Intelligent Data Analysis

Week 9
Query Expansion

Martin Russell

Objectives

- Use **semantic** relationships between words to improve the performance of a text IR system
- Understand **Query Expansion**
- **Knowledge-driven** approaches
  - Synonyms
  - WordNet
- **Data-driven** approaches
  - Word vectors

Query Processing

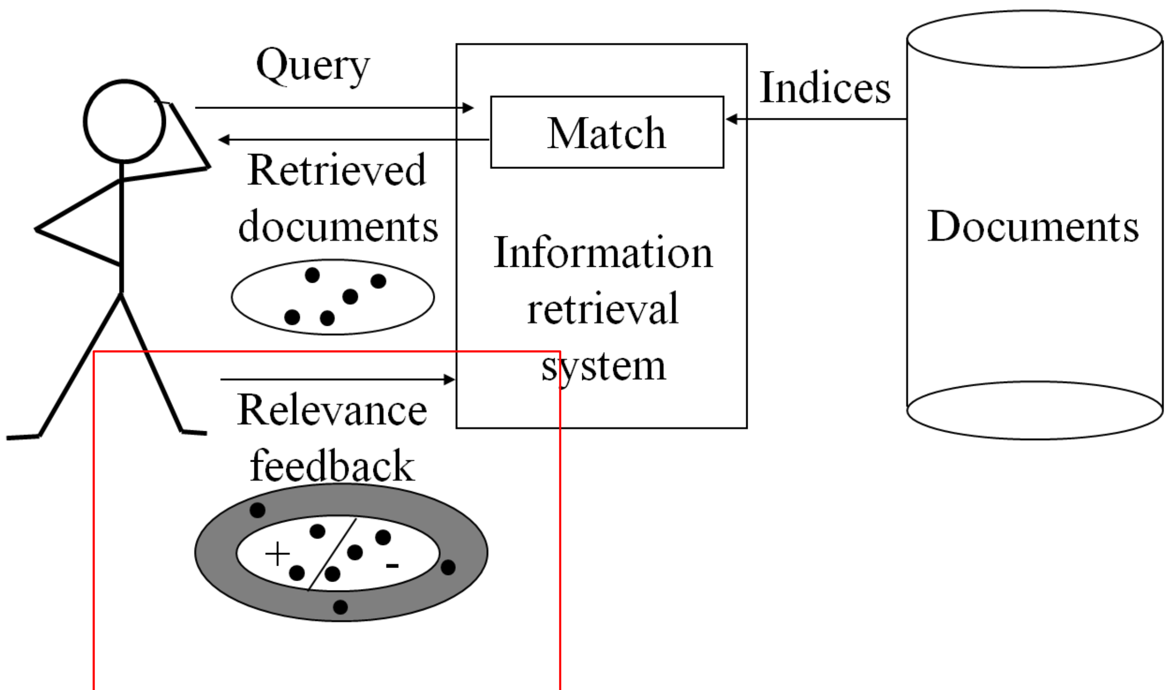
- Remember how we previously processed a query:
- Example:
  - "I need information about local physicians"
- Stop word removal
  - information, local, physician
- Stemming
  - info, local, physic
- But what about:
  - "This directory lists all of the doctors in the city …"

Query Expansion (1)

- Add terms to the query to increase the overlap between it and potentially relevant documents...
- ...but not irrelevant documents
- Two approaches:
  - Apply user feedback
  - Exploit semantic relationships between words

Feedback-based Query Expansion

- User provides feedback on the results of retrieval
  - Which of the returned documents are particularly relevant and which are irrelevant



Query reformulation

- Revise the query in response to user feedback
- Query **expansion**: Add terms in relevant documents that are not in query (or just those with large TF-IDF weights)
- Term **reweighting**: Increase the weight of query terms in relevant documents and decrease the weight of query terms in irrelevant documents. For example

$$w_{td} = \boxed{\lambda} \times f_{td} \times IDF(t)$$

- Various methods for determining λ proposed

Knowledge-Based Methods

- Remember:
  - *q* = "I need information about local physicians"
  - *d* = "This directory lists all of the doctors in the city …"
- We know there is a sematic relationship between
  – physician, and doctor
- Different words with same meaning are **synonyms**
- If w_1 is in q and w_1, w_2 are synonyms add w_2 to $\boldsymbol{q}$

Thesaurus

- A thesaurus is a 'dictionary' of synonyms and semantically related words and phrases
- E.G: Roget's Thesaurus
- Example:

physician

syn: || croaker, doc, doctor, MD, medical, mediciner, medico ||

rel: medic, general practitioner, surgeon

Peter Mark Roget 1779 –1869

- Born London 1779
- Founder of the Royal Society of Medicine
- Invented the log-log slide rule
- Professor of Physiology at the Royal Institution, 1834
- Retired 1840
- Roget's *Thesaurus of English Words and Phrases Classified and Arranged so as to Facilitate the Expression of Ideas and Assist in Literary Composition* appeared in 1852.
- Died 1869. Buried St James' Church, West Malvern, Worcestershire.


PETER MARK ROGET
M.D. F.R.S.
DIED AT WEST MALVERN
AGED 90

Other semantic relationships

- **Hyponyms** (subordinate words) ↓
  - Query $\boldsymbol{q}$ = "Tell me about England"
  - Document $\boldsymbol{d}$ = "A visit to London should be on everyone's itinerary"
  - 'London' is a **hyponym** of 'England'
- **Hypernyms** (generalisations) ↑
  - Query $\boldsymbol{q}$ = "Tell me about England"
  - Document $\boldsymbol{d}$ = "Places to visit in the British Isles"
  - 'British Isles' is a **hypernym** of 'England'

WordNet

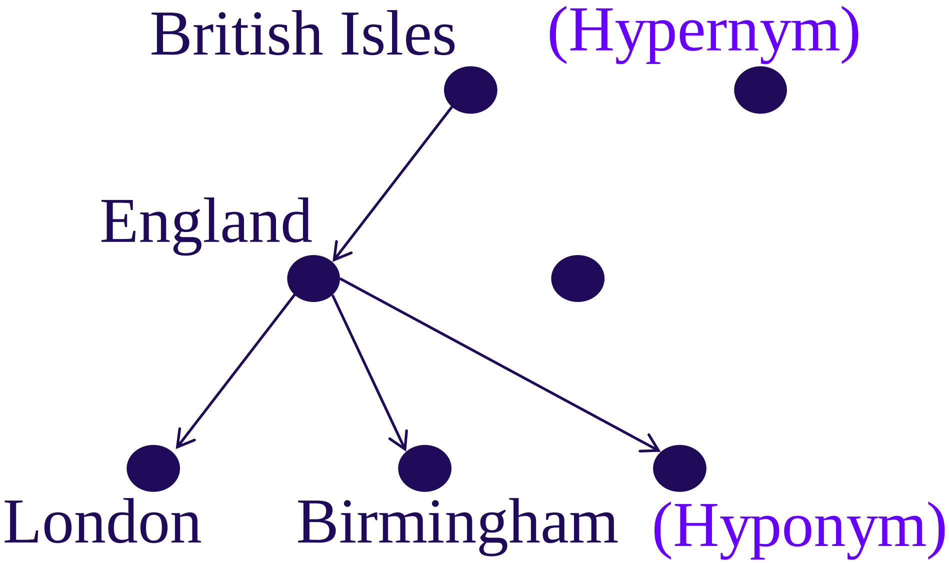
- WordNet is an online lexical database for English
- http://www.cogsci.princeton.edu/~wn

Category	*Forms*	*Meanings (syn sets)*
Nouns	57,000	48,800
Adjectives	19,500	10,000
Verbs	21,000	8,400

See Belew, chapter 6

WordNet

- Organised as a set of hierarchical trees
- For example, 25 trees for nouns
- 'Children' of a node are hyponyms
- Words become more specific as you move deeper into the tree



Summary

- Use **knowledge** (WordNet) to identify new words that are **semantically related** to query words
- Add these new words to the query

Query Expansion: Data Driven

- Let w be the n^{th} word in the vocabulary
- Can represent w as a "one hot" vector, $vec(w)$:

$$vec(w) = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

- $vec(w)_m = 1$ if $m=n$, 0 otherwise

LSA revisited

- Recall from LSA, $W = USV^T$, where the columns of V can be interpreted as topics
- Let $\boldsymbol{V}_{(\mathbf{n})}$ be the $V \times n$ matrix comprising first n columns of $\boldsymbol{V}$
- $top(w) = \boldsymbol{V}^{\mathbf{T}} vec(w)$ is a **topic-based** representation of $\boldsymbol{w}$
- $top(w) = \boldsymbol{V}_{(\mathbf{n})}{}^{\mathbf{T}} vec(w)$ is a reduced-dimensional **topic-based** representation of $\boldsymbol{w}$
- $top(w)$ represents a word w in terms of the **topics for which it is significant**

Vector representation of words

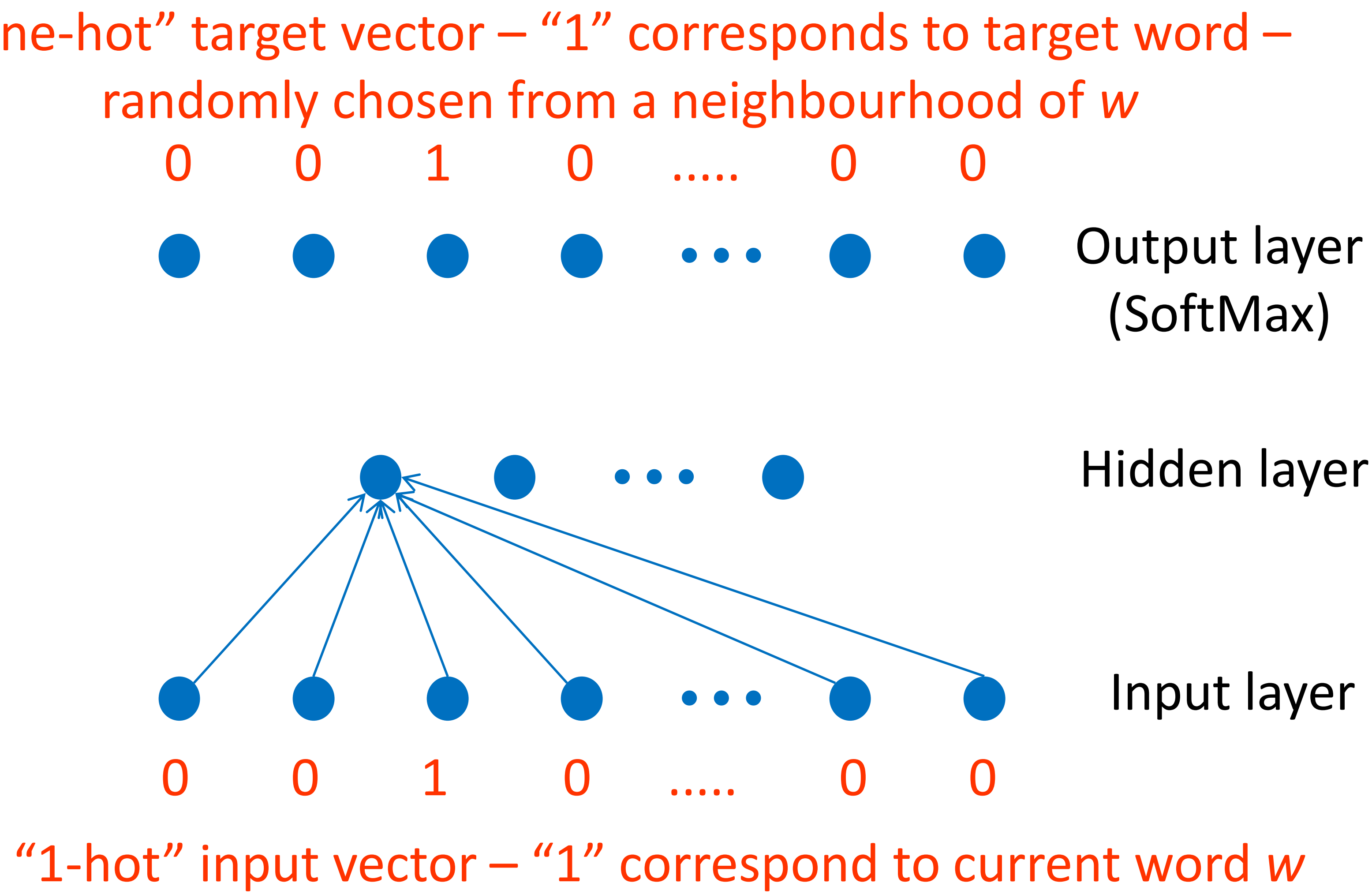
- Suppose u and w are words
- If u and w are synonyms they will be important for the same topics
- In this case $top(u)$ and $top(w)$ will point in similar directions
- Hence $Csim(u,w) = \dfrac{top(u) \cdot top(w)}{\|top(u)\| \|top(w)\|} = \cos(\theta)$

  is a **measure of the similarity** between u and w (θ is the angle between $top(u)$ and $top(w)$)
- If $Csim(u,w)$ sufficiently large treat u,w as synonyms for query expansion

Other approaches

- There are more modern approaches to vector representation of words and documents
- All based on the idea that if u,w occur in the context of the same set of words then they are related
- Google's *word2vec* uses *a* **Neural Network** to predict the neighbouring words (or the next word) in a document from the current (and previous) words

"word2vec"

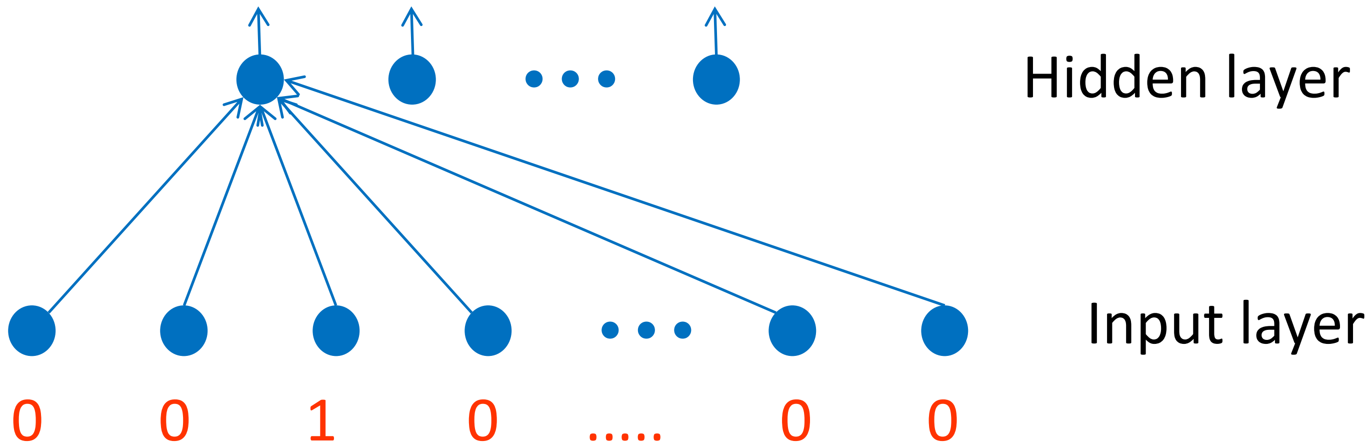

"one-hot" target vector – "1" corresponds to target word – randomly chosen from a neighbourhood of *w*
0 0 1 0 ..... 0 0
Output layer (SoftMax)
Hidden layer
Input layer
0 0 1 0 ..... 0 0
"1-hot" input vector – "1" correspond to current word *w*

What will the network learn?

- The network tries to map a word *w* onto each of the words that occur in a neighbourhood of *w* in documents
- The number of times that word *v* appears as a target output depends on the number of times *v* co-occurs with *w*
- For each w, the NN will learn the distribution of words that co-occur with w
- The values in the hidden layer are a low dimensional encoding of this relationship
- Values in the hidden layer when w is input is a low-dimensional encoding of the probability distribution of words that co-occur with *w*

"word2vec" (continued)

- Output of hidden layer is "word2vec" – a low dimensional representation of the word.
- (Assuming Num. Hidden Units << Num. Input Units)



"1-hot" input vector – "1" corresponds to current word

Summary so far

- Two approaches to identifying semantic relationships between words:
- **Knowledge-driven:** uses online thesaurus (e.g. WordNet)
- **Data-drive:** infer semantic relationships between words by converting words to vectors and measuring similarity between vectors (LSA, word2vec)
- **Query expansion:** augment query with new words related to query words

Query-document scoring

- Expand query $\boldsymbol{q}$ to include synonyms
- Recall that for a document $\boldsymbol{d}$

$$w_{td} = f_{td} \cdot IDF(t)$$

$$Sim(q,d) = \frac{\sum_{t \in q \cap d} w_{td} \cdot w_{tq}}{\|d\| \cdot \|q\|}$$

Query expansion

- Suppose:
  - t is the original term in the query,
  - t' is a synonym of t which occurs in d
- Then we could define:

$$w_{t'd} = \lambda_{tt'} \times f_{t'd} \times IDF(t) \qquad 0 \le \lambda_{tt'}$$

- Where $\lambda_{tt'}$ is a weighting depending on how 'far' t and t' are apart (according to WordNet or word-vector similarity)

Example

- Query q is:
  - *Is the Dark Knight on at the town cinema?*
  - q becomes: *dark knight town cinema*
- Document d is:
  - *The latest Batman movie places the caped crusader in a dark urban environment*
  - d becomes: *late batman move cape crusade dark urban environment*

Example (continued)

- In the similarity calculation, $q \cap d = \{dark\}$
- But:
  - *move* and *cinema* are synonyms (compare "go to the cinema" with "go to the movies")
  - *crusader* is a hyponym of *knight*
  - *urban* is a hypernym of *town*
- Therefore, after query expansion,

$$q \cap d = \{dark,\ move\ (syn(cinema)),\ crusade(hypo(knight)),\ urban(hyper(town))\}$$

Example (continued)

- As well as increasing the overlap between *q* and a **relevant** document *d*, may also increase the overlap with an **irrelevant** document
- Consider:

  The crusades were a dark period in our history when knights moved from across Europe to join crusades to the holy land
- This becomes:

  crusade dark period history knight move europe crusade holy land

Example (continued)

- In this case

$$q \cap d = \{dark, knight, move\ (syn(cinema)),$$
$$2 \times crusade(hypo(knight)),$$
$$urban(hyper(town)), land(hyper(town))\}$$

- Document may score higher similarity than previous one
- Challenge is:
  - Expand queries *enough* to promote overlap with relevant documents...
  - ...but not so much that they overlap with irrelevant documents

Summary

- Query expansion
  - Feedback-based
  - Knowledge-based: Synonyms, etc - WordNet
  - Data/(ML) – based approaches to synonym detection
- Goals:
  - Increase overlap with query and relevant documents,
  - And maintain separation from irrelevant documents
- Generalization