Lab 2 - Correlation and Purity

Semantic Relatedness and Concept Categorization

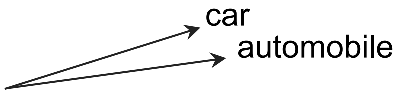
Alberto Testoni, 12nd November 2020

Semantic Relatedness Task

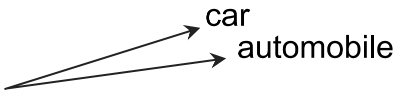
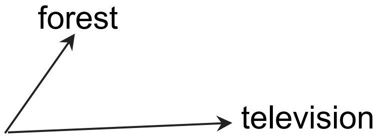
- Human subjects are asked to **rate** the degree of semantic similarity between two words on a numerical scale.

- The performance of a computational model is assessed in terms of **correlation** between the average scores that subjects assigned to the pairs and the cosine between the corresponding word embeddings.

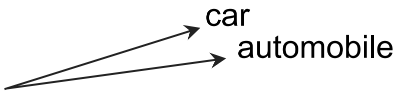
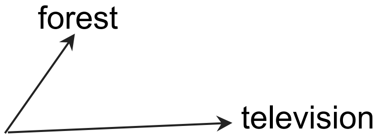
Semantic Relatedness Task

Word pair	Relatedness assigned by human annotators (0-4 scale)	Word embeddings in Word2Vec	Cosine similarity between word embeddings
automobile-car	3.92		≈ 0.8-0.9

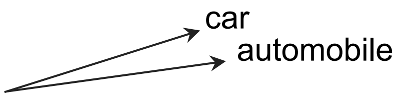
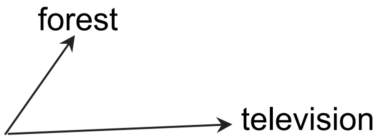
Semantic Relatedness Task

Word pair	Relatedness assigned by human annotators (0-4 scale)	Word embeddings in Word2Vec	Cosine similarity between word embeddings
automobile-car	3.92		≈ 0.8-0.9
forest-television	1.2		≈ 0.1-0.2

Semantic Relatedness Task

Word pair	Relatedness assigned by human annotators (0-4 scale)	Word embeddings in Word2Vec	Cosine similarity between word embeddings
automobile-car	3.92		≈ 0.8-0.9
forest-television	1.2		≈ 0.1-0.2

Semantic Relatedness Task

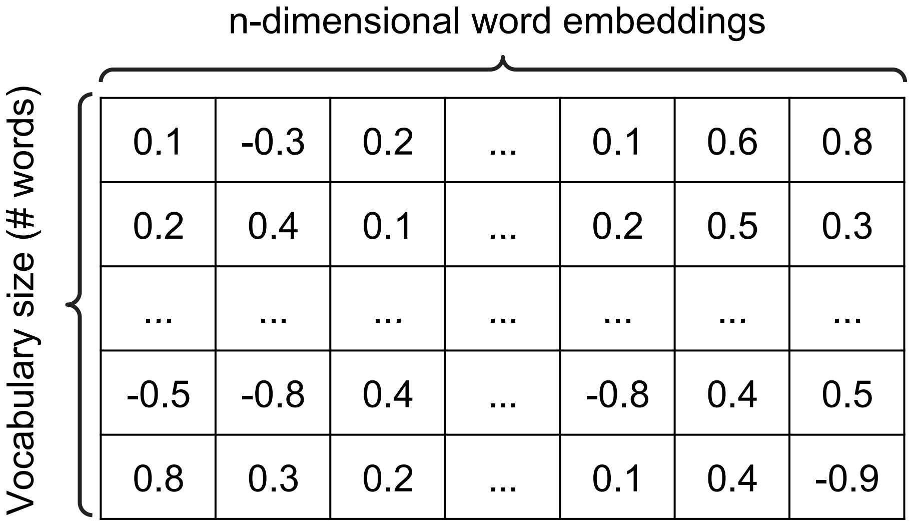
Word pair	Relatedness assigned by human annotators (0-4 scale)	Word embeddings in Word2Vec	Cosine similarity between word embeddings
automobile-car	3.92 ↑		≈ 0.8-0.9 ↑
forest-television	1.2 ↓		≈ 0.1-0.2 ↓

Semantic Relatedness Task

What we need:

- A **dataset** of word pairs with corresponding human relatedness scores
- A **matrix** of word embeddings
- A “**dictionary**” that maps each word to a row in the matrix, and vice versa
- A statistical measure of **correlation**

Matrix of word embeddings & mapping dictionaries



n-dimensional word embeddings

Vocabulary size (# words)

0.1	-0.3	0.2	...	0.1	0.6	0.8
0.2	0.4	0.1	...	0.2	0.5	0.3
...	...	...	...	...	...	...
-0.5	-0.8	0.4	...	-0.8	0.4	0.5
0.8	0.3	0.2	...	0.1	0.4	-0.9

word2idx	**idx2word**
dog: 0	0 : dog
city : 1	1 : city
….	….
friend : 3999	3999 : friend
Paris : 4000	4000 : Paris

Semantic Relatedness Datasets

Two datasets:

1. Rubenstein and Goodenough (1965): the *rg* dataset consists of 65 noun pairs. Performance evaluated using **Pearson** correlation.

2. Bruni et al. (2013): The *MEN* dataset comprises 1000 word pairs. Performance evaluated with **Spearman** correlation.

Pseudocode

```
Load a matrix of word embeddings, and two mapping dictionaries word2idx and idx2word
Load two empty lists: human_relatedness and word2vec_relatedness
Open the dataset file

Repeat for each line in the dataset file:
        Save word1, word2, and the relatedness score assigned to this pair by human annotators
        Append the relatedness score score to human_relatedness list
        Get the word embeddings of word1 and word2 from the matrix
        Compute the cosine similarity between the two word embeddings
        Append this cosine similarity to the word2vec_relatedness list

Compute the Pearson/Spearman correlation between human_relatedness and
word2vec_relatedness
```

Let’s Look at the Code!

https://colab.research.google.com/drive/1RPY23jC3QXymfIZy4ihOSdvOLzJKAJ0F?usp=sharing

Concept Categorization - The Task

- Given a set of nominal concepts, the task is to **group** them into natural categories (e.g., *helicopters* and *motorcycles* should go to the *vehicle* class, *dogs* and *elephants* into the *animal* class).

- The performance of a computational model is assessed in terms of **purity**, a measure of the extent to which each cluster (group) contains concepts from a single category.

Concept Categorization - The Dataset

ANIMALS

dog

cat

duck

VEGETABLES

onion

potato

pumpkin

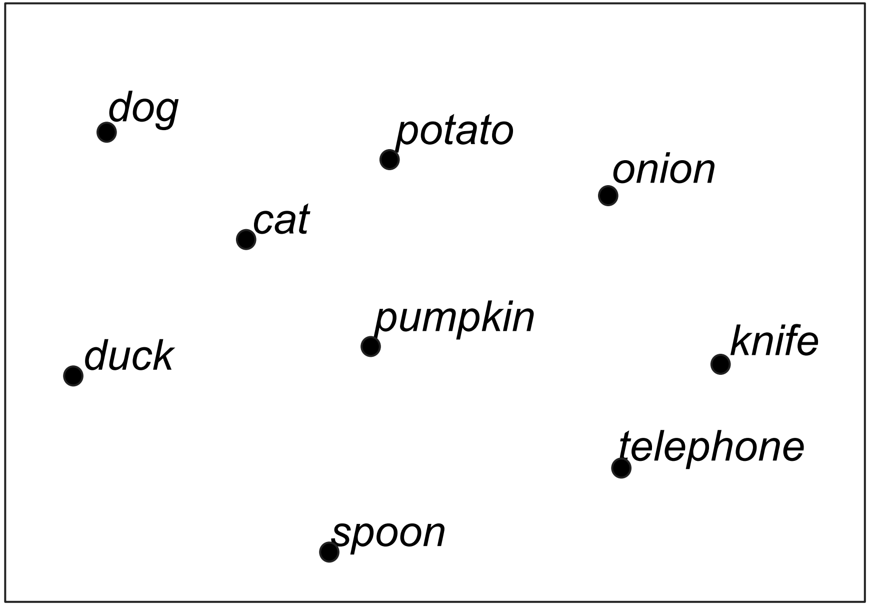
TOOLS

knife

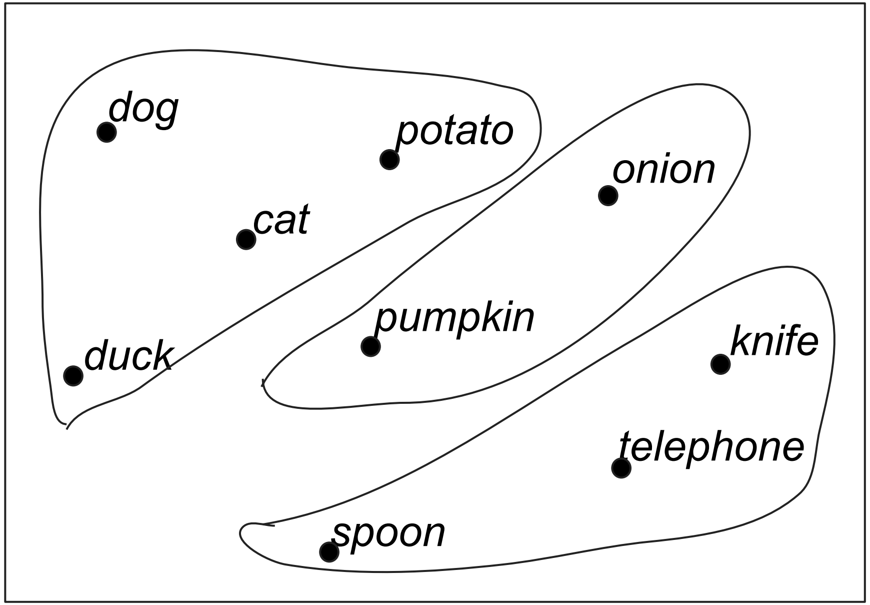
telephone

spoon

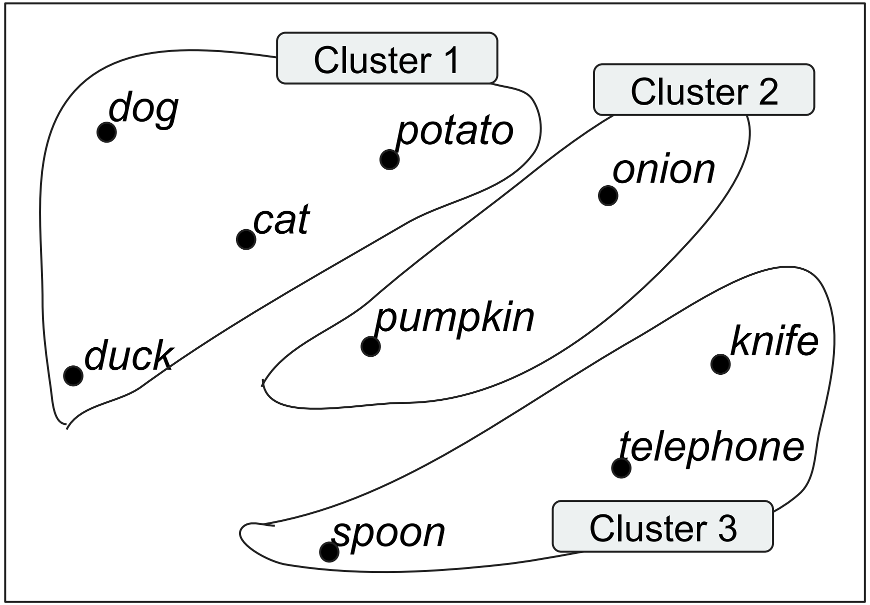
Concept Categorization - Word Embeddings



Concept Categorization - Clustering



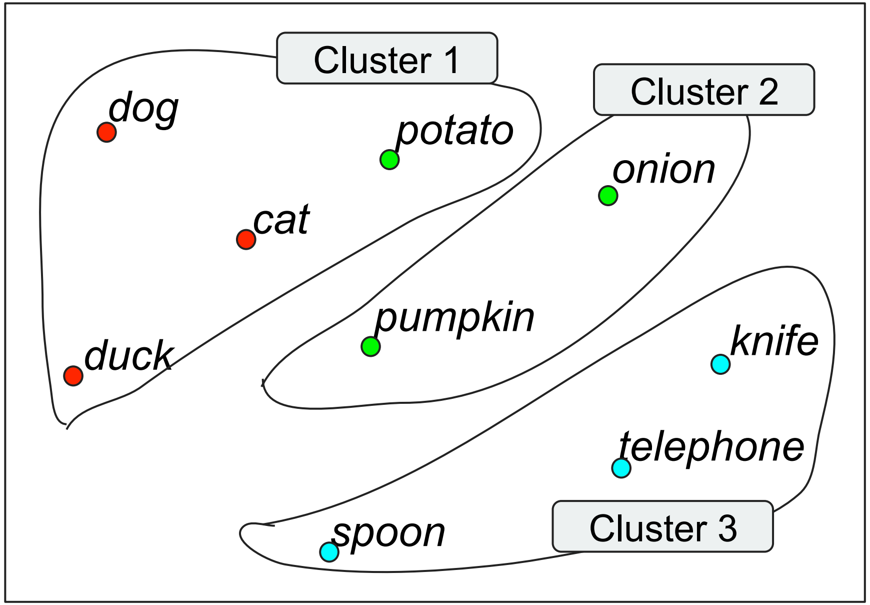
Concept Categorization - Purity



Contingency matrix

	Cluster 1	Cluster 2	Cluster 3
animals			
vegetables			
tools			

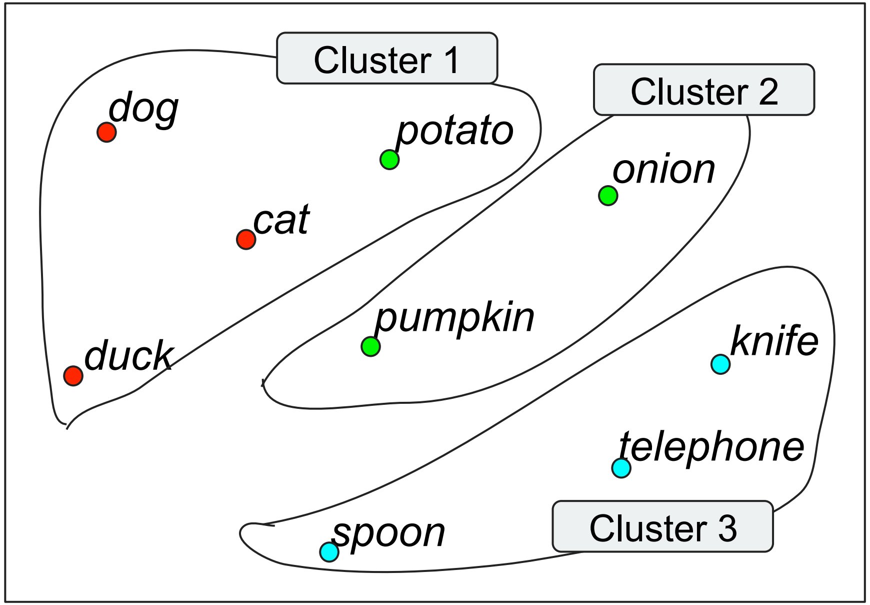
Concept Categorization - Purity



Contingency matrix

	Cluster 1	Cluster 2	Cluster 3
animals	3		
vegetables	1		
tools	0		

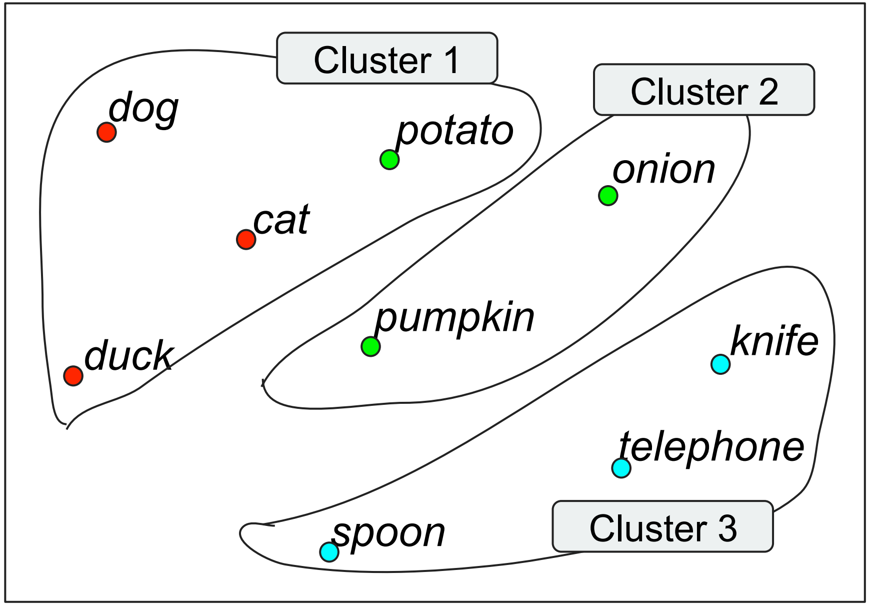
Concept Categorization - Purity



Contingency matrix

	Cluster 1	Cluster 2	Cluster 3
animals	3	0	
vegetables	1	2	
tools	0	0	

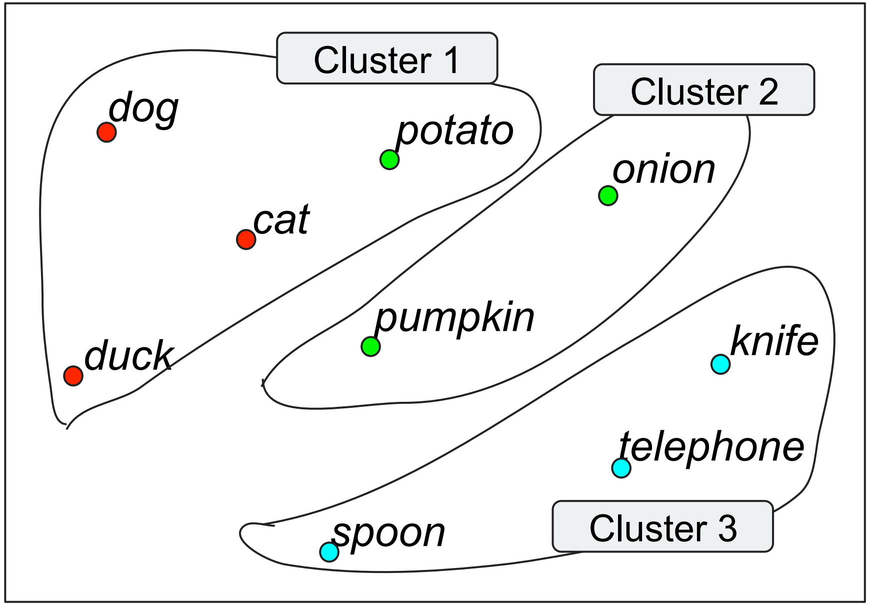
Concept Categorization - Purity



Contingency matrix

	Cluster 1	Cluster 2	Cluster 3
animals	3	0	0
vegetables	1	2	0
tools	0	0	3

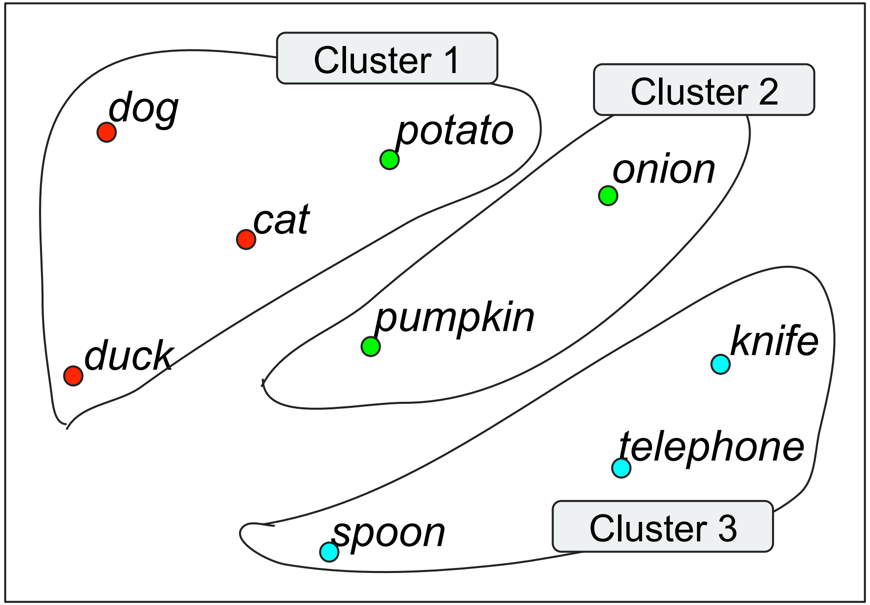
Concept Categorization - Purity



Contingency matrix

	Cluster 1	Cluster 2	Cluster 3
animals	3	0	0
vegetables	1	2	0
tools	0	0	3
	max: 3	**max**: 2	**max**: 3

Concept Categorization - Purity

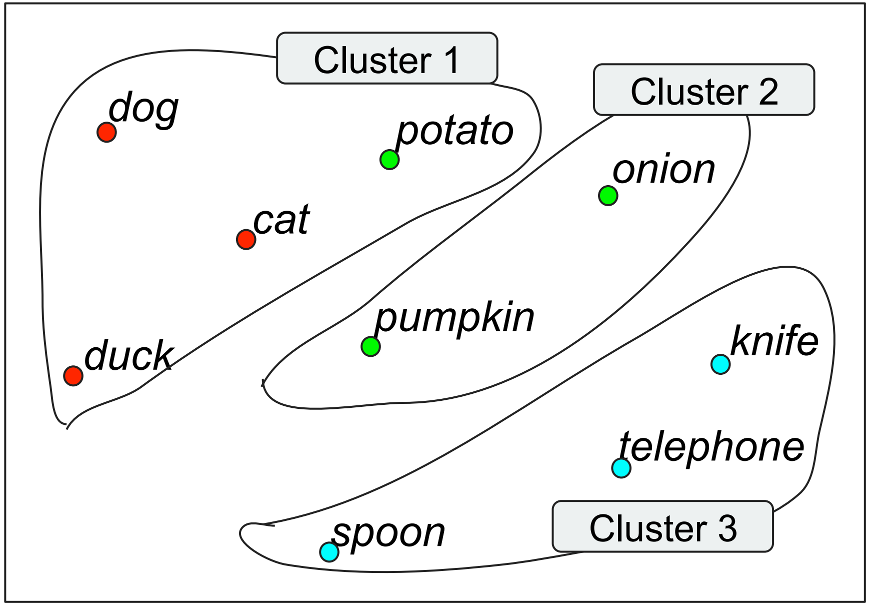


Contingency matrix

	Cluster 1	Cluster 2	Cluster 3
animals	3	0	0
vegetables	1	2	0
tools	0	0	3
	max: 3	**max**: 2	**max**: 3

sum: 8

Concept Categorization - Purity



Contingency matrix

	Cluster 1	**Cluster 2**	**Cluster 3**
animals	3	0	0
vegetables	1	2	0
tools	0	0	3

max: 3 **max**: 2 **max**: 3

sum: 8

$$\textbf{purity} = \frac{8}{9} \approx 0.89$$

Total number of datapoints

Concept Categorization

What we need:

- A **dataset** of words paired with their category
  (ESSLLI 2008 dataset: 44 words belonging to 6 categories)
- A **matrix** of word embeddings
- A “**dictionary**” that maps each word to a row in the matrix, and vice versa
- A **clustering algorithm** (K-Means) - more in ML for NLP course
- A metric to evaluate the cluster quality (**purity**)

Pseudocode

```
Load a matrix of word embeddings, and two mapping dictionaries word2idx and idx2word
Load an empty matrix where you will save the word emb. of each word in the dataset:
test_word_embeddings
Load an empty list: gold_standard_labels
Open the dataset file

Repeat for each line in the dataset file:
        Save the input word and its semantic_category
        Append the semantic category to gold_standard_labels list
        Get the embedding of word from the matrix
        Save this word embedding in the test_word_embeddings matrix

Run the clustering algorithm over the test_word_embeddings
Compute the purity of the clusters with respect to the gold_standard_labels
```

Let’s Look at the Code!

https://colab.research.google.com/drive/1RPY23jC3QXymfIZy4ihOSdvOLzJKAJ0F?usp=sharing