Vector space models and semantic analysis

Simon Musgrave
Alice Gaby
Gede Primahadi Wijaya Rajeg
Harris (1954:156):
“difference of meaning correlates with difference of distribution”

Firth (1957):
“a word is characterized by the company it keeps”

Can such distributional relations be made precise?

And specifically, can minimally different pairs (or groups) of words be identified, revealing oppositions?
The distribution of words can be expressed mathematically as *vectors*. A vector is a table with a single row. The vector for any given word records its co-occurrence with other words:

- Each entry in the row corresponds to another word.
- The entry records some information about the co-occurrence of the two words.
Two questions:

- What is the domain within which co-occurrence is tracked?
- What information is stored? E.g. is it just the fact of co-occurrence or is it richer information such as distance between words?

Example with a simple approach:

- Domain is a sentence
- Information stored is number of times a word occurs
Vector example

<table>
<thead>
<tr>
<th></th>
<th>Half</th>
<th>Mushrooms</th>
<th>Onion</th>
<th>The</th>
<th>thinly</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. Thinly SLICE half the onion</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>b. SLICE the mushrooms thinly</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
- Vectors derived from any large corpus will be very large
  - At least each lemma will have a vector
  - For our COCA data, there are almost 0.5 million vectors
- Data is sparse – a very large proportion of the entries are zeroes
- Various algorithms have been developed to reduce the size of the output while preserving information
Spatial vectors

- One approach reduces the raw vectors to a multidimensional spatial model
- **Word2vec** uses this approach
  - Word2vec uses neural networks to get from text to spatial model
- **Output** is an n-dimensional model which locates all words (lemmas) in relation to each other
  - Still a lot of data – our 100-dimensional model is a c190mb file
“dans la langue il n’y a que des différences” (Cours, 166)

But also:
“deux signes comportant chacun un signifié et un significant ne sont pas différents, ils sont seulement distincts. Entre eux il n’y a qu’opposition” (Cours, 167)

Empirical investigation of oppositions is limited
– Possible in limited domains
– Very difficult for a language as a whole
Model locates every word in text relative to all the others
Relationships can be quantified
Is this close to a Saussurean semantic analysis?

We look at a group of verbs:
– CUT and BREAK concepts
– Previously studied in detail (Majid et al 2007, 2008a, b))

Concentrate on clustering:
– Do clusters make intuitive sense?
– How do they correspond to previous work?
– Do Saussurean oppositions emerge?
What we are looking for

- Exploring thematic cluster
  - Do the CB verbs fall into clusters?
  - If so, how many? How do we determine that?
  - What semantic theme could be explored from the clustering of particular CB words?

- Exploring “nearest” verbs to each of the CB verbs
  - Which other verbs are closely similar to each of the CB verbs?
  - Are there overlaps of the closest verbs between particular CB verbs?

- Exploring degree of similarity of the CB verbs to either cut_v and break_v
Whole collection of COCA corpus (the POS-tagged version) ([http://corpus.byu.edu/coca/](http://corpus.byu.edu/coca/))

Pre-processing steps:
1. Define words as consisting of alphabets [a-z], hyphens (to retain *machine-readable*), and single quote (to retain genitive ‘s and negation won’t)

2. Remove punctuation and numbers

3. Collapse various *Verb-tag labels* (e.g. for infinitive, participle, etc.) into simply “v”

4. Collapse *lemma* and *POS* columns into a single, big text

[1] "by_ii jill_np1 mccorkle_np1 anna_np1 craven_np1 have_vhz"
Methods

- Use the *wordVectors* R package by Ben Schmidt
  - Creates vector space model for every lemma in the COCA corpus
    - Reduces the original raw vectors into 100-dimensional vectors
    - On the basis of collocational window-span of 12 words (default in the `train_word2vec()` function)
  - Has a number of functions for exploring the vector space model
    - Finding nearest words to a particular target word
    - Computing similarity scores between words
    - *Inter alia* (cf. [https://github.com/bmschmidt/wordVectors](https://github.com/bmschmidt/wordVectors))
Methods

Exploring thematic cluster of Cut and Break verbs

- Analyse 22 *Cut and Break* verbs (cf. the plot below)
- Retrieve the vector space matrix of these verbs’ lemmas
- Compute distance matrix for the lemmas
- Perform *Hierarchical Cluster Analysis* (HCA)
  - with `hclust()` function in R
- Compute *Average Silhouette Width* (ASW) on the basis of the HCA results (cf. Levshina, 2015, p. 312)
  - To assume the optimal number of cluster solution
  - Compute ASW from 2 up to 21 clusters (i.e. N of CB verbs – 1)
  - For our data, 8-cluster solution produces the highest ASW score
- Visualise the results into a dendrogram
Results

*Exploring thematic cluster of Cut and Break verbs*

Hierarchical Cluster Analysis for the CUT and BREAK verbs
with 8-cluster solution

The optimal number of clusters is identified using the "Average Silhouette Width (ASW)" statistic.
The cluster solutions tested range from 2 up to 21 clusters; the 8-cluster solution produces the highest ASW score.
Similarity spaces – cut and slash
Similarity spaces – peel and slice
Similarity spaces – peel and slice
An alternative approach

Majid et al.:

\[ \text{event} \rightarrow \text{verb} \rightarrow \text{similarity space} \]

Musgrave et al.:

\[ \text{verb} \rightarrow \text{collocates} \rightarrow \text{similarity space} \]

To what extent do these converge?
Next steps

• Build models for:
  – German
  – Swedish
  – Dutch

• Comparison for Majid et al 2007

• Compare clusterings produced by two different approaches


Thank you

- Asifa Majid for sharing her data
- MonARCH (Monash Advanced Research Computing Hybrid), especially Philip Chan