have "latent" relationships.

- e.g., a particular class of verb may be partly associated with particular negative polarity items ("any", "nobody", etc).
- Downside: often lose all direct human interpretability of "reduced" feature space.

Common dimensionality reduction

There are very many ways to generate lower-dim spaces.Examples:

- Tensor factorization approaches:
  - Factor the vector space into multiple smaller-dimensional matrices.
  - Select rows/columns by importance heuristic
  - Examples: Latent Semantic Indexing, Principal Component Analysis
- Discriminative training/machine learning approaches
  - Iteratively update smaller-dimensional vectors.
  - Update based on ability to reconstruct "objective" data.
  - Examples: autoencoders, deep learning

Latent semantic analysis

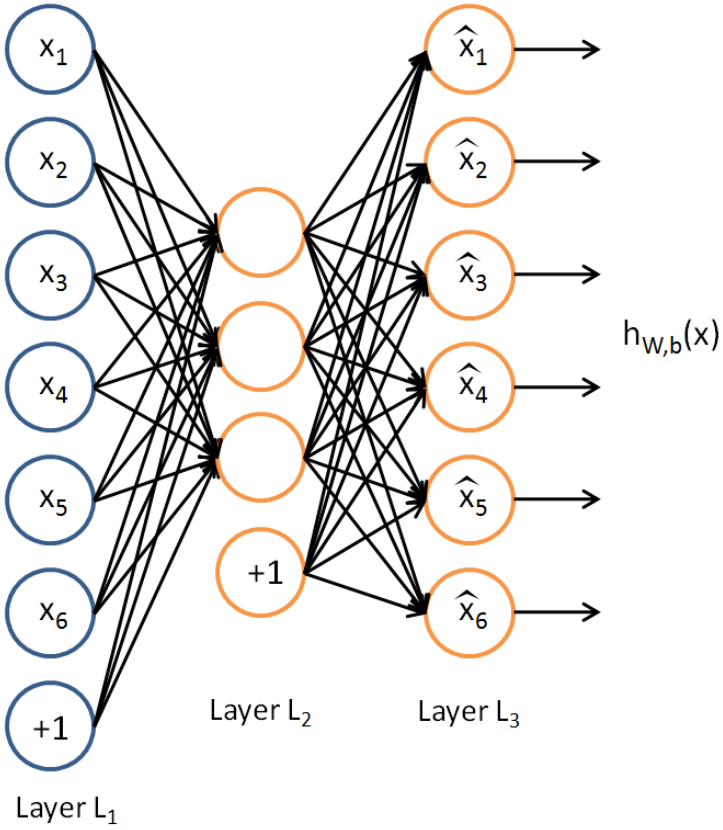
Factoring via Singular Value Decomposition (SVD) – very widely used. From Wikipedia:

$$\underset{(\mathbf{t}_i^T)\rightarrow}{\overset{X}{\overset{(\mathbf{d}_j)}{\overset{\downarrow}{}}}}\begin{bmatrix} x_{1,1} & \cdots & x_{1,n} \\ \vdots & \ddots & \vdots \\ x_{m,1} & \cdots & x_{m,n} \end{bmatrix} = \underset{(\hat{\mathbf{t}}_i^T)\rightarrow}{U}\begin{bmatrix} \begin{bmatrix} \\ \mathbf{u}_1 \\ \\ \end{bmatrix} \cdots \begin{bmatrix} \\ \mathbf{u}_l \\ \\ \end{bmatrix} \end{bmatrix} \cdot \underset{}{\Sigma}\begin{bmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_l \end{bmatrix} \cdot \underset{}{\overset{V^T}{\overset{(\hat{\mathbf{d}}_j)}{\overset{\downarrow}{}}}}\begin{bmatrix} [& \mathbf{v}_1 &] \\ & \vdots & \\ [& \mathbf{v}_l &] \end{bmatrix}$$

The σ are ranked, simply cut off dimensions at low-enough σ.
All vectors are similarly reduced so can multiply again to get low-dim matrix.
Dimensions represent "fuzzy" clusters, not directly interpretable.

Autoencoder

From Stanford deep learning tutorial:



Learn compressed representation of the input by learning the identity function via a neural network.

So now let's make it concrete.

Part 3: applications in natural language processing

We talked on previous days about world knowledge. . .

. . . and that heavily with reference to events

What does an event consist of?

- A predicate – whatever is happening (or in some cases the description of a state)
- Participants – the objects/entities/abstract constructs that make the predicate specific.

Participants are usually defined by "thematic" or semantic roles.

- Traditionally: agent, patient, goal, etc.
- Some roles are "required" by particular events (often agents and patients for transitive verbs), most are "adjuncts" (locations, instruments, etc.)

Thematic fit

A challenge in building computational models of events.

The thematic fit problem

Given a verb/event-type v, an entity x, how well does v fit x in role r?

We ask typically ask humans to give us this data.

- Need to get ratings. Possible questions:
  - "How common is it for a cake to bake something?" (agent)
  - "An oven is something you can use for baking." (instrument)

  Rate from 1-7.

(How you ask actually makes things complicated. . .)

Agent/patient (subj/obj) ratings

Verb	Noun	Semantic role	Score
advise	doctor	subj	6.8
advise	doctor	obj	4.0
confuse	baby	subj	3.7
confuse	baby	obj	6.0
eat	lunch	subj	1.1
eat	lunch	obj	6.9
kill	lion	subj	2.7
kill	lion	obj	4.9

Data source for thematic fit norms

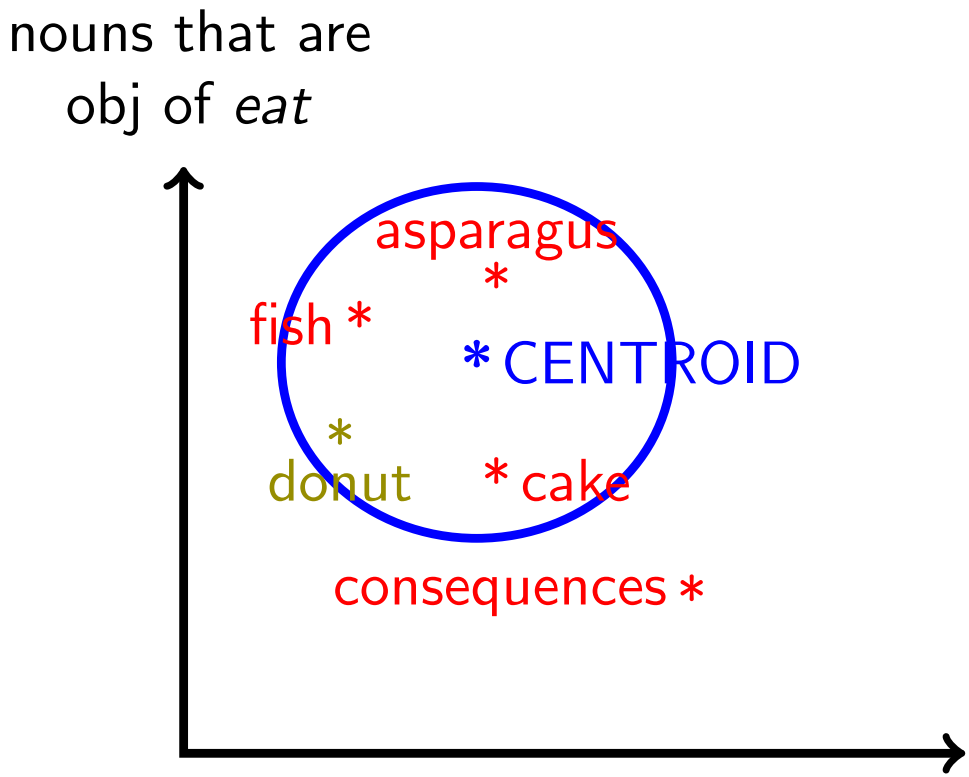
Here are some widely available thematic fit rating sources.

- Padó agent/patient ratings
  - Balanced rating set of 18 verbs with 12 nouns each extracted from WSJ.
- McRae agent/patent ratings: 1444 ratings, unbalanced
- Ferretti et al.: instruments (248) and locations (274).
- Greenberg et al.: patients balanced for number of senses (from WordNet).

Now assume for a moment that we have a vector space.

How to evaluate thematic fit with a DSM

Query: how good is "donut" as an object of "eat"?

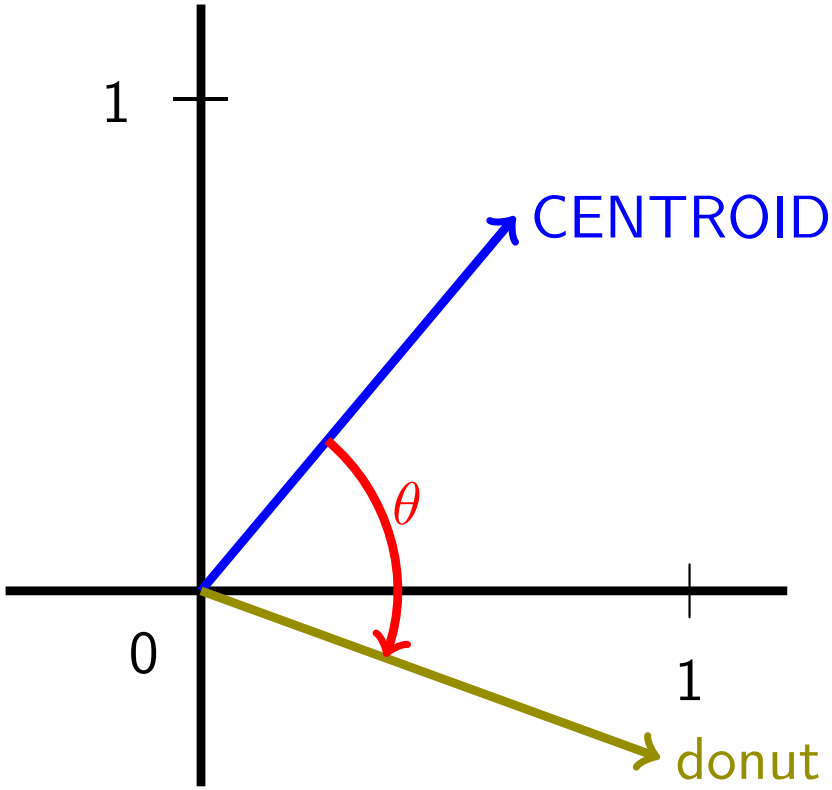


(Special thanks to A. Zarcone.)

Find an average vector (centroid) based on 20 nouns that are typical "eat"-objects.

Thematic fit measurement

Query: how good is "donut" as an object of "eat"?



Then take the cosine of "donut" with the centroid.

So assuming a cosine similarity approach. . .

. . . what do we need to build a vector space of this kind?

- At minimum, a space that allows us to assess most frequent fillers of given role.
- Bonus: space that gives us semantically-relevant features to compare.

First let's try a count space . . .

Distributional Memory

Baroni and Lenci [2010]: Distributional Memory (DM) approach:

1. Parse entire corpus (using MaltParser).
2. Read into data structure (order-3 tensor) as counts of $<$*word0, link, word1*$>$ dimensions.
   - Where "link" is a feature derived from a dependency between "word0" and "word1".
3. Reweight counts with Local Mutual Information (LMI).

Local mutual information

$$O \log \frac{O}{E}$$

where O is observed counts of triples in corpus and E is counts expected under independence of words and links.

This process results in a tensor space of tens of millions of dims.

What are the feature spaces like?

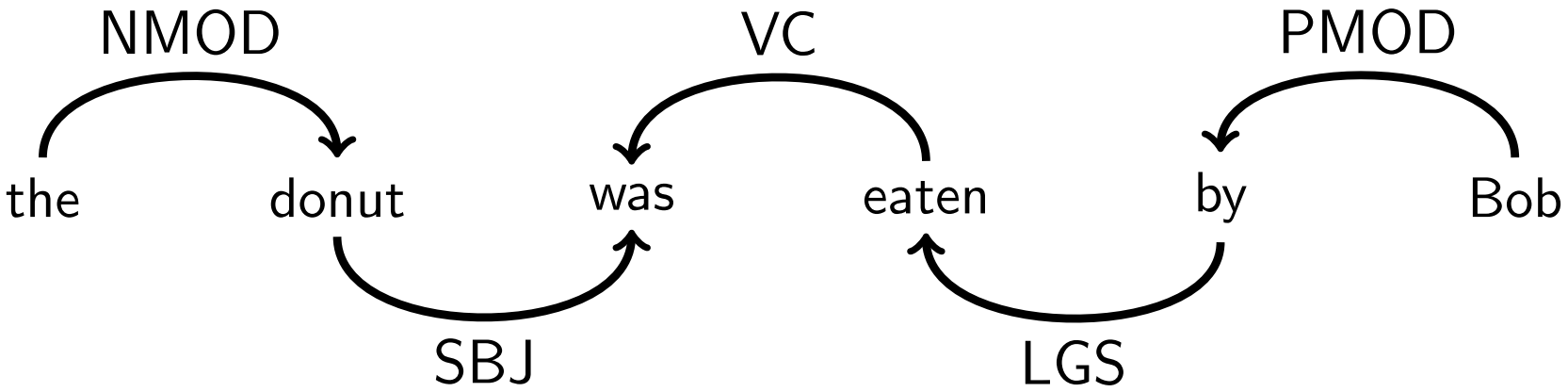
Baroni and Lenci come up with three different tensors:

- DepDM – Raw dependencies from MaltParser, adjusted in process similar to ours.
- LexDM – Lexicalized links based on DepDM, expanded by handcrafted rules.
- TypeDM (publicly available) – Counts reflect *number of types* of links in LexDM, rather than raw counts.

Corpora: UKWAC, WackyPedia, BNC.

TypeDM feature space

Baroni and Lenci's TypeDM model: "semantic" features hand-crafted from syntactic dependencies.



Donut

A small excerpt of a Baroni and Lenci DM

	⟨*verb,bomb*⟩	⟨*subj,kill*⟩	⟨*verb,gun*⟩	⟨*subj,shoot*⟩	⟨*verb,book*⟩	⟨*subj,read*⟩
marine	40.0	82.1	85.3	44.8	3.2	3.3
teacher	5.2	7.0	9.3	4.7	48.4	53.6

How well does all of this work?

Evaluation via Spearman's ρ.
(Rank-based correlation – is this a good idea?)

- Average human agreement $= 68$ on Padó data.
- TypeDM on Padó agent/patient: Spearman's ρ correlation: 53

Other roles do significantly worse. (e.g. Ferretti locations get 23)...

. . . so consider some potential linguistic explanations for that.

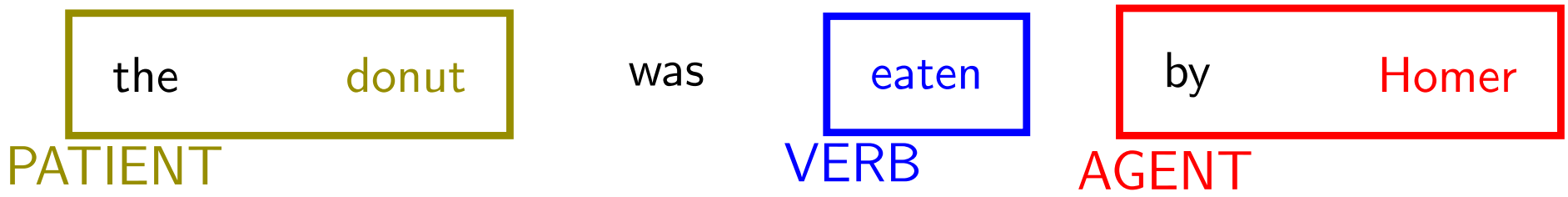
Can we do better?

Even with count-based models, yes.

- Why do we want to do better with count-based models when we have prediction models?
  - Mainly, if we want to advance the state of (mostly) unsupervised modeling.
- Other than that: things to consider.
  - TypeDM uses syntactic parses as "stand-in" for semantics.
  - Consider verb (predicate) senses: cutting a budget is not the same as cutting class, etc.

Using semantics "directly"

We can label the data source in any way we want.

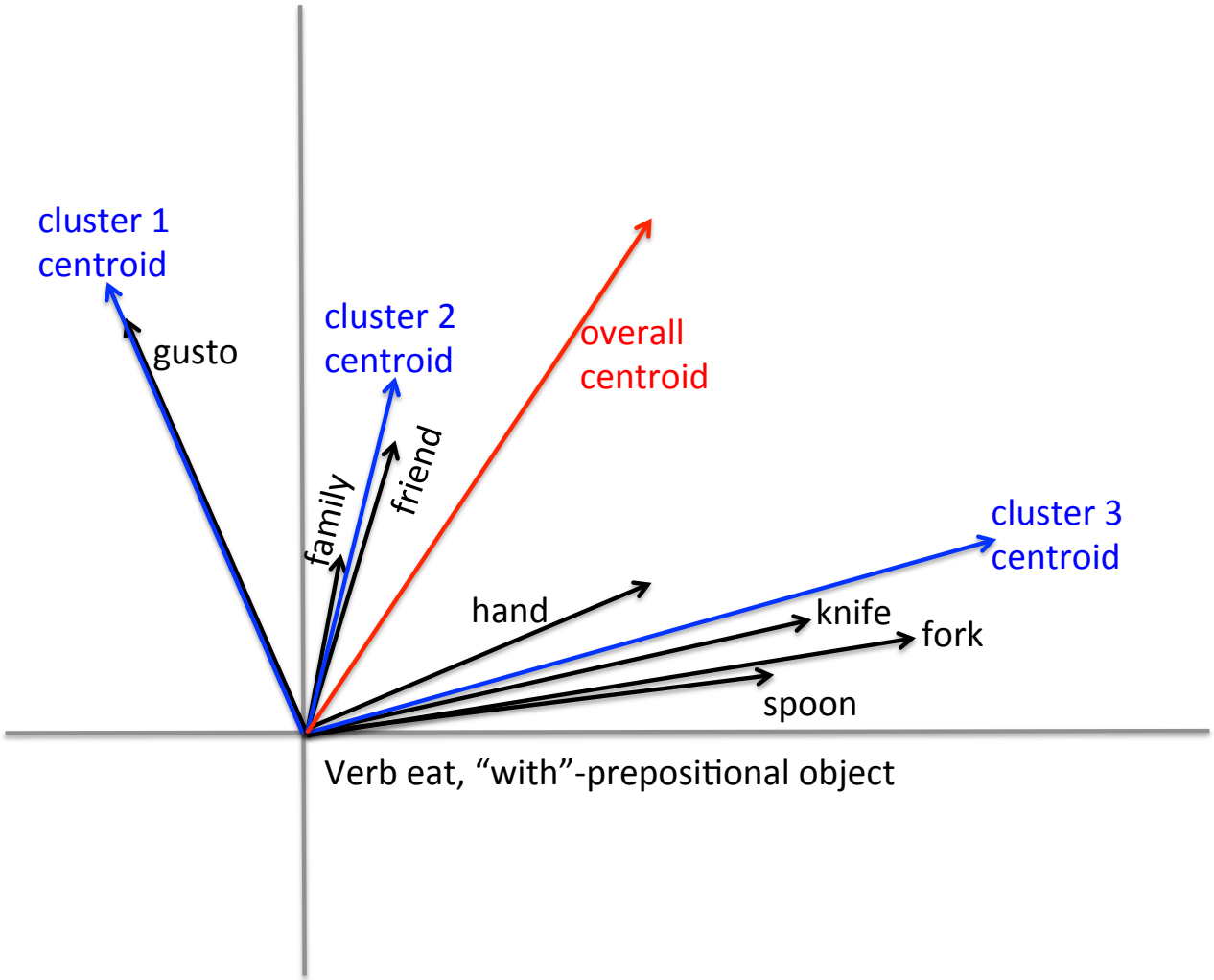


Get this from a neurally-trained semantic role labeller: SENNA.

Sample of experiments

- Does it work? The answer is **yes** [Sayeed and Demberg, 2014; Sayeed, Demberg, and Shkadzko, 2016]:
  - Best model on agent/patient data: combine syntax-derived [Baroni and Lenci, 2010] features with SRL-based features.
    - Our result: $\rho = 59$
    - Syntax-only baseline: $\rho = 53$.
  - Best model on instrument role data [Ferretti et. al, 2001]:
    - Our result: $\rho = 45$
    - Syntax-only baseline: $\rho = 36$

Consider sense clustering



Increasing implicit knowledge

Greenberg, Sayeed, and Demberg [2015]: yes, it improves thematic fit modeling.

- Observed systematic improvements on Spearman's ρ on most thematic fit data sets.
- Created Greenberg et al. patient data set with human judgements via Mechanical Turk.
  - Intuition: humans have more trouble coming up with sharp judgements for highly polysemous predicates.
  - Confirmed by analysis of human judgements, independent of verb mention frequency in corpus.

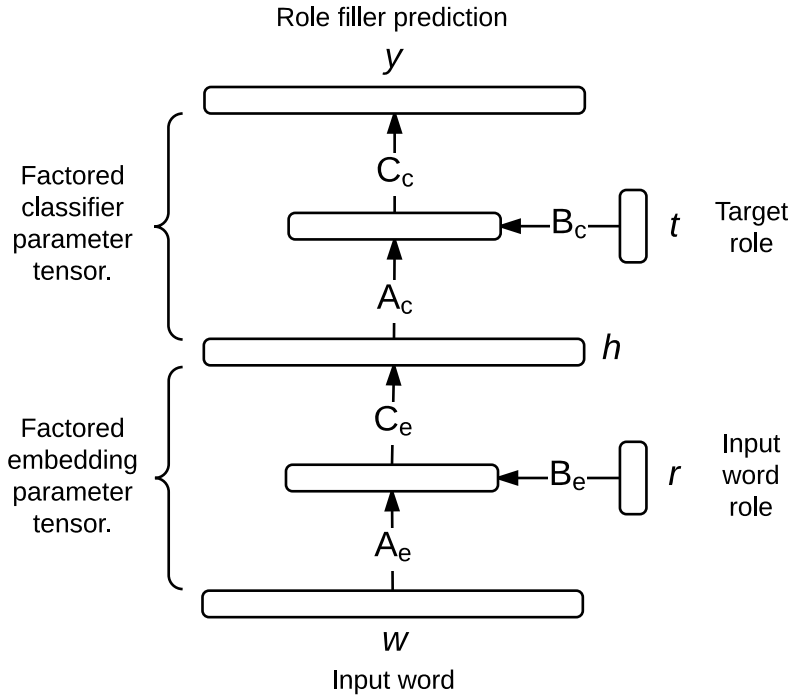
A note on prediction

Mostly, neural network/deep learning approaches.
We'll mention work on this a little more on Friday, but some things to think about:

- Processing intensive – iterative adjustment of vector space takes a lot of GPUs.
- Very effective for similarity problems.
  - Baroni et al. [2014]: tested count vs. predict vectors over numerous tasks and parameter settings.
  - Only task on which count did better than predict is. . . thematic fit!
  - (Thematic fit different from analogic similarity: why?)

What prediction might look like

- Deep learning for compositionality across multiple roles [Tilk, Demb erg, Sayeed, Klakow, Thater, 2016].
  - Simulate compositionality, prediction with neural network.



Top locations for serving given a subject:

```
Where does a clerk serve?
0.029378 office
0.026096 committee
0.025173 room
0.018917 meeting
0.018850 hall

Where does a waiter serve?
0.162362 restaurant
0.076326 bar
0.047854 room
0.023533 table
0.012684 pub

Where does a priest serve?
0.069048 church
0.050872 army
0.034693 war
0.017941 there
0.017477 room

Where does a prisoner serve?
0.074051 prison
0.066616 war
0.055304 army
0.024465 force
0.022381 raf
```

But the moral of the story is. . .

. . . there's no getting away from structure.

- Huge gains (in terms of modeling performance) from building linguistically **relatively** naive feature spaces.
- But most gains on top of that require some thought about language.
  - Sense clustering: automatically induce senses from the vectors, but have encoded that senses matter.
  - Feature space: sure they seem naive, but the choice has a systematic effect.
    What knowledge do they encode?

Some roles are stubborn, don't improve easily, e.g., location. Why?

Teamwork time!

A small-group exercise: consider a research direction

Try to answer the following questions in small group format:

- In your combination of fields/research interests, try to come up with a problem of lexical knowledge – e.g., relevant semantic content that has a relationship to the "real world" is is non-trivial to solve.
- Identify human experiments that might be helpful in building a behavioural model.
- Identify the kinds of data sources you might need to build a computational model.
- Identify underlying algorithms or approaches you would use to construct a computational model.

You have 30 minutes here to discuss, then on Friday we will spend 30 minutes on presentations/Q&A/discussion of these points.