

# Thesauri and Semantic Networks

1

## Overview

- Thesauri
  - Manual
  - Automatic
- Semantic Networks

2

## Thesauri

- It is intuitive to use one or more thesauri to expand a query.
- A query about “dogs” might well be expanded to include “canine” if a thesauri was consulted.
- Only problem is that you can easily add a “bad” word. A synonym for “dog” might well be “pet” and then the query would be too generic.

3

## Manual vs Automatic

- Manual
  - use a readily available machine-readable form of a thesauri (e.g. Roget’s, etc.)
- Automatic
  - build a thesarus automatically in a language independent fashion
  - Notion is that an algorithm that could build a thesaurus automatically could be used on many different languages.

4

## Automatic Thesauri Generation

- Two approaches (that we will describe -- others are in the book)
  - term-term co-occurrence (Salton 1971)
  - use context vectors (Gauch, 1996)

5

## Need SC for two terms

- With the Vector Space Model, we have a vector that represents the query and a vector that represents each document.
- The components of the vector are the list of terms.
- Essentially we have a DOC-TERM matrix that says, for each document what terms appear in the document.

6

## SC for terms

- For terms, we would like a vector for each term.
- Suppose we return to our language of two terms  $a$  and  $b$  and only use a binary vector space model. For a document  $D_1$  that contains only  $a$  and a document  $D_2$  that contains only  $b$  we would have:

$$D_1 \quad \langle 1 \ 0 \rangle$$

$$D_2 \quad \langle 0 \ 1 \rangle$$

- If we want term vectors, we now have a component for each *term*. This results in:

$$a \quad \langle 1 \quad 0 \rangle$$

$$b \quad \langle 0 \quad 1 \rangle$$

7

## SC for terms

- Usually we build document vectors and then to obtain an SC (Q, D) we treat the query as a document.
- With terms, we build term vectors and then measure the similarity between the two vectors.

8

## Components of Term vectors

- Good old tf-idf will work
- A newer one in 1995 by Chen and Ng that weights phrases higher.
- $(tf) (\log (N / df) p)$
- where  $p$  is the number of words in the phrase.

9

## Automatic Thesaurus with term-term vectors

- Simple dot product may be used.
- For a given word, we can compute the top  $t$  words related to this word.
- These words can now be used for query expansion.

10

## Term-term co-occurrence

- Build a term-term co-occurrence matrix. For each term show how often it appears in related documents.
- Premise here is that terms are related if they often appear in the same document.
- Now, for a given term we can define a similarity measure that will rank terms in order of their similarity to the term in question.
- So, for a term “dog” we might find that “canine” is the most similar.

11

## Problems with term-term co-occurrence

- A very frequent term will co-occur with everything
- Very general terms will co-occur with other general terms (*hairy* will co-occur with *furry*)
- A great paper by Smeaton (1983) showed random addition of words was sometimes more effective than expansion by term-term co-occurrence.

12

## Thesaurus Generation with Term Context

- Notion here is that term co-occurrence is nice, but many unrelated terms will co-occur.
- Proposed improvement is that words that are used *with similar context words* are similar.

13

## Context Words

- Consider
  - The **dog** ran up the hill
  - The **canine** ran down the hill.
- Hope is that we will find that “dog” and “canine” are synonyms because of the context words around them.

14

# Context Vectors

- Step 1
  - Identify context words that will be used
  - Identify target terms (terms that we want to build a thesaurus)
  - Select window of how many context words we care about. For a given target term, we are going to choose how many context words to the left and to the right we will watch. A window of size 3 says that we will watch context words at
    - -3, -2, -1, +1, +2, +3
  - Determine the weights for the components of the context vector
- Step 2
  - Build the context vectors
- Step 3
  - Compute the similarity between two context vectors

15

## Step 1: Choose Key Parameters

- Identify context words that will be used
  - pick the top 200 most common terms
- Identify target terms (terms that we want to build a thesaurus)
  - This is the hard part, we do not want too frequent as they will be vague, general terms, don't want too infrequent because they will not co-occur with anything. Need words that fall in the middle of the term distribution.
- Select window of how many context words we care about
  - Lets choose -3 to +3, six word window.

16

## Build Context Vectors

- Each vector consists of an element *for each context word for each position* in the term window.
- So if we have 200 context words and six positions (-3,-2,-1,+1,+2,+3) each vector will have 1200 components.

17

## Component Weights

- For a given context term  $j$  and target term  $t$ 
  - $a$  = total occurrences of term  $t$  in the collection
  - $b$  = total occurrences of term  $j$  in the collection
  - $c$  = total documents that contain the co-occurrence of term  $t$  and term  $j$
  - $w = \log(Nc / (a)(b) + 1)$
- Goal is to give term a high weight if co-occurrence is happening more than a random occurrence.

18

## Identifying Expansion Terms

- For each target term, identify its similarity to all other target terms using their context vectors.
- Expand target terms in the query using the top  $t$  most similar terms. Various thresholds for  $t$  can be used.

19

## Semantic Networks

- Build a network that shows, for each word its relationships to other words.
- For *dog* and *canine* a *synonym* arc would exist.
- To expand a query, find the word in the semantic network and follow the various arcs to other related words.
- Different *distance measures* can be used to compute the distance from one word in the network to another.
- Check out WordNet at Wordnet ([www.cogsci.princeton.edu/~wn](http://www.cogsci.princeton.edu/~wn))

20

## Types of Links in Wordnet

- Synonyms
  - dog, canine
- Antonyms (opposite)
  - night, day
- Hyponyms (is-a)
  - dog, mammal
- Meronyms (part-of)
  - roof, house
- Entailment (one entails the other)
  - buy, pay
- Troponyms (two words related by entailment must occur at the same time)
  - limp, walk

21

## Summary

- Pros
  - Thesauri and wordnet can be used to find good words for users “more like this”
- Cons
  - Little improvement has been found with automatic techniques to expand query without user intervention
  - Manual thesauri and Wordnet are language dependent

22