

Natural Language Categorization

Trevor Fountain

T.Fountain@sms.ed.ac.uk

University of Edinburgh



Introduction

Categorization is the process by which people group items into categories and use those categories to reason about new items. Whereas traditional models of categorization deal mainly with real-world objects, ours is a model of **natural language categorization**; e.g., we model how people group *words* into categories.

Unfortunately, recasting the problem as a language task makes it significantly more difficult to extract a representation for items; we therefore explore a number of vector-space representations for words within the context of supervised and semi-supervised models.

Theories

Models of categorization tend to fall under one of three theories: the classical, prototype, or exemplar model.

- In a **classical model** categories are represented by a list of features which are both *necessary* and *sufficient* to describe all items within the category.

EXAMPLE: "Tree": *has-leaves* + *is-tall* + *made-of-wood*

- Similarly, a **prototype model** represents categories by a list of features, but replaces the necessary-and-sufficient restriction by assigning each feature a weight.

EXAMPLE: "Tree": *has-leaves* (0.6), *is-tall* (0.9), *made-of-wood* (1.0)

- An **exemplar model**, however, represents categories by simply storing a list of known exemplars.

EXAMPLE: "Tree": "Pine", "Oak", "Elm"

Model

We use a simplified variant of Nosofsky's exemplar model, in which the similarity $\eta_{x,c}$ between an exemplar x and a category c is computed:

$$\eta_{x,c} = \frac{1}{|C|} \sum_{i \in C} sim(x, i) \quad (1)$$

where $sim(i, j)$ is the similarity between two items.

Representation

The critical difference between our model and existing models of categorization is in how items are represented:

- Items are represented by feature vectors
- In a cogsci model these are typically real-world attributes (color, shape, size)
- Similarity is the distance between vectors in the feature-space

This begs the question: what are the contextual 'features' of words?

- Difficult to determine in a general fashion, **but**:
- There are plenty of established ways for computing word *similarity*, why don't we try some of those?
 - Latent Semantic Analysis (LSA): document co-occurrence
 - Latent Dirichlet Allocation (LDA): topic co-occurrence

Tasks

Because categorization is such a broad topic, we model performance on three related tasks:

- **Category Naming:** Given a word, can we predict the proper category label?
EXAMPLE: "Apple" → "Fruit"
- **Typicality Rating:** Given a word-category pair, can we rate how 'typical' the word is among members of that category?
EXAMPLE: "Apple/Fruit" → 0.9 (typical)
EXAMPLE: "Lychee/Fruit" → 0.1 (atypical)
- **Exemplar Generation:** Given a category label, can we come up with a set of likely exemplars?
EXAMPLE: "Fruit" → "Apple", "Pear", "Banana", etc...

Data

Data for all three tasks was collected in an elicitation study conducted using Amazon Mechanical Turk (AMT).

- In the exemplar generation study participants generated exemplars for 68 distinct categories distributed equally according to both level of specificity (abstract, basic, specific) and type (natural, artificial, and abstract).
- For each resulting exemplar-category pair, additional annotators rated how typical the exemplar was among members of the category.
- For each exemplar yet another set of annotators were asked to provide the correct category label.

Training

In an exemplar model of categorization we need a number of things:

- Feature representation for exemplars: LSA, LDA, and Depspace
- Similarity function between exemplars: cosine distance (exemplars are in vector space)
- List of exemplars belonging to each category: from AMT (for supervised training) or output of a clustering algorithm (semi-supervised)

Summary: *lots* of possible combinations of the form *representation* + *distance function* + *clustering algorithm*.

Evaluation

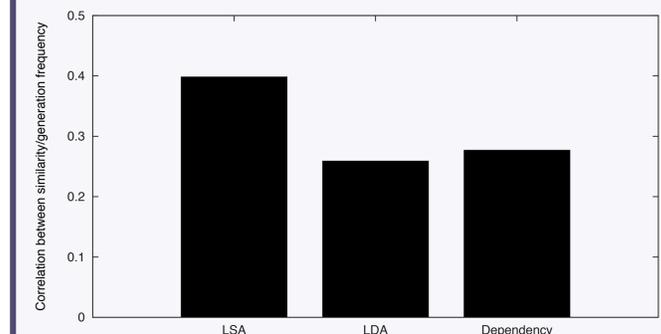
Evaluate performance on each task separately:

- **Category Naming:** Proportion of exemplars correctly labeled
- **Typicality Rating:** Correlation between $\eta_{x,c}$ and rated typicality
- **Exemplar Generation:** Average overlap between exemplars with highest $\eta_{x,c}$ and most frequently generated in AMT data

Results

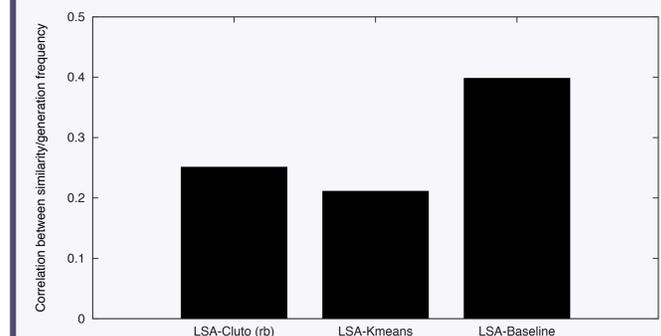
To compare spaces and clustering algorithms we find the correlation between the predicted and actual typicality ratings:

Which space should we use to calculate similarity? Does it matter?



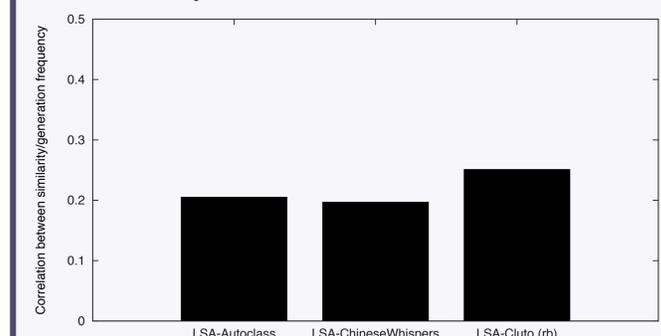
LSA is better than LDA or Depspace ($p < 0.01$).

Can we group words into categories automatically?



Not great, but not too bad either.

Can we also determine the number of categories automatically?



Yes, and with almost no performance hit.