

COM6115: Text Processing

Information Retrieval: Task definition

Mark Hepple

Department of Computer Science
University of Sheffield

Overview

- Definition of the information retrieval problem
- Approaches to document indexing
 - ◊ manual approaches
 - ◊ automatic approaches
- Automated retrieval models
 - ◊ boolean model
 - ◊ ranked retrieval methods (e.g. vector space model)
- Term manipulation:
 - ◊ stemming, stopwords, term weighting
- Web Search Ranking
- Evaluation

Google search

jaguar

[Jaguar International - Market selector page](#)

www.jaguar.com/

Official worldwide web site of **Jaguar** Cars. Directs users to pages tailored to country-specific markets and model-specific websites.

[Jaguar International - Home](#)



www.jaguar.com/gi/en/

8 Jul 2009

Our mission at **Jaguar** has been to create and build beautiful fast cars. The XK, XF, and XJ bring the ...

[More videos for jaguar »](#)

[Jaguar Cars - Wikipedia, the free encyclopedia](#)

en.wikipedia.org/wiki/Jaguar_Cars

Jaguar Cars Ltd, known simply as **Jaguar** is a British luxury and sports car manufacturer, headquartered in Whitley, Coventry, England. It is part of the **Jaguar** ...

[Jaguar - Wikipedia, the free encyclopedia](#)

en.wikipedia.org/wiki/Jaguar

The **jaguar** is a big cat, a feline in the *Panthera* genus, and is the only *Panthera* species found in the Americas. The **jaguar** is the third-largest feline after the tiger ...

Google search (contd)

jaguar south america

[Jaguar - Wikipedia, the free encyclopedia](#)

en.wikipedia.org/wiki/Jaguar

The **jaguar's** present range extends from Southern United States and Mexico across much of Central **America** and **south** to Paraguay and northern Argentina.

[Jaguar Cars](#) - [Jaguar \(disambiguation\)](#) - [Jacksonville Jaguars](#) - [Jaguarundi](#)

[Images for jaguar south america](#) - Report images



[South America - Jaguar](#)

library.thinkquest.org/5053/SouthAmerica/jaguar.html

Jaguars are magnificent cats that prowl the **South American** jungles. They are fascinating to learn about! To jump to a section, use our Quick Jump below by ...

[Jaguars, Jaguar Pictures, Jaguar Facts - National Geographic](#)

animals.nationalgeographic.com/animals/mammals/jaguar/

Jaguars are the largest of **South America's** big cats. They once roamed from the southern tip of that continent north to the region surrounding the U.S.-Mexico ...

Google search (contd)

black fast jaguar

[Jaguars, Jaguar Pictures, Jaguar Facts - National Geographic](#)

animals.nationalgeographic.com/animals/mammals/jaguar/

Learn all you wanted to know about **jaguars** with pictures, videos, photos, facts, ... **Fast** Facts. Type: Mammal; Diet: Carnivore; Average life span in the wild: 12 to ... Most **jaguars** are tan or orange with distinctive **black** spots, dubbed "rosettes" ...

[Jaguar XKR black fast on trackday - YouTube](#)



www.youtube.com/watch?v...

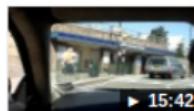
2 Jul 2012 - 16 sec - Uploaded by PrestigeCarCompany

Enjoy this **Jaguar** XKR trackday video.

<http://www.prestigecarcompany.co.uk/> Prestige Super Car Sales

...

[Jaguar XKR Black Pack fast racing - YouTube](#)



www.youtube.com/watch?v...k

16 Aug 2011 - 16 min - Uploaded by MrBobkumar

My XKR being driven hard thru town, hear the sounds of this beast....left window open so noise from air etc....but ...

[Jaguar XKR black fast on trackday. Rear Shot - YouTube](#)



www.youtube.com/watch?v=RmvW...

2 Jul 2012 - 17 sec - Uploaded by PrestigeCarCompany

Enjoy this **Jaguar** XKR trackday video.

<http://www.prestigecarcompany.co.uk/> Prestige Super Car Sales

...

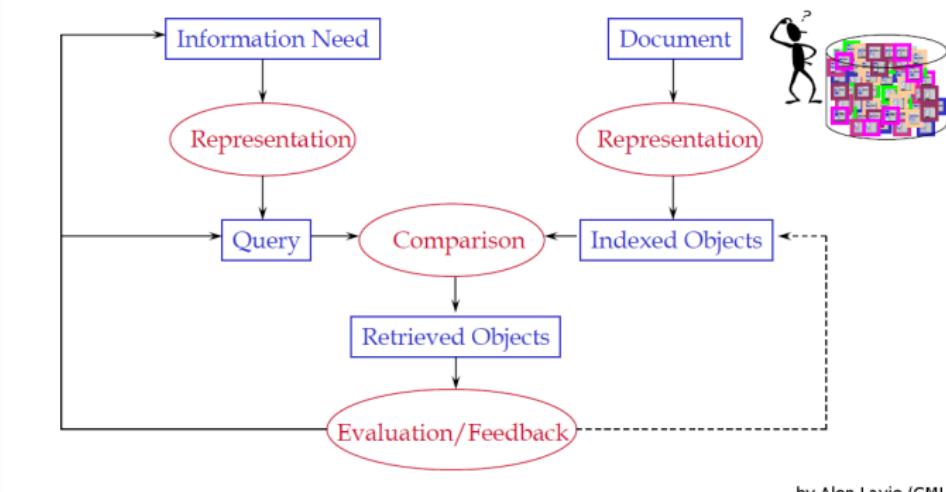
[More videos for black fast jaguar »](#)

Google search (contd)

- What is Google's IR system doing?
 - ◊ Finding pages that contain the words in the query.
- How does it rank these pages?
 - ◊ By “relevance” to the query.
- How does it do it so fast?
 - ◊ By clever indexing (and a lot of hardware!)

Information Retrieval: the task

Text Retrieval: find documents that are “relevant” to a user query.



by Alon Lavie (CMU)

- **Given:** a large, static document collection
- **Given:** an information need (keyword-based query)
- **Task:** find all and only documents relevant to query

Information Retrieval: the task

Typical IR systems:

- Search a set of abstracts
- Search newspaper articles
- Library search
- Search the Web

Typically:

- method more statistics than 'language'
- but the object to retrieve (and process) is language

- How can I formulate a query?
 - ◊ query type: normally keywords, could be natural language
- How are the documents represented?
 - ◊ indexing
- How does the system find the best-matching document?
 - ◊ retrieval model
- How does the system find it *efficiently*?
- How are the results presented to me?
 - ◊ unsorted list, ranked list, clusters
- How do we know whether the system is any good?
 - ◊ evaluation

Reading

- Baeza-Yates and Ribeiro-Neto, Modern Information Retrieval. New York: ACM Press, 1999.
- C. Manning, P. Raghavan and H. Schütze, Introduction to Information Retrieval, Cambridge University Press. 2008.
- I.H. Witten, A. Moffat and T.C. Bell, Managing Gigabytes: Compressing and Indexing Documents and Images, 2nd edition, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999.

COM6115: Text Processing

*Information Retrieval:
Document Indexing — Manual*

Mark Hepple

Department of Computer Science
University of Sheffield

Overview

- Definition of the information retrieval problem
- Approaches to document indexing
 - ◊ manual approaches
 - ◊ automatic approaches
- Automated retrieval models
 - ◊ boolean model
 - ◊ ranked retrieval methods (e.g. vector space model)
- Term manipulation:
 - ◊ stemming, stopwords, term weighting
- Web Search Ranking
- Evaluation

- How can I formulate a query?
 - ◊ query type: normally keywords, could be natural language
- How are the documents represented?
 - ◊ indexing
- How does the system find the best-matching document?
 - ◊ retrieval model
- How does the system find it *efficiently*?
- How are the results presented to me?
 - ◊ unsorted list, ranked list, clusters
- How do we know whether the system is any good?
 - ◊ evaluation

Indexing

The task of finding terms that describe documents well

- Manual:
 - ◊ indexing by humans (using fixed vocabularies)
 - ◊ labour and training intensive
- Automatic:
 - ◊ Term manipulation (certain words count as the same term)
 - ◊ Term weighting (certain terms are more important than others)
 - ◊ Index terms must only derive from text

Manual Indexing

- Large vocabularies (several thousand items)
 - ◊ Dewey Decimal System
 - ◊ Library of Congress Subject Headings
 - ◊ ACM – subfields of CS
 - ◊ MeSH – Medical Subject Headings

Example: Manual Indexing — ACM

ACM Computing Classification System (1998)

- B Hardware
 - B.3 Memory structures
 - B.3.0 General
 - B.3.1 Semiconductor Memories (NEW) (was B.7.1)
 - Dynamic memory (DRAM) (NEW)
 - Read-only memory (ROM) (NEW)
 - Static memory (SRAM) (NEW)
 - B.3.2 Design Styles (was D.4.2)
 - Associative memories
 - Cache memories
 - Interleaved memories
 - Mass storage (e.g., magnetic, optical, RAID)
 - Primary memory
 - Sequential-access memory

Example: Manual Indexing — MeSH

MeSH — Medical Subject Headings

- a very large *controlled vocabulary* for describing/indexing medical documents, e.g. journal papers and books
- provides a *hierarchy* of **descriptors** (a.k.a. *subject headings*)
 - ◊ assigned to documents to describe their content
- hierarchy has a number of *top-level* categories, e.g.:
 - ◊ Anatomy [A]
 - ◊ Organisms [B]
 - ◊ Diseases [C]
 - ◊ Chemicals and Drugs [D]
 - ◊ Analytical, Diagnostic and Therapeutic Techniques and Equipment [E]
 - ◊ Psychiatry and Psychology [F]
 - ◊ Biological Sciences [G]

...

Example: Manual Indexing — MeSH (contd)

- And a number of subcategories (more specific/detailed terms):

- Diseases [C]
 - MeSH [C01](#) --- bacterial infections and mycoses
 - MeSH [C02](#) --- virus diseases
 - MeSH [C03](#) --- parasitic diseases
 - MeSH [C04](#) --- neoplasms
 - MeSH [C05](#) --- musculoskeletal diseases
 - MeSH [C06](#) --- digestive system diseases
 - MeSH [C07](#) --- stomatognathic diseases
 - MeSH [C08](#) --- respiratory tract diseases
 - MeSH [C09](#) --- otorhinolaryngologic diseases
 - MeSH [C10](#) --- nervous system diseases
 - MeSH [C11](#) --- eye diseases
 - MeSH [C12](#) --- urologic and male genital diseases
 - MeSH [C13](#) --- female genital diseases and pregnancy complications
 - MeSH [C14](#) --- cardiovascular diseases

Example: Manual Indexing — MeSH (contd)

- And a number of subsubcategories (even more specific/detailed terms):

[Eye Diseases \[C11\]](#)

[Asthenopia \[C11.093\]](#)

► [Conjunctival Diseases \[C11.187\]](#)

[Conjunctival Neoplasms \[C11.187.169\]](#)

[Conjunctivitis \[C11.187.183\]](#) +

[Pterygium \[C11.187.781\]](#)

[Xerophthalmia \[C11.187.810\]](#)

[Corneal Diseases \[C11.204\]](#) +

[Eye Abnormalities \[C11.250\]](#) +

[Eye Diseases, Hereditary \[C11.270\]](#) +

[Eye Hemorrhage \[C11.290\]](#) +

[Eye Infections \[C11.294\]](#) +

Example: Manual Indexing — MeSH (contd)

- And a number of subsubsubcategories (yet again more specific/detailed terms):

Eye Diseases [C11]

Conjunctival Diseases [C11.187]

Conjunctival Neoplasms [C11.187.169]

► Conjunctivitis [C11.187.183]

Conjunctivitis, Allergic [C11.187.183.200]

Conjunctivitis, Bacterial [C11.187.183.220] +

Conjunctivitis, Viral [C11.187.183.240] +

Keratoconjunctivitis [C11.187.183.394] +

Reiter Syndrome [C11.187.183.749]

Pterygium [C11.187.781]

Xerophthalmia [C11.187.810]

Example: Manual Indexing — MeSH (contd)

- MEDLINE — Medical Literature Analysis and Retrieval System Online
 - ◊ international database of literature for medicine and the life sciences
 - ◊ includes papers from ≈5600 different sources (mostly journals), in various languages
 - ◊ database now holds records for ≈26 million papers
- Each MEDLINE article indexed with 10-15 descriptors from MeSH
 - ◊ papers accessed by PubMed search engine interface, using MeSH terms (and other terms, e.g. author name, etc)
 - ◊ by default, all descriptors below a given one in the hierarchy are also included in search

Manual Indexing

- Advantages:
 - ◊ High precision searches
 - ◊ Works well for closed collections (books in a library)
- Problems:
 - ◊ Searchers need to know terms to achieve high precision
 - ◊ Labellers need to be trained to achieve consistency
 - Not feasible to expect this from all content creators on the web
 - ◊ Collections are dynamic → schemes change constantly

Reading

- Baeza-Yates and Ribeiro-Neto, Modern Information Retrieval. New York: ACM Press, 1999.
- C. Manning, P. Raghavan and H. Schütze, Introduction to Information Retrieval, Cambridge University Press. 2008.
- I.H. Witten, A. Moffat and T.C. Bell, Managing Gigabytes: Compressing and Indexing Documents and Images, 2nd edition, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999.

COM6115: Text Processing

*Information Retrieval:
Document Indexing — Automatic*

Mark Hepple

Department of Computer Science
University of Sheffield

Overview

- Definition of the information retrieval problem
- Approaches to document indexing
 - ◊ manual approaches
 - ◊ automatic approaches
- Automated retrieval models
 - ◊ boolean model
 - ◊ ranked retrieval methods (e.g. vector space model)
- Term manipulation:
 - ◊ stemming, stopwords, term weighting
- Web Search Ranking
- Evaluation

Automatic Indexing

- No predefined set of *index terms*
- Instead: use **natural language** as indexing language
- Words in the document give information about its content
- Implementation of indices: **inverted files**
- This is what Google's IR system does
 - ◊ at least, it's an important **part** of the story

Automatic Indexing

- A small collection of documents ...

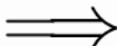
<i>Document</i>	<i>Text</i>
1	Pease porridge hot , pease porridge cold
2	Pease porridge in the pot
3	Nine days old
4	Some like it hot , some like it cold
5	Some like it in the pot
6	Nine days old

- Say we want to search for word **hot**. How do we do it?

Inverted files

- A basic inverted file index
 - ◊ records for each term, the ids of the documents in which it appears
 - ◊ only matters if it *does* or *does not* appear – not how many times

Doc	Text
1	Pease porridge hot, pease porridge cold
2	Pease porridge in the pot
3	Nine days old
4	Some like it hot, some like it cold
5	Some like it in the pot
6	Nine days old

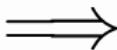


Num	Token	Docs
1	cold	1, 4
2	days	3, 6
3	hot	1, 4
4	in	2, 5
5	it	4, 5
6	like	4, 5
7	nine	3, 6
8	old	3, 6
9	pease	1, 2
10	porridge	1, 2
11	pot	2, 5
12	some	4, 5
13	the	2, 5

Inverted files (contd)

- A more sophisticated version ...
 - ◊ also record count of occurrences within each document
 - ◊ help find documents *more relevant* to query

Doc	Text
1	Pease porridge hot, pease porridge cold
2	Pease porridge in the pot
3	Nine days old
4	Some like it hot, some like it cold
5	Some like it in the pot
6	Nine days old

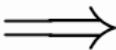


Num	Token	Docs
1	cold	1:1, 4:1
2	days	3:1, 6:1
3	hot	1:1, 4:1
4	in	2:1, 5:1
5	it	4:2, 5:1
6	like	4:2, 5:1
7	nine	3:1, 6:1
8	old	3:1, 6:1
9	pease	1:2, 2:1
10	porridge	1:2, 2:1
11	pot	2:1, 5:1
12	some	4:2, 5:1
13	the	2:1, 5:1

Inverted files (contd)

- A more sophisticated version . . .
 - ◊ also record *position* of each term occurrence within documents
 - ◊ may be useful for searching for **phrases** in documents

Doc	Text
1	Pease porridge hot, pease porridge cold
2	Pease porridge in the pot
3	Nine days old
4	Some like it hot, some like it cold
5	Some like it in the pot
6	Nine days old



Num	Token	Docs
1	cold	1:(6), 4:(8)
2	days	3:(2), 6:(2)
3	hot	1:(3), 4:(4)
4	in	2:(3), 5:(4)
5	it	4:(3, 7), 5:(3)
6	like	4:(2, 6), 5:(2)
7	nine	3:(1), 6:(1)
8	old	3:(3), 6:(3)
9	pease	1:(1, 4), 2:(1)
10	porridge	1:(2, 5), 2:(2)
11	pot	2:(5), 5:(6)
12	some	4:(1, 5), 5:(1)
13	the	2:(4), 5:(5)

Reading

- Baeza-Yates and Ribeiro-Neto, Modern Information Retrieval. New York: ACM Press, 1999.
- C. Manning, P. Raghavan and H. Schütze, Introduction to Information Retrieval, Cambridge University Press. 2008.
- I.H. Witten, A. Moffat and T.C. Bell, Managing Gigabytes: Compressing and Indexing Documents and Images, 2nd edition, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999.

COM6115: Text Processing

*Information Retrieval:
Retrieval models — boolean approach*

Mark Hepple

Department of Computer Science
University of Sheffield

Overview

- Definition of the information retrieval problem
- Approaches to document indexing
 - ◊ manual approaches
 - ◊ automatic approaches
- Automated retrieval models
 - ◊ boolean model
 - ◊ ranked retrieval methods (e.g. vector space model)
- Term manipulation:
 - ◊ stemming, stopwords, term weighting
- Web Search Ranking
- Evaluation

Bag-of-Words Approach

- Standard approach to representing documents (and queries) in IR:
 - ◊ record what words (terms) are present
 - ◊ usually, plus count of term in each document
- Ignores relations between words
 - ◊ i.e. of order, proximity, etc
 - ◊ e.g. rabbit eating = eating rabbit



- Such representations known as **bag of words** approaches
 - ◊ c.f. mathematical structure “bag”
 - like a set (i.e. unordered), but records a count for each element

Information Retrieval: Methods

- Boolean search:
 - ◊ binary decision: is document relevant or not?
 - ◊ presence of term is necessary and sufficient for match
 - ◊ boolean operators are set operations (AND, OR)
- Ranked algorithms:
 - ◊ frequency of document terms
 - ◊ not all search terms necessarily present in document
 - ◊ Incarnations:
 - The vector space model (SMART, Salton et al, 1971)
 - The probabilistic model (OKAPI, Robertson/Spärck Jones, 1976)
 - Web search engines

The Boolean model

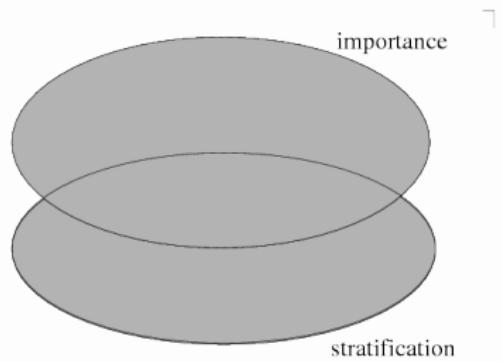
- Approach: construct *complex search commands*, by
 - ◊ combining *basic* search terms (keywords)
 - ◊ using *boolean operators*
- *Boolean Operators*:
 - ◊ AND, OR, NOT, BUT, XOR (*exclusive* OR)
- E.g.:
`Monte-Carlo AND (importance OR stratification) BUT gambling`
- Boolean query provides a simple logical basis for deciding whether any document should be returned, based on:
 - ◊ whether basic terms of query do/do not appear in the document
 - ◊ the meaning of the logical operators

The Boolean model: set-theoretic interpretation

- Boolean operators have a **set-theoretic interpretation** for **efficient** retrieval
- Overall document collection forms **maximal document set**
- let $d(E)$ denote the document set for expression E
 - ◊ E either a basic term or boolean expression
- Boolean operators map to set-theoretic operations:
 - ◊ AND $\mapsto \cap$ (intersection): $d(E_1 \text{ AND } E_2) = d(E_1) \cap d(E_2)$
 - ◊ OR $\mapsto \cup$ (union): $d(E_1 \text{ OR } E_2) = d(E_1) \cup d(E_2)$
 - ◊ NOT $\mapsto {}^c$ (complement): $d(\text{NOT } E) = d(E)^c$
 - ◊ BUT $\mapsto -$ (difference): $d(E_1 \text{ BUT } E_2) = d(E_1) - d(E_2)$

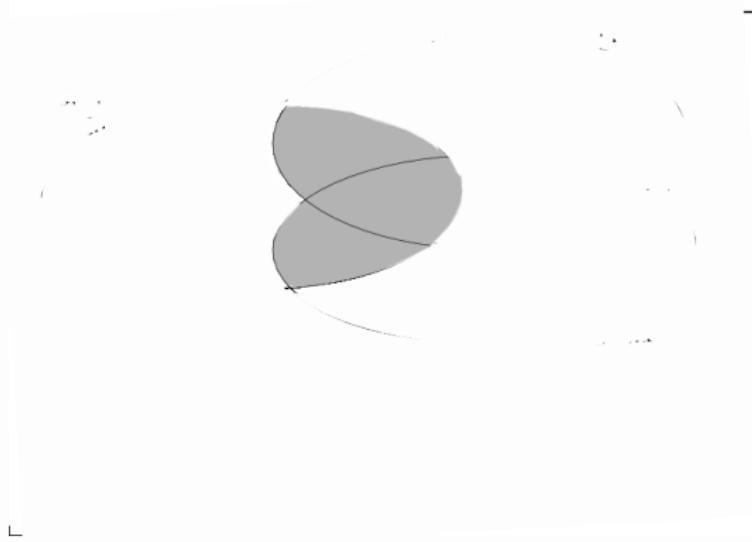
The Boolean model: set-theoretic interpretation (contd)

E.g. Monte-Carlo AND (importance OR stratification) BUT gambling



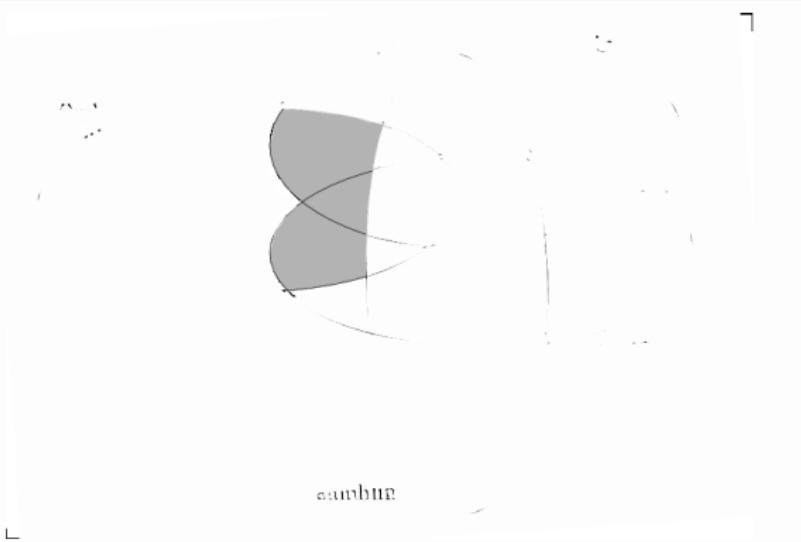
The Boolean model: set-theoretic interpretation (contd)

E.g. Monte-Carlo AND (importance OR stratification) BUT gambling



The Boolean model: set-theoretic interpretation (contd)

E.g. Monte-Carlo AND (importance OR stratification) BUT gambling



Boolean Queries: Complexity

- Question: Magnetic resonance imaging, magnetic resonance arthrography and ultrasonography for assessing rotator cuff tears in people with shoulder pain for whom surgery is being considered
- Query: ((Ultrasonography [mh] OR ultrasound [tw] OR ultrasonograph* [tw] OR sonograp* [tw] OR us [sh]) OR (Magnetic Resonance Imaging [mh] OR MR imag* [tw] OR magnetic resonance imag* [tw] OR MRI [tw])) AND (Rotator Cuff [mh] OR rotator cuff* [tw] OR musculotendinous cuff* [tw] OR subscapularis [tw] OR supraspinatus [tw] OR infraspinatus OR teres minor [tw])) AND (Rupture [mh:noexp] OR tear* [tw] OR torn [tw] OR thickness [tw] OR lesion* [tw] OR ruptur* [tw] OR injur* [tw])

From Lenza, M., Buchbinder, R., Takwoingi, Y., Johnston, R. V., Hanchard, N. C., & Faloppa, F. (2013). Magnetic resonance imaging, magnetic resonance arthrography and ultrasonography for assessing rotator cuff tears in people with shoulder pain for whom surgery is being considered. The Cochrane Library.

The Boolean model: summary

- Documents either match or don't match
 - ◊ Expert knowledge needed to create high-precision queries → OK for expert users
 - ◊ Often used by bibliographic search engines (library)
- Not good for the majority of users
 - ◊ Most users not familiar with writing Boolean queries → not natural
 - ◊ Most users don't want to wade through lists of 1000s unranked results → unless very specific search in small collections
 - ◊ This is particularly true of web search → large set of docs

COM6115: Text Processing

*Information Retrieval:
retrieval models — ranked retrieval methods*

Mark Hepple

Department of Computer Science
University of Sheffield

Overview

- Definition of the information retrieval problem
- Approaches to document indexing
 - ◊ manual approaches
 - ◊ automatic approaches
- Automated retrieval models
 - ◊ boolean model
 - ◊ ranked retrieval methods (e.g. vector space model)
- Term manipulation:
 - ◊ stemming, stopwords, term weighting
- Web Search Ranking
- Evaluation

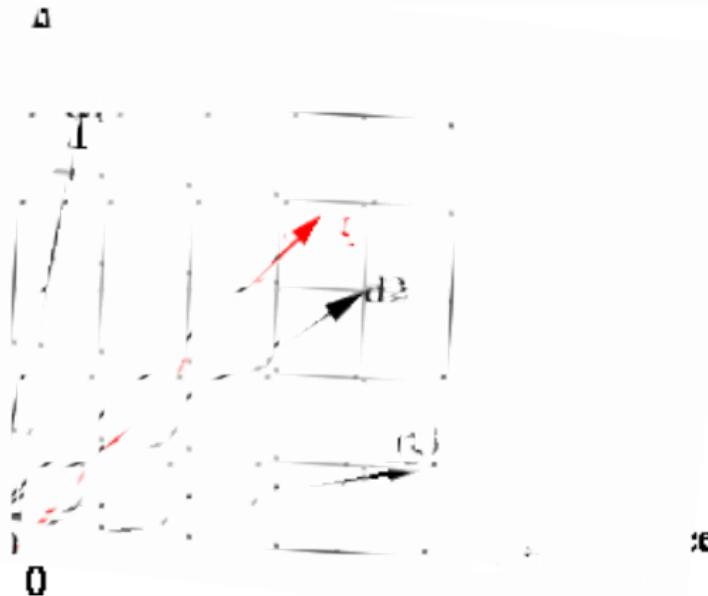
The Vector Space model

- Documents are also represented as “bags of words”:
 - ◊ “John is quicker than Mary” = “Mary is quicker than John”
- Documents are points in **high-dimensional** vector space
 - ◊ each term in index is a dimension → sparse vectors
 - ◊ values are **frequencies** of terms in documents, or variants of frequency
- Queries are also represented as vectors (for terms that exist in index)
- Approach
 - ◊ Select document(s) with highest document–query similarity
 - ◊ Document–query similarity is a model for relevance (ranking)
 - ◊ With ranking, **the number of returned documents is less relevant** → users start at the top and stop when satisfied

The Vector Space model (contd)

2 dimensions:

Query: car insurance



The Vector Space Model (contd)

- Approach: compare vector of **query** against vector of each **document**
 - to rank documents according to their **similarity** to the query

	Term ₁	Term ₂	Term ₃	...	Term _n
Doc ₁	9	0	1	...	0
Doc ₂	0	1	0	...	10
Doc ₃	0	1	0	...	2
...
Doc _N	4	7	0	...	5

Q	0	1	0	...	1
---	---	---	---	-----	---

How to measure similarity between vectors?

- Each document and the query are represented as a vector of n values:

$$\vec{d}^i = (d_1^i, d_2^i, \dots, d_n^i), \quad \vec{q} = (q_1, q_2, \dots, q_n)$$

- Many metrics of similarity between 2 vectors, e.g.: [Euclidean](#)

$$\sqrt{\sum_{k=1}^n (q_k - d_k)^2}$$

- E.g.: Distance between:

$$Doc_1 \text{ and } Q = \sqrt{(9-0)^2 + (0-1)^2 + (1-0)^2 + (0-1)^2} = \sqrt{84} = 9.15$$

$$Doc_2 \text{ and } Q = \sqrt{(0-0)^2 + (1-1)^2 + (0-0)^2 + (10-1)^2} = \sqrt{81} = 9$$

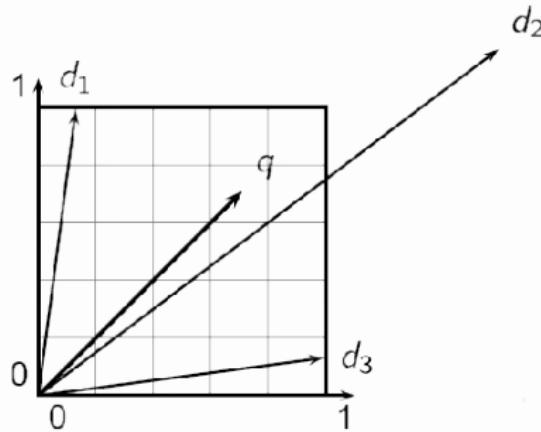
$$Doc_3 \text{ and } Q = \sqrt{(0-0)^2 + (1-1)^2 + (0-0)^2 + (2-1)^2} = \sqrt{1} = 1$$

Doc 3 is the closest (shortest distance)

How to measure similarity between vectors? (contd)

Is it a good idea?

- distance is large for vectors of different lengths, even if by only one term (e.g. Doc_2 and Q)
- means frequency of terms given *too much impact*



How to measure similarity between vectors? (contd)

- Better **similarity** metric, used in *vector-space* model:
cosine of the **angle** between two vectors \vec{x} and \vec{y} :

$$\cos(\vec{x}, \vec{y}) = \frac{\vec{x} \cdot \vec{y}}{|\vec{x}| |\vec{y}|} = \frac{\sum_{i=1}^n x_i y_i}{\sqrt{\sum_{i=1}^n x_i^2} \sqrt{\sum_{i=1}^n y_i^2}}$$

- It can be interpreted as the **normalised correlation coefficient**:
i.e. it computes how well the x_i and y_i correlate, and then divides by the length of the vectors, to scale for their magnitude
 - ◊ The vector \vec{x} is normalised by dividing its components by its length:

$$|\vec{x}| = \sqrt{\sum_{i=1}^n x_i^2}$$

How to measure similarity between vectors? (contd)

- The cosine value ranges from:
 - ◊ 1, for vectors pointing in the same direction, to
 - ◊ 0, for orthogonal vectors, to
 - ◊ -1, for vectors pointing in opposite directions
- Specialising the equation to comparing a query q and document d :

$$\text{sim}(\vec{q}, \vec{d}) = \cos(\vec{q}, \vec{d}) = \frac{\sum_{i=1}^n q_i d_i}{\sqrt{\sum_{i=1}^n q_i^2} \sqrt{\sum_{i=1}^n d_i^2}}$$

i.e. computes how well occurrences of each term i correlate in query and document, then scales for the magnitude of the overall vectors

COM6115: Text Processing

*Information Retrieval:
Term Manipulation*

Mark Hepple

Department of Computer Science
University of Sheffield

Overview

- Definition of the information retrieval problem
- Approaches to document indexing
 - ◊ manual approaches
 - ◊ automatic approaches
- Automated retrieval models
 - ◊ boolean model
 - ◊ ranked retrieval methods (e.g. vector space model)
- Term manipulation:
 - ◊ stemming, stopwords, term weighting
- Web Search Ranking
- Evaluation

What counts as a term?

Common to just use the **words**, but pre-process them for generalisation

- **Tokenisation**: split words from punctuation (get rid of punctuation)
e.g. word-based. → word based three issues: → three issues
- **Capitalisation**: normalise all words to lower (or upper) case
e.g. Cat and cat should be seen as the same term, but should we conflate Turkey and turkey?
- **Lemmatisation**: conflate different inflected forms of a word to their basic form (singular, present tense, 1st person):
e.g. cats, cat → cat have, has, had → have worried, worries → worry

What counts as a term? (ctd)

- **Stemming**: conflate morphological variants by chopping their affix:

CONNECT
CONNECTED
CONNECTING
CONNECTION
CONNECTIONS

WORRY
WORRIED
WORRIES
WORRYING
WORRYINGLY

GALL
GALLING
GALLED
GALLEY
GALLERY

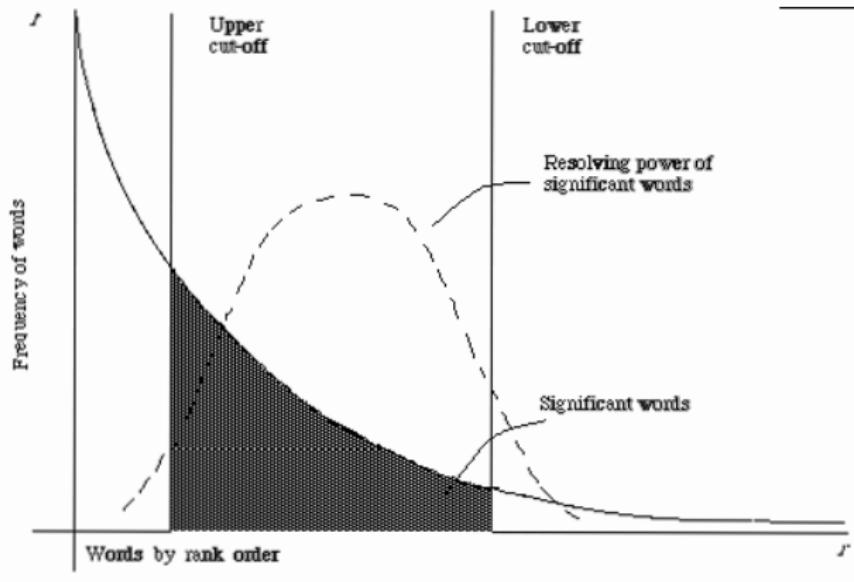
- **Normalisation**: heuristics to conflate variants due to spelling, hyphenation, spaces, etc.

e.g. USA and U.S.A. and U S A → USA

e.g. chequebook and cheque book → cheque book

e.g. word-sense and word sense → word-sense

Word Frequency and Term Usefulness



- The most and least frequent terms are not the most useful for retrieval
 - ◊ (Figure from van Rijsbergen (1979) *Information Retrieval*
<http://www.dcs.gla.ac.uk/Keith/Preface.html>)

Stop words

- Use **Stop list** removal to exclude “non-content” words
- Usually most frequent (and least useful for retrieval)

a	always	both
about	am	being
above	among	co
across	amongst	could

- ◊ greatly reduces the size of the inverted index
- ◊ but what if we want to search for *phrases* that include these terms?
 - Kings of Leon
 - Let it be
 - To be or not to be
 - Flights to London

Single vs. Multi-word Terms

- To aid recognition of **phrases**, might allow ***multi-word terms***
e.g. **Sheffield University**
- Possible approach — allow ***multi-word indexing***
e.g. bigram indexing: store each bigram as a term in index

For **pease porridge in the pot** get:

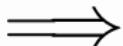
pease porridge
porridge in
in the
the pot

- ◊ Problem: number of bigrams is v.large c.f. number of words
 - leads to a huge increase in size of the index
- Alternative: identify multi-word phrases during retrieval
 - ◊ **Positional indexes**, storing position terms in documents, can help
 - use to compute if occurrences of search terms in document are adjacent / close / far apart

Single vs. Multi-word Terms (ctd)

- Positional indexes:

Doc	Text
1	Pease porridge hot, pease porridge cold
2	Pease porridge in the pot
3	Nine days old
4	Some like it hot, some like it cold
5	Some like it in the pot
6	Nine days old



Num	Token	Docs
1	cold	1:(6), 4:(8)
2	days	3:(2), 6:(2)
3	hot	1:(3), 4:(4)
4	in	2:(3), 5:(4)
5	it	4:(3, 7), 5:(3)
6	like	4:(2, 6), 5:(2)
7	nine	3:(1), 6:(1)
8	old	3:(3), 6:(3)
9	pease	1:(1, 4), 2:(1)
10	porridge	1:(2, 5), 2:(2)
11	pot	2:(5), 5:(6)
12	some	4:(1, 5), 5:(1)
13	the	2:(4), 5:(5)

COM6115: Text Processing

Information Retrieval: Term Weighting

Mark Hepple

Department of Computer Science
University of Sheffield

Overview

- Definition of the information retrieval problem
- Approaches to document indexing
 - ◊ manual approaches
 - ◊ automatic approaches
- Automated retrieval models
 - ◊ boolean model
 - ◊ ranked retrieval methods (e.g. vector space model)
- Term manipulation:
 - ◊ stemming, stopwords, term weighting
- Web Search Ranking
- Evaluation

Term Weighting

What values do we assign for terms in document (and query) vectors?

- **binary weights - 0/1:** whether or not term is present in document
 - ◊ But documents with multiple occurrences of query keyword may be more relevant
- **Frequency of term in document:** like the examples we have seen
 - ◊ But what if the term is also **frequent in collection?**
 - ◊ Common terms: not very useful for discriminating relevant documents
- **Frequency in document vs in collection:** weight terms highly if
 - ◊ They are **frequent** in relevant documents ... *but*
 - ◊ They are **infrequent** in collection as a whole

Term Weighting (ctd)

- Key concepts:

document collection	D	collection (set) of documents
size of collection	$ D $	total number of documents in collection
term freq	$tf_{w,d}$	number of times w occurs in document d
collection freq	cf_w	number of times w occurs in collection
document freq	df_w	number of documents containing w

Term Weighting (ctd)

The informativeness of terms

- Idea that *less common* terms are *more useful* to finding relevant docs:
i.e. these terms are more *informative*
- Is this idea best addressed using *document frequency* or *collection frequency*?
- Consider following counts (from New York Times data, $|D| = 10000$):

Word	cf_w	df_w
insurance	10440	3997
try	10422	8760

- ◊ term *insurance* semantically focussed, term *try* very general
 - document frequency reflects this difference
 - collection frequency fails to distinguish them (i.e. very similar counts)

Term Weighting (ctd)

- Informativeness is **inversely related** to (document) frequency
 - i.e. *less common* terms are *more useful* to finding relevant documents
 - more common* terms are *less useful* to finding relevant documents
- Compute metric such as: $\frac{|D|}{df_w}$
 - ◊ Value reduces as df_w gets larger, tending to 1 as df_w approaches $|D|$
e.g. $\frac{10000}{3997} = 2.5$ (insurance) $\frac{10000}{8760} = 1.14$ (try)
 - ◊ Value very large for small df_w — **over-weights** such cases
e.g. $\frac{10000}{350} = 28.6$ (mischief)
- To moderate this, take \log : **Inverse document frequency** (idf)

$$idf_{w,D} = \log \frac{|D|}{df_w}$$

$$\log \frac{10000}{3997} = 0.398 \text{ (insurance)} \quad \log \frac{10000}{8760} = 0.057 \text{ (try)} \quad \log \frac{10000}{350} = 1.456 \text{ (mischief)}$$

Term Weighting (ctd)

- **BUT** Not all terms describe a document equally well
- Putting it all together: **tf.idf**
 - ◊ Terms which are frequent in a document are better:

$$tf_{w,d} = freq_{w,d}$$

- ◊ Terms that are rare in the document collection are better:

$$idf_{w,D} = \log \frac{|D|}{df_w}$$

- ◊ Combine the two to give **tf.idf** term weighting:

$$tf.idf_{w,d,D} = tf_{w,d} \cdot idf_{w,D}$$

- Most commonly used method for term weighting.
 - ◊ Used in other fields too (e.g. summarisation)

Term Weighting (ctd)

tf.idf example:

Term	<i>tf</i>	<i>df</i>	<i>D</i>	<i>idf</i>	<i>tf.idf</i>
the	312	28,799	30,000	0.018	5.54
in	179	26,452	30,000	0.055	9.78
general	136	179	30,000	2.224	302.50
fact	131	231	30,000	2.114	276.87
explosives	63	98	30,000	2.486	156.61
nations	45	142	30,000	2.325	104.62
haven	37	227	30,000	2.121	78.48

For term *the*:

$$idf(\text{the}) = \log_{10}\left(\frac{30,000}{28,799}\right) = 0.018$$

$$tf.idf(\text{the}) = 312 \cdot 0.018 = 5.54$$

Putting things together

Example: Vector Space Model, tf.idf term weighting, cosine similarity

- tf.idf values for words in two documents D_1 and D_2 , and in a query Q
“**hunter gatherer Scandinavia**”:

	Q	D_1	D_2
hunter	19.2	56.4	112.2
gatherer	34.5	122.4	0
Scandinavia	13.9	0	30.9
30,000	0	457.2	0
years	0	12.4	0
BC	0	200.2	0
prehistoric	0	45.3	0
deer	0	0	23.6
rifle	0	0	452.2
Mesolithic	0	344.2	0
$\sqrt{\sum_{i=1}^n x_i^2}$	41.9	622.9	467.5

(i.e. length of vector)

Putting things together (ctd)

- $\text{sim}(\vec{q}, \vec{d}) = \cos(\vec{q}, \vec{d}) = \frac{\sum_{i=1}^n q_i d_i}{\sqrt{\sum_{i=1}^n q_i^2} \sqrt{\sum_{i=1}^n d_i^2}}$

$$\begin{aligned}\cos(Q, D_1) &= \frac{(19.2 * 56.4) + (34.5 * 122.4) + \dots + (0 * 0) + (0 * 344.2)}{41.9 * 622.9} \\ &= \frac{5305.68}{26071.72} \\ &= 0.20\end{aligned}$$

$$\begin{aligned}\cos(Q, D_2) &= \frac{(19.2 * 112.2) + (34.5 * 0) + \dots + (0.0 * 452.2) + (0.0 * 0.0)}{41.9 * 467.5} \\ &= \frac{2583.8}{19570.0} \\ &= 0.13\end{aligned}$$

- so document D_1 is more similar to Q than D_2

COM6115: Text Processing

*Information Retrieval:
Web Search Ranking*

Mark Hepple

Department of Computer Science
University of Sheffield

Overview

- Definition of the information retrieval problem
- Approaches to document indexing
 - ◊ manual approaches
 - ◊ automatic approaches
- Automated retrieval models
 - ◊ boolean model
 - ◊ ranked retrieval methods (e.g. vector space model)
- Term manipulation:
 - ◊ stemming, stopwords, term weighting
- Web Search Ranking
- Evaluation

Web Search Ranking

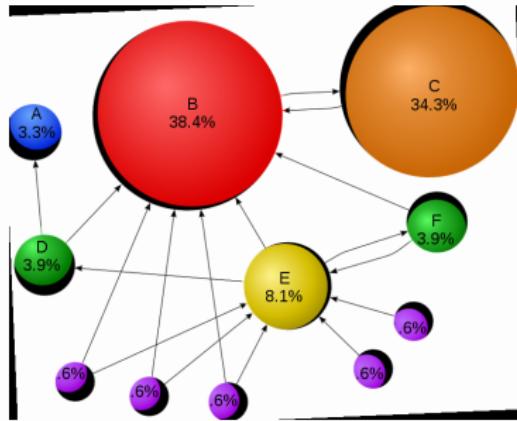
- Web docs contain info beyond their mere “*textual content*”
 - ◊ *state-of-the-art web search* engines, like Google, exploit this
 - ◊ achieve *much more effective* retrieval than could without it
- HTML contains clues that some terms are *more important*
 - e.g. terms in regions marked as title or headings
 - e.g. terms *emphasised by formatting*: bold / bigger / colour
 - ◊ can use clever term weighting schemes, that add weight to such terms
- Link text — commonly provide *description of target* doc
 - ◊ often a better description than doc provides of *itself*
 - e.g. “Hey, here’s a great intro to calculus for beginners – check it out!”
 - ◊ Google treats link text as *part of* target doc
- Link structure of web *more broadly*
 - ◊ if page A *points to* page B, implies B is worth looking at
 - ◊ can be used as a measure of *authority / quality*

Exploiting Link Structure: the PageRank Algorithm

- Key method to exploit link structure of web: **PageRank algorithm**
 - ◊ named after its inventor: **Larry Page** (co-founder of Google)
 - ◊ assigns a score to each page on web: its **PageRank score**
 - can be seen to represent the page's **authority** (or **quality**)
- **PageRank algorithm** — key idea:
 - ◊ link from page A to page B confers **authority** on B
 - ◊ **how much** authority is conferred depends on:
 - the authority (PageRank score) of A, and its number of **out-going links**
i.e. A's authority is **shared out** amongst its out-going links
 - ◊ note that this measure is **recursively defined**
 - i.e. score of any page depends on score of every other page
- **PageRank** scores have an alternative interpretation:
 - ◊ probability that a **random surfer** will visit that page
 - i.e. one who starts at a random page, clicks randomly-chosen links forward, then (getting bored) jumps to a new random page, and so on ...

Exploiting Link Structure: the PageRank algorithm (ctd)

- Graphical intuition:



- During retrieval, rank score of doc d is a *weighted combination* of:
 - its PageRank score: a measure of its authority
 - its IR-Score: how well d matches the query q , based on
 - Vector Space model, TF.IDF, *up-weighting* of important terms, etc

COM6115: Text Processing

*Information Retrieval:
Evaluating IR systems*

Mark Hepple

Department of Computer Science
University of Sheffield

Overview

- Definition of the information retrieval problem
- Approaches to document indexing
 - ◊ manual approaches
 - ◊ automatic approaches
- Automated retrieval models
 - ◊ boolean model
 - ◊ ranked retrieval methods (e.g. vector space model)
- Term manipulation:
 - ◊ stemming, stopwords, term weighting
- Web Search Ranking
- Evaluation

Evaluation of IR systems – Why?

- There are various retrieval models/algorithms/IR systems
 - ◊ How determine which is the best?
- What is the best component/technique for:
 - ◊ Ranking? (cosine, dot-product, ...)
 - ◊ Term selection? (stopword removal, stemming, ...)
 - ◊ Term weighting? (binary, TF, TF.IDF, ...)
- How far down the ranked list will a user need to look to find some/all relevant items?

Evaluation – Relevance

- Evaluation of effectiveness in relation to the **relevance** of the documents retrieved
- Relevance is judged in a **binary** way, even if it is in fact a continuous judgement
 - ◊ Impossible when the task is to **rank thousands or millions of options**: too subjective, too difficult
- Other factors could also be evaluated:
 - ◊ User effort/ease of use
 - ◊ Response time
 - ◊ Form of presentation

Evaluation – Relevance (Benchmarking)

- In IR research/development scenarios, one cannot afford **humans** looking at results of every system/variant of system
- Instead, performance measured/compared using a pre-created **benchmarking** corpus, a.k.a. **gold-standard dataset**, which provides:
 - ◊ a standard set of documents, and queries
 - ◊ a list of documents judged relevant for each query, by human subjects
 - ◊ relevance scores, usually treated as binary
- Example: TREC IR evaluation corpora (<http://trec.nist.gov/>)
 - ◊ TREC has run annually since 1991

Evaluation of IR systems – Metrics

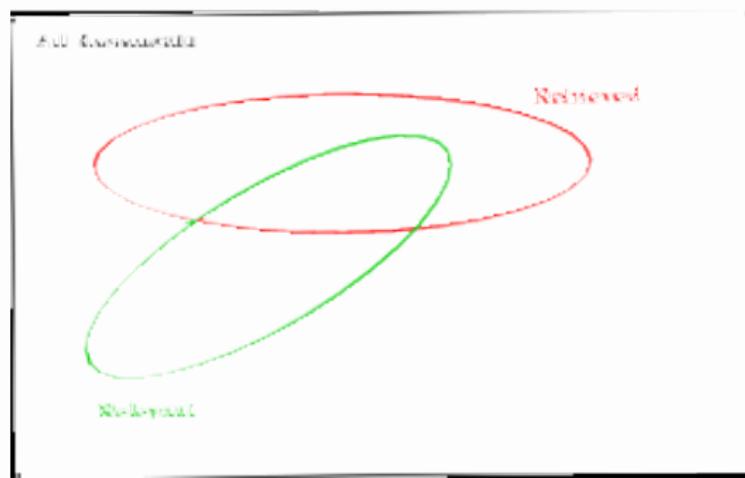
- AIM:
 1. get as much good stuff as possible
 2. get as little junk as possible
- The two aspects of this aim are addressed by two separate measures — **recall** and **precision**

	Relevant	Non-relevant	Total
Retrieved	A	B	A+B
Not retrieved	C	D	C+D
Total	A+C	B+D	A+B+C+D

- **Recall:** $\frac{A}{A+C}$ = proportion of relevant documents returned
- **Precision:** $\frac{A}{A+B}$ = proportion of retrieved documents that are relevant
 - ◊ Both measures have **range**: [0...1]

Retrieved vs. Relevant Documents

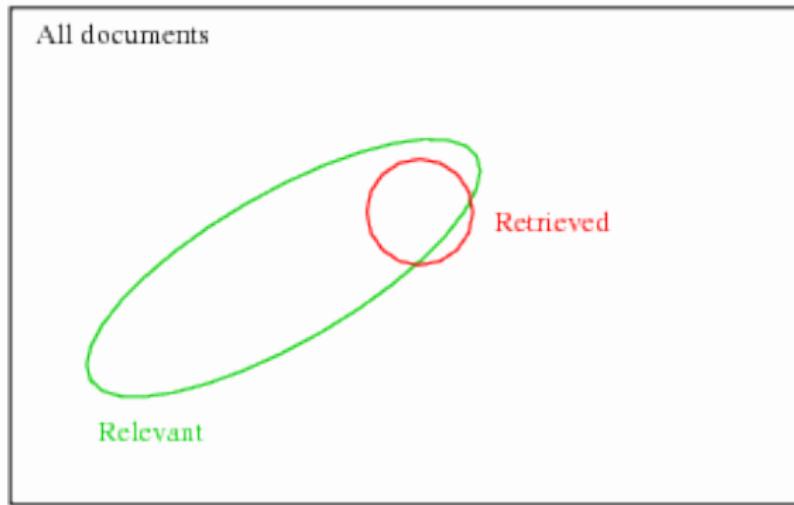
- Precision and Recall address the relation between the *retrieved* and *relevant* sets of documents



- Various situations that arise can be pictorially represented in these terms

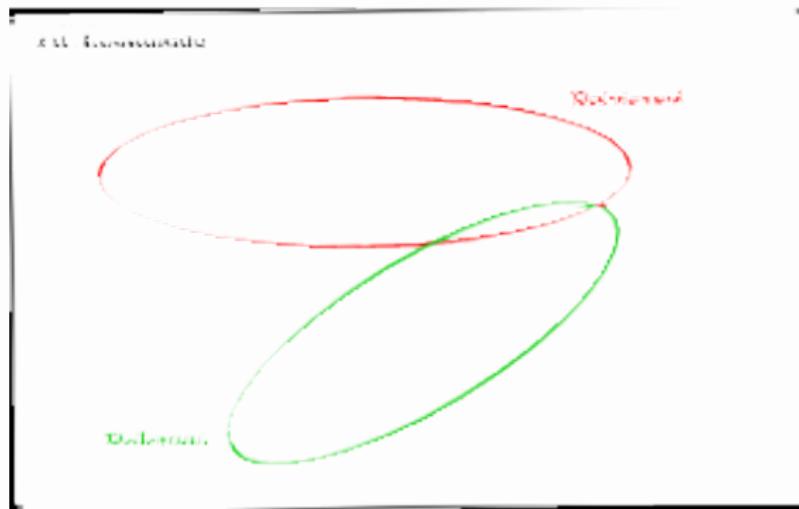
Retrieved vs. Relevant Documents (contd)

- High precision, low recall:



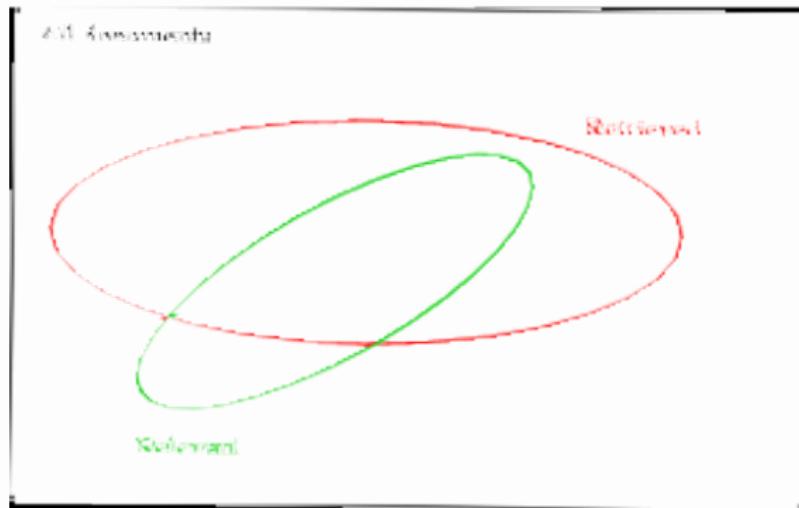
Retrieved vs. Relevant Documents (contd)

- Low precision, low recall:



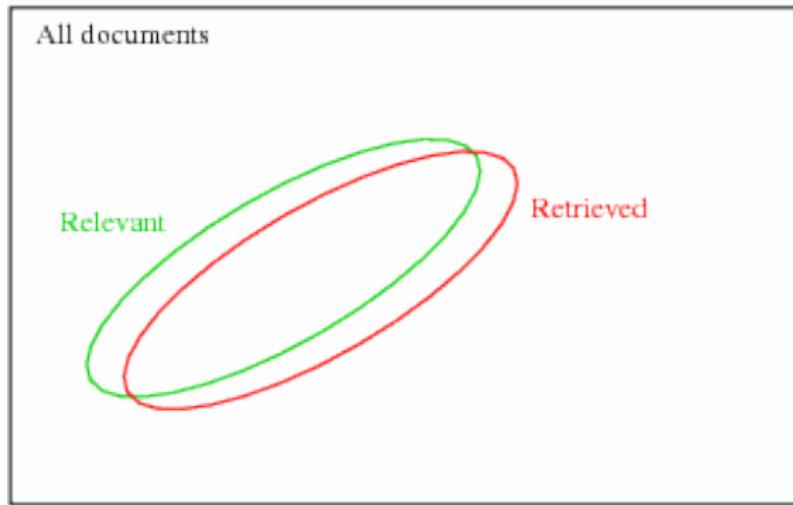
Retrieved vs. Relevant Documents (contd)

- Low precision, high recall:



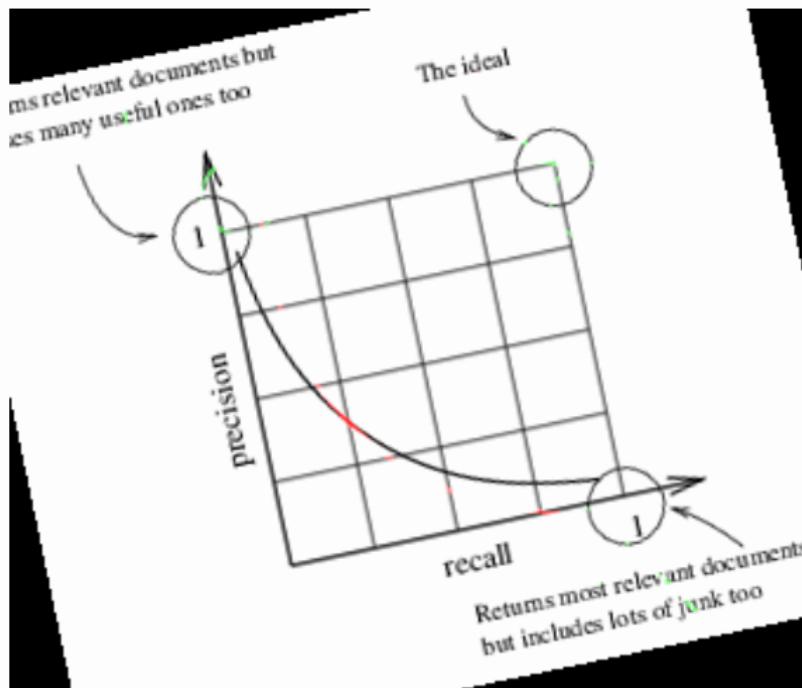
Retrieved vs. Relevant Documents (contd)

- High precision, high recall:

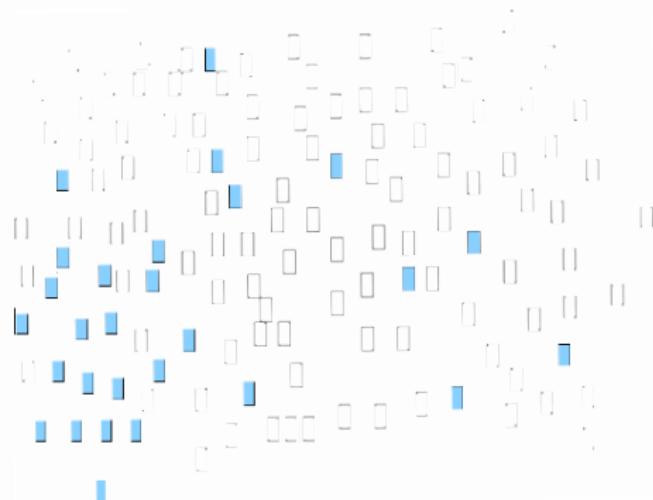


Trade-off Between Recall and Precision

- There is always a trade-off between precision and recall
 - ◊ For IR: as more results are considered down the list, precision generally drops, while recall generally increases



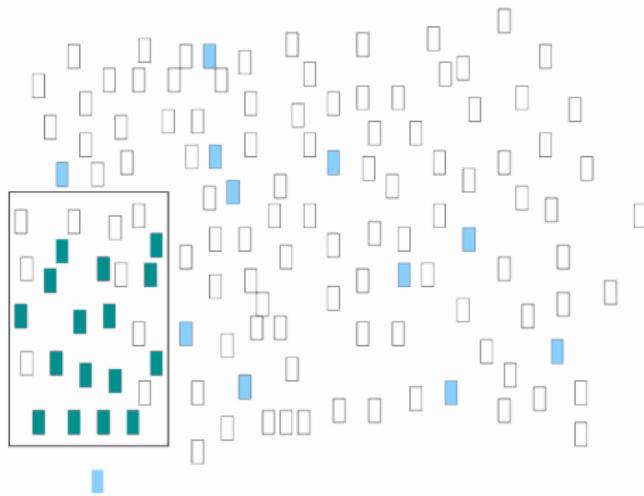
Recall and Precision: Example



	Rel	Non-rel
Ret	A	B
Not ret	C	D

- All documents: $A+B+C+D = 130$
- Relevant documents for query: $A+C = 28$

Recall and Precision: System 1



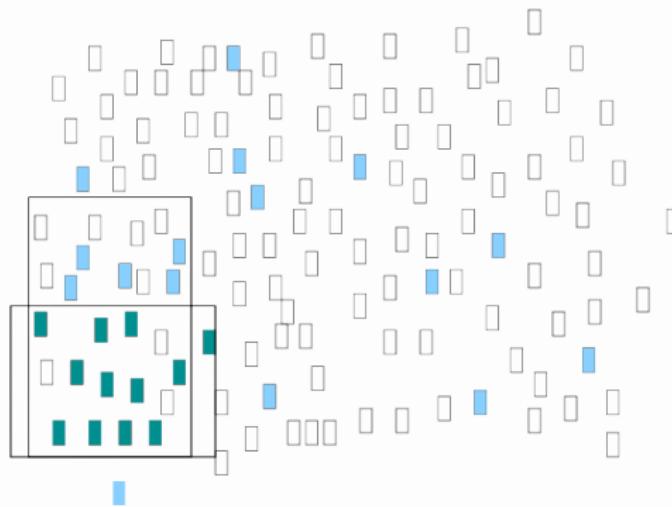
	Rel	Non-rel
Ret	A	B
Not ret	C	D

$$R_1 = \frac{A_1}{A_1+C_1} = \frac{16}{28} = .57$$

$$P_1 = \frac{A_1}{A_1+B_1} = \frac{16}{25} = .64$$

- System 1 retrieves 25 items: $A_1+B_1 = 25$
- Relevant and retrieved items: $A_1 = 16$
- Relevant documents for query: $A_1+C_1 = 28$

Recall and Precision: System 2



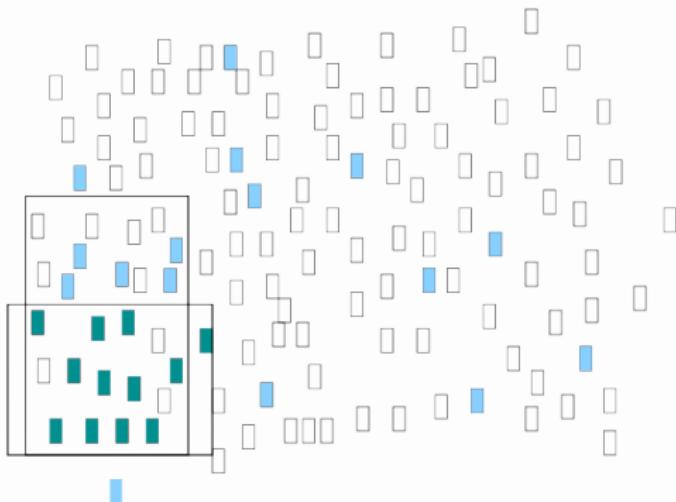
	Rel	Non-rel
Ret	A	B
Not ret	C	D

$$R_2 = \frac{A_2}{A_2+C_2} = \frac{12}{28} = .43$$

$$P_2 = \frac{A_2}{A_2+B_2} = \frac{12}{15} = .8$$

- System 2 retrieves 15 items: $A_2+B_2 = 15$
- Relevant and retrieved items: $A_2 = 12$
- Relevant documents for query: $A_2+C_2 = 28$

Recall and Precision: Which is the better system?



$$R_1 = \frac{A_1}{A_1+C_1} = \frac{16}{28} = .57$$

$$P_1 = \frac{A_1}{A_1+B_1} = \frac{16}{25} = .64$$

$$R_2 = \frac{A_2}{A_2+C_2} = \frac{12}{28} = .43$$

$$P_2 = \frac{A_2}{A_2+B_2} = \frac{12}{15} = .8$$

- Which did better: System 1 or System 2?

F-measure

- F measure (also called F_1):
 - ◊ combines precision and recall into a single figure
 - ◊ gives equal weight to both:
- F is a harmonic mean:
 - ◊ penalises low performance in one value more than *arithmetic* mean:

$$F = \frac{2PR}{P + R}$$

e.g.	values	mean	F
	P=0.5, R=0.5	0.5	0.5
	P=0.1, R=0.9	0.5	0.18

- ◊ Previous example:
- | | R | P | F |
|----------|-----|-----|-------|
| System 1 | .57 | .64 | 0.603 |
| System 2 | .43 | .8 | 0.559 |
- Related measure F_β :
 - ◊ allows user to determine relative importance of P vs. R , by *varying* β
 - ◊ F_1 is a *special case* of F_β (where $\beta = 1$)

Precision at a cutoff

- Measures how well a method **ranks relevant documents** before non-relevant documents
- E.g. there are **5 relevant documents = d1,d2,d3,d4,d5** – compute precision at top 5

	System 1	System 2	System 3
rank 5:	d1: ✓	d10: ✗	d6: ✗
	d2: ✓	d9: ✗	d1: ✓
	d3: ✓	d8: ✗	d2: ✓
	d4: ✓	d7: ✗	d10: ✗
	d5: ✓	d6: ✗	d9: ✗
rank 10:	d6: ✗	d1: ✓	d3: ✓
	d7: ✗	d2: ✓	d5: ✓
	d8: ✗	d3: ✓	d4: ✓
	d9: ✗	d4: ✓	d7: ✗
	d10: ✗	d5: ✓	d8: ✗
<i>precision at rank 5:</i>		1.0	0.0
<i>precision at rank 10:</i>		0.5	0.5

Precision at a cutoff (ctd)

- Note precision at top 5 for System 1: inner order of relevant documents doesn't matter as long as they are all relevant

	System 1	System 2	System 3
rank 5:	d5: ✓	d10: ✗	d6: ✗
	d4: ✓	d9: ✗	d1: ✓
	d3: ✓	d8: ✗	d2: ✓
	d1: ✓	d7: ✗	d10: ✗
	d2: ✓	d6: ✗	d9: ✗
	d6: ✗	d1: ✓	d3: ✓
	d7: ✗	d2: ✓	d5: ✓
	d8: ✗	d3: ✓	d4: ✓
	d9: ✗	d4: ✓	d7: ✗
	d10: ✗	d5: ✓	d8: ✗
<i>precision at rank 5:</i>			
	1.0	0.0	0.4
<i>precision at rank 10:</i>			
	0.5	0.5	0.5

Average Precision

- Aggregates many precision numbers into one evaluation figure
- Precision computed for each point a relevant document is found, and figures averaged

	System 1	System 2	System 3
d1:	✓ (1/1)	✗	✗
d2:	✓ (2/2)	✗	✓ (1/2)
d3:	✓ (3/3)	✗	✓ (2/3)
d4:	✓ (4/4)	✗	✗
d5:	✓ (5/5)	✗	✗
d6:	✗	✓ (1/6)	✓ (3/6)
d7:	✗	✓ (2/7)	✓ (4/7)
d8:	✗	✓ (3/8)	✓ (5/8)
d9:	✗	✓ (4/9)	✗
d10:	✗	✓ (5/10)	✗

<i>precision at rank 5:</i>	1.0	0.0	0.4
<i>precision at rank 10:</i>	0.5	0.5	0.5
<i>avg. prec:</i>	1.0	0.354	0.573

COM6115: Text Processing

Sentiment Analysis

Chenghua Lin

Department of Computer Science
University of Sheffield

Agenda

The course covers four topics in text processing:

- **Sentiment analysis**
- **Natural language generation**
- **Information extraction**
- Information retrieval

Learning Outcomes

By the end of the SA sessions, you will be able to:

- Explain the relevance of the topic
- Differentiate between objective and subjective texts
- List the main elements in a sentiment analysis system
- Provide a critical summary of the main approaches to the problem
- Explain how sentiment analysis systems are evaluated.

Overview

- **Definition of the problem of sentiment analysis**
- **Approaches to sentiment analysis**
- Evaluation of sentiment analysis approaches

Based on survey and slides by Bing Liu (University of Illinois at Chicago), 2012.

General goal

Certain texts, particularly on the Web, have **emotions** or **sentiments** or **opinions**, e.g.:

- Blogs and microblogs (Twitter, etc.)
- Social networks (Facebook, myspace, etc.)
- User comments, like on Youtube, or on products, like on Amazon
- Review websites, like Rotten Tomatoes, yelp
- Community websites, like Symantec Forums

Size of blogosphere: over 112 million blogs, 75,000 created each day,
1.2 million posts/day¹

Social networks like Twitter...

¹<http://technorati.com/state-of-the-blogosphere/>

General goal

Extract **opinions**, **sentiments** and **emotions** expressed by humans in texts and use this information for business, intelligence, etc. purposes. Can't be done manually: huge volumes of opinionated text (esp. **Big Data** on the Web). Examples of applications:

- **Product review mining**: Which features of the iPhone 11 customers like and which do they dislike?
- **Review classification**: Is a movie review positive or negative?
- **Tracking sentiments toward topics over time**: Is anger about the government policies growing or cooling down?
- **Prediction (election outcomes, market trends)**: Will the Tories win the next election?

Here: opinion = sentiment = emotion

Here: sentiment analysis = opinion mining

Although sentiment doesn't always express opinion: "I am sad today".

Importance of opinions

- Whenever we need to **make a decision**, we may want to hear others' opinions
- In the past: surveys, focus groups, consultants, opinions from friends and family
- Nowadays: Word-of-mouth on the Web
 - ◊ User-generated media: one can express opinions on anything in reviews, forums, discussion groups, blogs ...
 - ◊ Opinions of global scale: no longer limited to one's circle of friends (individuals), small scale surveys, focus groups, etc. (businesses)

Importance of opinions

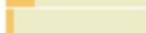
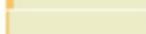
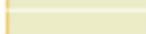
- **Individuals:** interested in other's opinions when
 - ◊ purchasing a product or using a service,
 - ◊ finding opinions on political or other topics.
- **Businesses and organizations:**
 - ◊ product and service benchmarking.
 - ◊ market intelligence.
 - ◊ cost reduction: business spends a huge amount of money to find consumer sentiments and opinions with consultants, surveys and focused groups, etc.
- **Ad placement:** Placing ads in user-generated content
 - ◊ Place an ad when one praises a product.

Product Review Insights

Customer Reviews

Amazon Kindle Keyboard Leather Cover, Black

855 Reviews

5 star:	 (594)
4 star:	 (167)
3 star:	 (47)
2 star:	 (22)
1 star:	 (25)

Average Customer Review

 (855 customer reviews)

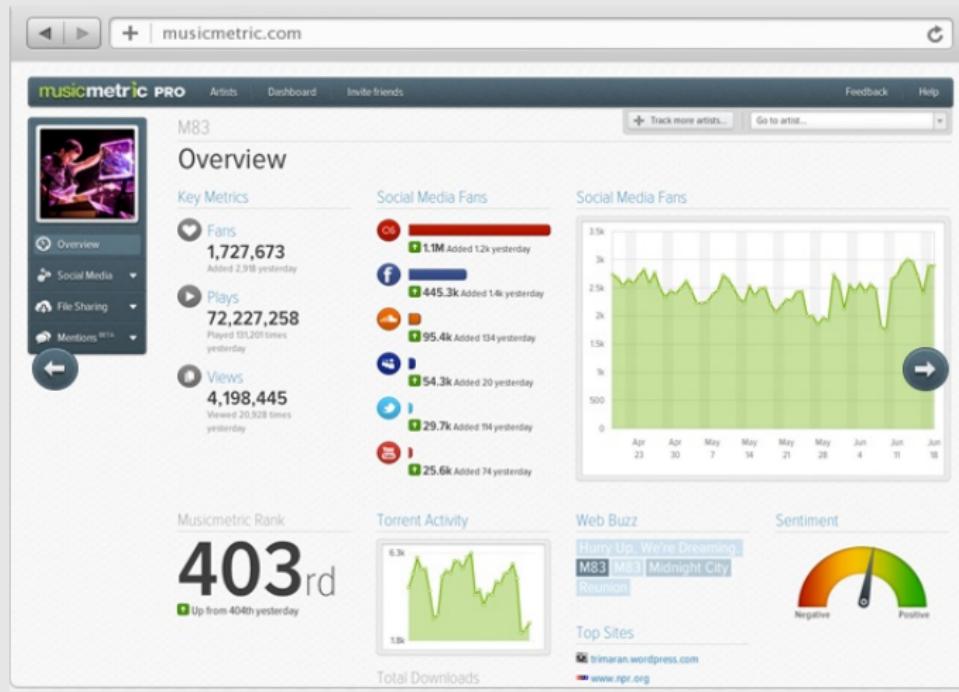
Share your thoughts with other customers

[Create your own review](#)

- What are people's opinions about this product?
- What are the pros and cons?

Brand and Consumer Perception

- **Music artists analytics:** provide aggregated sentiment statistics for artists, songs or albums over all reviews collected online.



Motivations

"I bought the new iPhone a few days ago. It was such a nice phone. The touch screen was really cool. The voice quality was clear too. Although the battery life is not long, that is ok for me. However, my mother was mad with me as I did not tell her before I bought the phone. She also thought the phone was too expensive, and wanted me to return it to the shop."



What do we see in this text? Positive or negative opinions?

Motivations

"I bought the new iPhone a few days ago. It was such a nice phone. The touch screen was really cool. The voice quality was clear too. Although the battery life is not long, that is ok for me. However, my mother was mad with me as I did not tell her before I bought the phone. She also thought the phone was too expensive, and wanted me to return it to the shop."

Objective sentence

Motivations

"I bought the new iPhone a few days ago. It was such a nice phone. The touch screen was really cool. The voice quality was clear too. Although the battery life is not long, that is ok for me. However, my mother was mad with me as I did not tell her before I bought the phone. She also thought the phone was too expensive, and wanted me to return it to the shop."

Positive and negative opinions about what?

Motivations

"I bought the new iPhone a few days ago. It was such a nice phone. The touch screen was really cool. The voice quality was clear too. Although the battery life is not long, that is ok for me. However, my mother was mad with me as I did not tell her before I bought the phone. She also thought the phone was too expensive, and wanted me to return it to the shop."

Targets of opinions

Motivations

"I bought the new iPhone a few days ago. It was such a nice phone (I). The touch screen was really cool (I). The voice quality was clear too (I). Although the battery life is not long, that is ok for me (I). However, my mother was mad with me as I did not tell her before I bought the phone (**mother**). She also thought the phone was too expensive, and wanted me to return it to the shop. (**mother**)"

Holders of opinions

Definitions

Facts versus Opinions

- Current text processing methods (e.g., web search, information extraction) work with **factual information**.
- Current search ranking strategy not appropriate for opinion retrieval.
- Sentiment analysis focuses on **subjective statements** - opinions, sentiments, emotions: hard to express with a few keywords. E.g.
What do people think of Motorola Cell phones?

Excellent phone, excellent service

Just double check with customer service to ensure the number provided by amazon is for the city you wanted

I'd always eyed the nokia phones and had heard decent things about t-mobile, so i gave it a whirl

It costed 500 dollars, not worth the price really

It costed 500 dollars!!!

Subjectivity analysis

Subjectivity classification is often the first step for sentiment analysis:
subjective versus objective texts, e.g.:

- **Objective:** *I bought an iPhone a few days ago.*
- **Subjective:** *It is such a nice phone.*

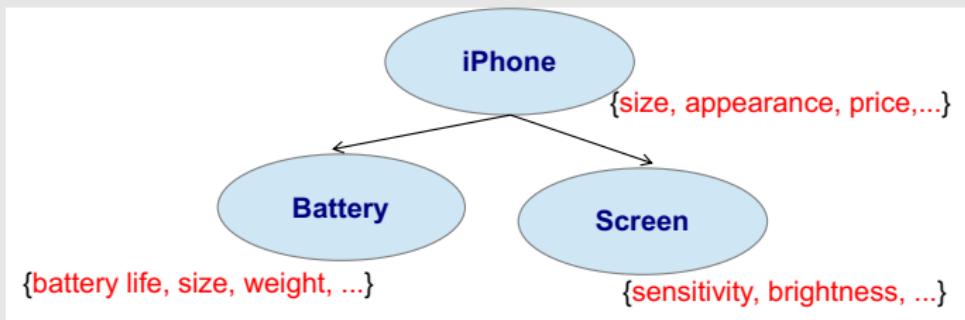
However:

- Subjective sentences do not always express positive or negative opinions, e.g.: *I think he came yesterday.*
- Objective sentences can express opinion indirectly, e.g.: *My phone broke in the second day.*

Sentiment Analysis

Target objects

- Product, person, event, organization, or topic: o . It is represented as
 - ◊ A hierarchy of **components**, **sub-components**, etc.
 - ◊ Each node represents a component and has a set of **attributes**.



An opinion can be expressed on any component or attribute of the component – call them both “**features**” of the object.

Bing Liu's model for Sentiment Analysis

An **opinion** is a quintuple $(o_j, f_{jk}, so_{ijkl}, h_i, t_l)$, where:

- o_j is a target object.
- f_{jk} is a feature of the object o_j .
- so_{ijkl} is the sentiment value of the opinion of the
- opinion holder h_i (usually the author of the post)
- on feature f_{jk} of object o_j at time t_l .

so_{ijkl} is positive, negative, neutral, or a more granular rating, such as 1-5 stars as in movie reviews.

Sentiment Analysis

For example:

"I bought the new iPhone a few days ago. It was such a nice phone. The touch screen was really cool. The voice quality was clear too. Although the battery life is not long, that is ok for me. However, my mother was mad with me as I did not tell her before I bought the phone. She also thought the phone was too expensive, and wanted me to return it to the shop."

- o_j : iPhone
- f_{jk} : phone, screen, voice quality, battery life, price
- so_{ijkl} : positive, positive, positive, negative, negative
- opinion holder h_i : I, I, I, I, mother
- time t_l : post's date

Sentiment Analysis

The task of **opinion mining** is: given an opinionated document:

- Discover all quintuples $(o_j, f_{jk}, so_{ijkl}, h_i, t_l)$, or
- Discover some of these components

With that, one can **structure the unstructured**:

- Traditional data and visualisation tools can be used to slice, dice and visualise the results.
- Qualitative and quantitative analysis can be done.

Sentiment Analysis

Granularity level:

- **Document level**: classify a document (e.g., a movie review) based on the overall sentiment expressed by opinion holder into, e.g.: positive, or negative (and neutral).
 - ◊ Assumption: Each document focuses on a single object and contains opinions from a single opinion holder: $(o_j, f_{jk}, so_{ijkl}, h_i, t_l)$, where $o_j = f_{jk}$
- **Sentence level**: idem, but for (subjective) sentences, so these need to be identified first.
- **Feature level**: documents and sentences may contain **mixed** opinions and analysis at this level does not identify specifically **what** people like/dislike.
 - ◊ An overall positive/negative opinion on an object does not mean that the opinion holder likes/dislikes everything about it. More informative to find opinions on components and/or attributes – allows all sorts of analyses.

Granularity level - feature level (ctd) - Steps:

- Identify and extract object features that have been commented on by an opinion holder (e.g., a reviewer).
- Determine whether the opinions on the features are positive, negative or neutral.
- Group synonym features, e.g. *screen* and *touch screen*.
- Optional: produce a feature-based opinion summary of multiple reviews.

Sentiment Analysis

Granularity level - feature level (ctd):

"I bought the new iPhone a few days ago. ..."

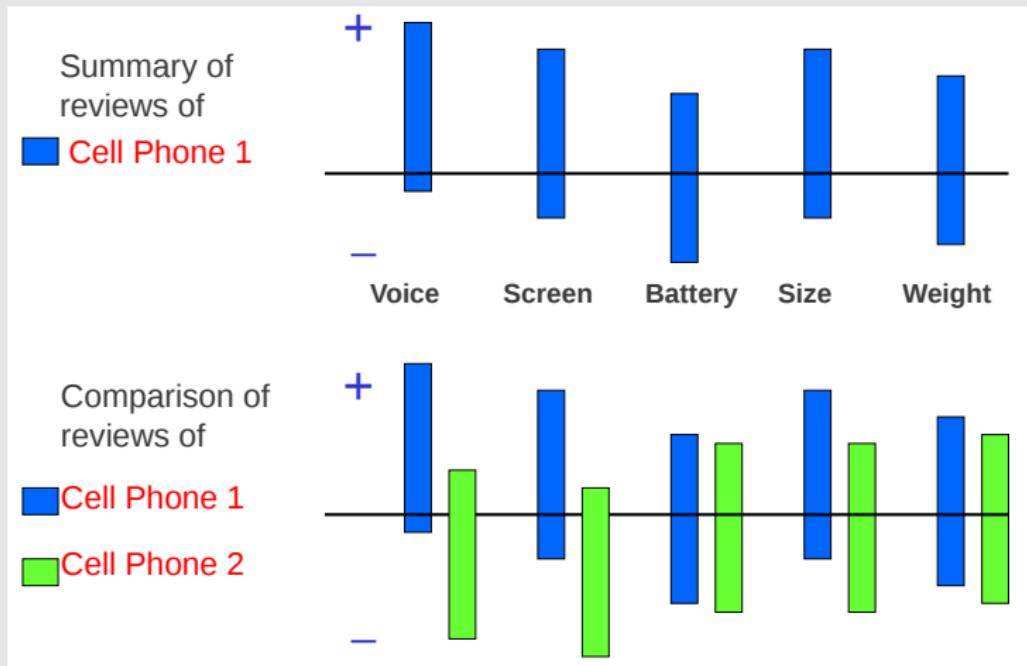


Feature Based Summary:

- Feature1: **touch screen**
 - Positive:
 - ◊ The touch screen was really cool.
 - ◊ The touch screen was so easy to use and can do amazing things.
 - ◊ ...
 - Negative:
 - ◊ The screen is easily scratched.
 - ◊ I have a lot of difficulty in removing finger marks from the touch screen.
- Feature2: **battery life**
 - ...

Sentiment Analysis

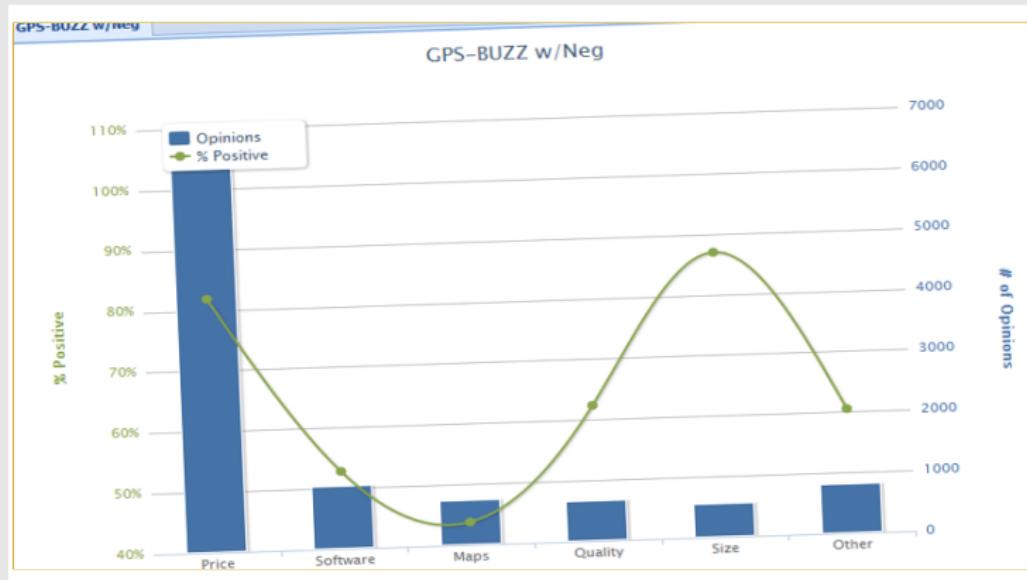
Granularity level - feature level (ctd): Visual Comparison



(Bing Liu)

Sentiment Analysis

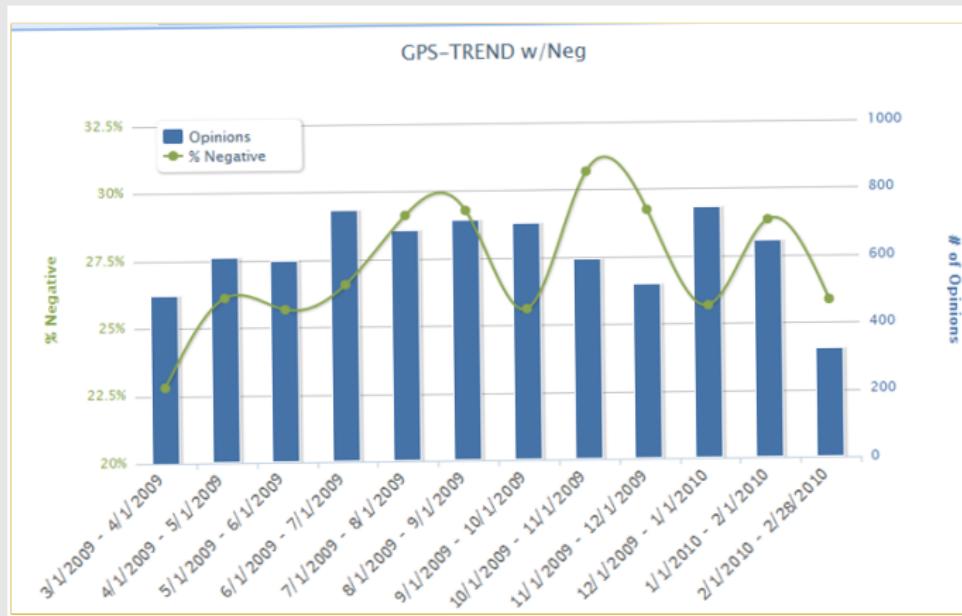
Granularity level - feature level (ctd): Frequency of opinions for a feature



(Bing Liu)

Sentiment Analysis

Granularity level - feature level (ctd): Aggregate opinions over time (trends)



(Bing Liu)

Challenges for Sentiment Analysis

This past Saturday, I bought a Nokia phone and my girlfriend bought a Motorola phone with Bluetooth. We called each other when we got home. The voice on my phone was not so clear, worse than my previous phone. The battery life was long. My girlfriend was quite happy with her phone. I wanted a phone with good sound quality. So my purchase was a real disappointment. I returned the phone yesterday.

Challenges for Sentiment Analysis

One has to solve a number of language processing problems:

$(o_j, f_{jk}, so_{ijkl}, h_i, t_l)$

- o_j : a target object: Named Entity Recognition
- f_{jk} : a feature of o_j : Information Extraction
- so_{ijkl} : a sentiment about f_{jk} : Sentiment determination
- h_i : an opinion holder: Information (or metadata) Extraction
- t_l : a time: Information (or metadata) Extraction

In addition:

- Co-reference resolution
- Relation extraction
- Synonym match (“voice” = “sound quality”)

None of them is a solved problem!

Main components

Identifying target objects

- Named Entity Recognition: well-known tools based on *gazetteers* and simple context rules. E.g.: Paris, BMW and Ford.
 - ◊ Need good gazetteers: Web is dynamic, new products appearing everyday.
 - ◊ Standard NE recognisers will not work for objects like names of movies, e.g., [White on Rice](#)
- Bootstrap from seed gazetteers: e.g. if know that iPhone 4 is an object, can find out that iPhone 5 is also an object.

Main components

Co-reference (and synonym) resolution

- Important to resolve objects and features.
- E.g.: “I bought a **Canon d500** **camera** yesterday. **It** looked beautiful. I took a few **photos** last night. **They** were amazing”. I am happy with the **device**.

→

- E.g.: “I bought a **Canon d500** **camera** yesterday. **The Canon d500 camera** looked beautiful. I took a few **photos** last night. **The photos** were amazing”. I am happy with the **camera**.

References

Bing Liu and Lei Zhang (2012). A survey on opinion mining and sentiment analysis. Kluwer Academic Publishers:

http://www.cs.uic.edu/~lzhang3/paper/opinion_survey.pdf

COM6115: Text Processing

Sentiment Analysis: Approaches

Chenghua Lin

Department of Computer Science
University of Sheffield

Learning Outcomes

By the end of the SA sessions, you will be able to:

- Explain the relevance of the topic
- Differentiate between objective and subjective texts
- List the main elements in a sentiment analysis system
- Provide a critical summary of the main approaches for the problem
- Explain how sentiment analysis systems are evaluated.

Overview

- Definition of the problem of sentiment analysis
- **Approaches to sentiment analysis**
- Evaluation of sentiment analysis approaches

Based on survey and slides by Bing Liu (University of Illinois at Chicago), 2012.

Two approaches to SA

- Lexicon-based
 - ◊ **Binary**
 - ◊ Gradable
- Corpus-based/Supervised machine learning

A simple approach to SA: lexicon-based

Use a lexicon of opinion/emotion words, like: good, bad, horrible, great, etc.

Rule-based sentiment classifier (sentence/document-level)

- 1 Rule-based **subjectivity classifier**: a sentence/document is **subjective** if it has at least n (say 2) words from the emotion words lexicon; a sentence/document is **objective** otherwise.
- 2 Rule-based **sentiment classifier**: for subjective sentences/documents, count positive and negative words/phrases in the sentence/document. If more negative than positive words/phrases, then **negative**; otherwise, **positive** (if equal, neutral).

Lexicon-based approach to SA

Rule-based sentiment classifier (feature-level)

- Assume features can be identified in previous step by information extraction techniques, e.g., battery, phone, screen.
- For each feature, count positive and negative emotion words/phrases from the lexicon.
- If more negative than positive words/phrases, then negative; otherwise, positive (if equal, neutral).

Lexicon-based approach to SA

Rule-based sentiment classifier (feature-based)

- Simple approach:
 - ◊ **Input:** a pair (f, s) , where f is a product feature and s is a sentence that contains f .
 - ◊ **Output:** whether the emotion on f in s is positive, negative, or neutral.
 - ◊ **Step 1:** work on the sentence s containing f .
 - ◊ **Step 2:** select emotion words in s : w_1, \dots, w_n .
 - ◊ **Step 3:** assign orientations for these emotion words: 1 = positive, -1 = negative, 0 = neutral.
 - ◊ **Step 4:** sum up the orientation and assign the orientation to (f, s) accordingly.
- More advanced approaches split the sentence in parts, e.g., based on BUT words ("but", "except that", ...).

Lexicon-based approach to SA

Caveats

- Certain words have context-independent orientations, e.g. “great”.
- Other emotion words have **context-dependent** orientations, e.g.
 - ◊ **small** power consumption = positive
 - ◊ **small** screen = negative
 - ◊ **consume** valuable resources = negative
 - ◊ **consume** disgusting waste = positive
- One has to deal with negation, e.g.:
 - ◊ **not great** = negative
 - ◊ **not bad** = positive
- One has to deal with intensifiers:
 - ◊ **very good** is more positive than **good**
 - ◊ **extremely boring** = is more negative than **boring** or **very boring**

Can store more fine-grained sentiment information in lexicon and add additional **rules**.

Two approaches to SA

- Lexicon-based
 - ◊ Binary
 - ◊ **Gradable**
- Corpus-based/Supevised machine learning

Lexicon-based approach to SA - **gradable**

Use of **ranges of sentiment** instead of a binary system, to deal with intensifiers like:

- absolutely, utterly, completely, totally, nearly, virtually, essentially, mainly, almost, e.g.: **absolutely awful**

And grading adverbs like:

- Very, little, dreadfully, extremely, fairly, hugely, immensely, intensely, rather, reasonably, slightly, unusually, e.g.: **a little bit cold**

Lexicon-based approach to SA - **gradable**

Rule-based gradable sentiment classifier

- Classifies general valence of a text (document-, sentence- or feature-level) based on the level of emotional content
- Level of emotional content given by:
 - 1 The **lexicon**: word-lists with pre-assigned emotional weights, e.g:
Neg. dimension (C_{neg}): $\{-5, \dots, -1\}$, Pos. dimension (C_{pos}): $\{+1, \dots, +5\}$

bore	-3	careful	3
boring	-3	careless	-2
bother	-1	cares	2
brave	3	caring	3
bright	2	casual	2
brilliant	2	casually	2
broke	-1	certain	2
brutal	-3	challeng	2
burden	-1	champ	2
calm	2	charit	2
care	2	charm	2
cared	2	cheat	-3
carefree	2		

Lexicon-based approach to SA - **gradable**

- Ctd:

2 Additional **general rules** to change the original weights:

Negation rule: E.g.: “I am not good today”.

$\text{Emotion(good)} = +3$; “not” is detected in neighbourhood (of 5 words around); so emotional valence of “good” is decreased by 1 and sign is inverted → $\text{Emotion(good)} = -2$

Capitalization rule: E.g. “I am GOOD today”.

$\text{Emotion(good)} = +3$; Add +1 to positive words → $\text{Emotion(GOOD)} = +4$

Likewise, in “I am AWFUL today”.

$\text{Emotion(awful)} = -4$; Add -1 to negative words → $\text{Emotion(awful)} = -5$

Lexicon-based approach to SA - **gradable**

Intensifier rule:

- Needs a list of intensifiers: “definitely”, “very”, “extremely”, etc.
- Each intensifiers has a weight, e.g. $\text{Weight}(\text{very})=1$;
 $\text{Weight}(\text{extremely})=2$
- The weight is **added to positive terms**
- The weight is **subtracted from negative terms**
- E.g.: “I am feeling very good”.
 $\text{Emotion}(\text{good}) = +3$; emotional valence of “good” increased by 1
→ **Emotion(good) = +4**
- E.g. “This was an extremely boring game”
 $\text{Emotion}(\text{boring}) = -3$; emotional valence of “boring” decreased by -2
→ **Emotion(boring) = -5**

Lexicon-based approach to SA - **gradable**

Diminisher rule:

- Need a list: “somewhat”, “barely”, “rarely”, etc.
- Each intensifiers has a weight
- The weight is **subtracted from positive terms**
- The weight is **added to negative terms**
- E.g.: “I am somewhat good”.

Emotion(good)= +3; emotional valence of “good” decreased by 1
→ Emotion(good) = +2

- E.g. “This was a slightly boring game”
Emotion(boring)=-3; emotional valence of “boring” increased by 1
→ Emotion(boring) = -2

Lexicon-based approach to SA - **gradable**

Exclamation rule: Functions like intensifiers. E.g.: “Great show!!!”.

$\text{Emotion(great)} = +3$; $\text{Weight}(!!) = 2$

→ $\text{Emotion(great)} = 5$

Emoticon rule: Each has its own emotional weight, like an emotion word.

E.g.: $\text{Emotion}(;) = +2$; $\text{Emotion}(;) = -2$. E.g.: “I can't believe this product ;”

→ $\text{Emotion}(;) = -2$

Lexicon-based approach to SA - **gradable**

- Final decision based on ALL emotion words:
 - ◊ If $|C_{pos}| > |C_{neg}|$ then {positive}
 - ◊ If $|C_{pos}| < |C_{neg}|$ then {negative}
 - ◊ If $|C_{pos}| = |C_{neg}|$ then {neutral}
- E.g.: “He is brilliant but boring”:
 $\text{Emotion(brilliant)} = 2$; $\text{Emotion(boring)} = -3$
 $\rightarrow C_{pos} = 2$, $C_{neg} = -3$, so {negative}
- E.g.: “I am not good today”:
 $\text{Emotion(good)} = -2$
 $\rightarrow C_{pos} = 0$, $C_{neg} = -2$, so {negative}
- E.g.: “I am not GOOD today”: ($\text{Emotion(good)}=3 \rightarrow ???$)
- E.g.: “I am so surprised by this product!!! 😲”: ($\text{Emotion}(;)=-2 \rightarrow ???$)

Lexicon-based approach to SA

Advantages:

- Works effectively with different texts: forums, blogs, etc.
- Language independent - as long as an up-to-date lexicon of emotion words is available
- Doesn't require data for training
- Can be extended with additional lexica, e.g. for new emotion words/symbols as they become popular, esp. in social media

Disadvantages:

- Requires a lexicon of emotion words, which should be fairly comprehensive, covering new words, abbreviations (LOL, m8, etc.), misspelled words, etc.

E.g.: In a dataset from MySpace, 95% of comments contained at least one spelling error!

Lexica of emotion words/phrases

For both binary and gradable approaches, how to obtain lexica of emotion words?

Task: Collect relevant words/phrases that can be used to express sentiment. Determine the **emotion** of these subjective word/phrases.

- **Manually:** word lists with pre-assigned emotional weights
- Semi-automatically
 - ◊ **Dictionary-based:** find synonyms/antonyms of seed emotion words in dictionaries like WordNet
 - ◊ **Corpus-based:** find synonyms/antonyms of seed emotion words in corpora

Lexica of emotion words/phrases (ctd)

Mostly **adjectives**

- **Positive**: e.g.: honest, important, mature, large, patient, ...
- **Negative**: harmful, hypocritical, inefficient, insecure

Verbs

- **Positive**: praise, love
- **Negative**: blame, criticize

Nouns

- **Positive**: pleasure, enjoyment
- **Negative**: pain, criticism

Phrases (esp. for collocations, but also alternative to having intensifiers weighted separately)

- **Positive**: high intelligence, low cost
- **Negative**: little variation, many problems

Lexica of emotion words/phrases (ctd)

Semi-automatically created resources, such as:

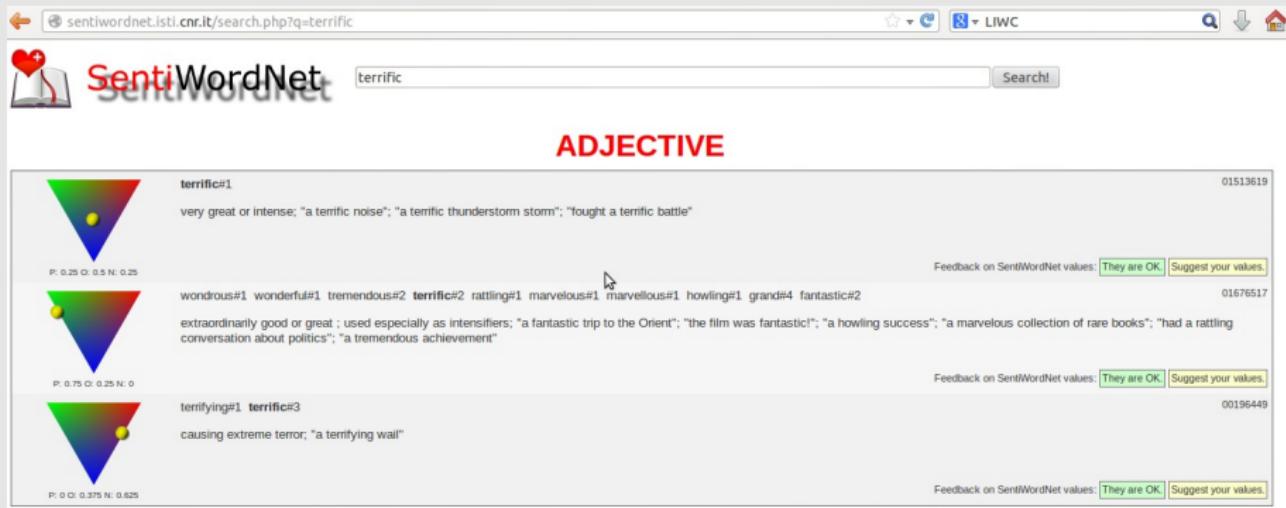
- **SentiWordNet**: Wordnet is a database with words grouped into sets of synonyms (synsets), and organised by semantic relations between them: synonyms, antonyms, hypernyms, etc. SentiWordNet is a version of it with one of three sentiment scores for each synset: **positivity**, **negativity**, **objectivity**.

Manually created resources, such as:

- **Linguistic Inquiry and Word Count (LIWC) lexicon**: made by psychologists with lists of words with various **emotional** and other **dimensions**.
- **General Inquirer**: terms with various types of **positive** or **negative** semantic orientation.

Lexica of emotion words/phrases (ctd)

SentiWordNet



Lexica of emotion words/phrases (ctd)

Linguistic Inquiry and Word Count lexicon

Category	Abbrev	Examples	Words In Category
Psychological Processes			
Social processes	social	Mate, talk, they, child	455
Family	family	Daughter, husband, aunt	64
Friends	friend	Buddy, friend, neighbor	37
Humans	human	Adult, baby, boy	61
Affective processes	affect	Happy, cried, abandon	915
Positive emotion	posemo	Love, nice, sweet	406
Negative emotion	negemo	Hurt, ugly, nasty	499
Anxiety	anx	Worried, fearful, nervous	91
Anger	anger	Hate, kill, annoyed	184
Sadness	sad	Crying, grief, sad	101
Cognitive processes	cogmech	cause, know, ought	730
Insight	insight	think, know, consider	195

Lexica of emotion words/phrases (ctd)

General Inquirer: words classified in many categories, including: positive (1,915) and negative (2,291).

Inquirerbasic.xls (read-only) - LibreOffice Calc

File Edit View Insert Format Tools Data Window Help

A10447:AMJ10447 f3 = TERRORIZE

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Entry	Source	Positiv	Negativ	Pstv	Affil	Ngtv	Hostile	Strong	Power	Weak	Submit	Active	Passive	Pleasur
10440	TERRIBLE	H4Lvd		Negativ											
10441	TERRIFIC	H4Lvd	Positiv				Ngtv								
10442	TERRIFY	H4Lvd		Negativ											
10443	TERRITORIAL	H4Lvd							Strong						Active
10444	TERRITORY	H4Lvd													
10445	TERROR	H4Lvd		Negativ			Ngtv								
10446	TERRORISM	H4Lvd		Negativ				Hostile							
10447	TERRORIZE	H4		Negativ				Hostile						Active	
10448	TEST#1	H4Lvd								Power					
10449	TEST#2	H4Lvd								Power				Active	
10450	TEST#3	H4Lvd													
10451	TESTAMENT	H4Lvd													
10452	TESTIFY	H4Lvd													
10453	TESTIMONY	H4Lvd													
10454	TEXAS	H4Lvd													
10455	TEXT	H4Lvd													
10456	TEXTILE	H4Lvd													
10457	TEXTURE	H4Lvd													

Free dictionary:

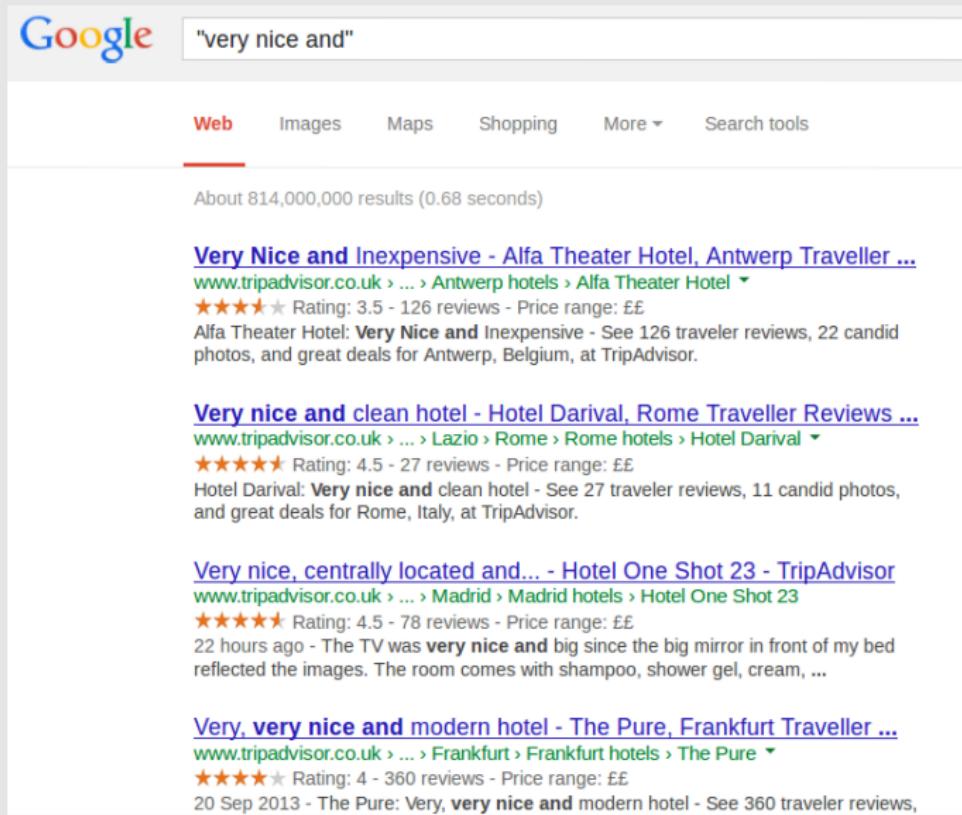
<http://www.wjh.harvard.edu/~inquirer/homecat.htm>

Lexica of emotion words/phrases (ctd)

Semi-automatically created from seed words: start with **seed positive and negative words**:

- Search for synonyms/antonyms in **dictionaries** like WordNet; OR
- Build **patterns** from seed words/phrases to search on large **corpora**, like the Web:
 - ◊ “beautiful and” (+)
 - ◊ “low cost but” (-)
 - ◊ “very nice and ” (+)

Lexica of emotion words/phrases (ctd) - from corpora



Google "very nice and"

Web Images Maps Shopping More ▾ Search tools

About 814,000,000 results (0.68 seconds)

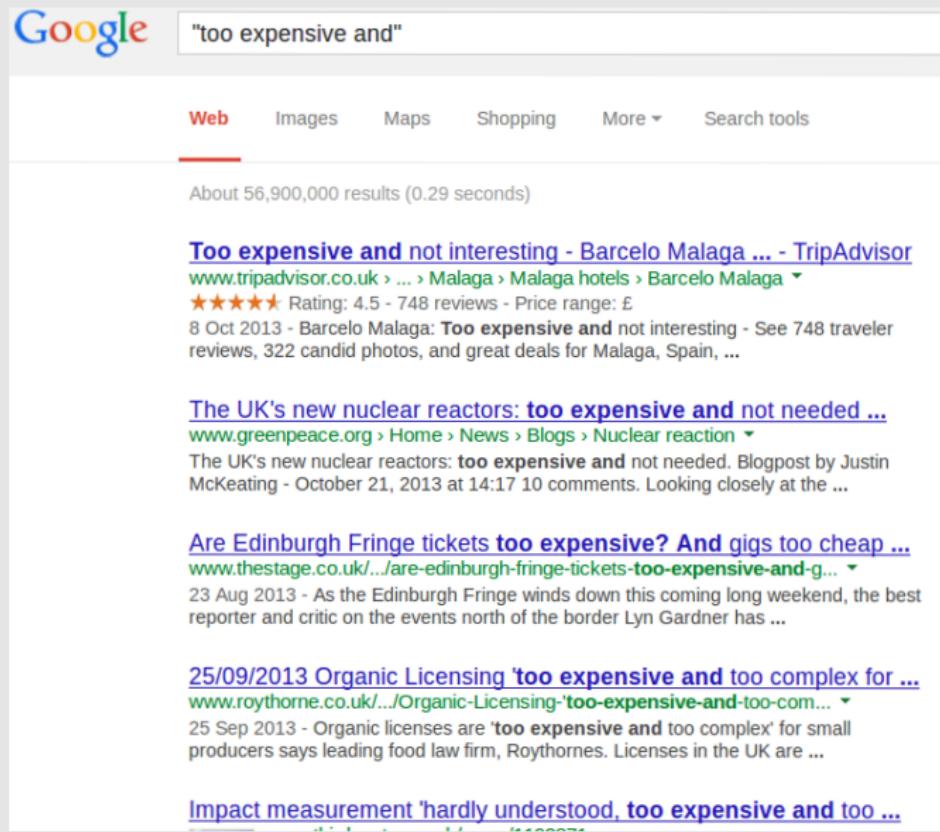
Very Nice and Inexpensive - Alfa Theater Hotel, Antwerp Traveller ...
www.tripadvisor.co.uk > ... > Antwerp hotels > Alfa Theater Hotel ▾
★★★★★ Rating: 3.5 - 126 reviews - Price range: ££
Alfa Theater Hotel: **Very Nice and Inexpensive** - See 126 traveler reviews, 22 candid photos, and great deals for Antwerp, Belgium, at TripAdvisor.

Very nice and clean hotel - Hotel Darival, Rome Traveller Reviews ...
www.tripadvisor.co.uk > ... > Lazio > Rome > Rome hotels > Hotel Darival ▾
★★★★★ Rating: 4.5 - 27 reviews - Price range: ££
Hotel Darival: **Very nice and** clean hotel - See 27 traveler reviews, 11 candid photos, and great deals for Rome, Italy, at TripAdvisor.

Very nice, centrally located and... - Hotel One Shot 23 - TripAdvisor
www.tripadvisor.co.uk > ... > Madrid > Madrid hotels > Hotel One Shot 23
★★★★★ Rating: 4.5 - 78 reviews - Price range: ££
22 hours ago - The TV was **very nice and** big since the big mirror in front of my bed reflected the images. The room comes with shampoo, shower gel, cream, ...

Very, very nice and modern hotel - The Pure, Frankfurt Traveller ...
www.tripadvisor.co.uk > ... > Frankfurt > Frankfurt hotels > The Pure ▾
★★★★★ Rating: 4 - 360 reviews - Price range: ££
20 Sep 2013 - The Pure: Very, **very nice and** modern hotel - See 360 traveler reviews,

Lexica of emotion words/phrases (ctd) - from corpora



A screenshot of a Google search results page. The search query is "too expensive and". The results are as follows:

- Too expensive and not interesting - Barcelo Malaga ... - TripAdvisor**
www.tripadvisor.co.uk › ... › Malaga › Malaga hotels › Barcelo Malaga ▾
★★★★★ Rating: 4.5 - 748 reviews - Price range: £
8 Oct 2013 - Barcelo Malaga: **Too expensive and** not interesting - See 748 traveler reviews, 322 candid photos, and great deals for Malaga, Spain, ...
- The UK's new nuclear reactors: too expensive and not needed ...**
www.greenpeace.org › Home › News › Blogs › Nuclear reaction ▾
The UK's new nuclear reactors: **too expensive and** not needed. Blogpost by Justin McKeating - October 21, 2013 at 14:17 10 comments. Looking closely at the ...
- Are Edinburgh Fringe tickets too expensive? And gigs too cheap ...**
[www.thestage.co.uk/.../are-edinburgh-fringe-tickets-too-expensive-and-g... ▾](http://www.thestage.co.uk/)
23 Aug 2013 - As the Edinburgh Fringe winds down this coming long weekend, the best reporter and critic on the events north of the border Lyn Gardner has ...
- 25/09/2013 Organic Licensing 'too expensive and too complex for ...**
www.roysthorne.co.uk/.../Organic-Licensing-'too-expensive-and-too-com... ▾
25 Sep 2013 - Organic licenses are 'too expensive and too complex' for small producers says leading food law firm, Roythornes. Licenses in the UK are ...
- Impact measurement 'hardly understood, too expensive and too ...**

Two approaches to SA

- Lexicon-based
 - ◊ Binary
 - ◊ Gradable
- **Corpus-based/Supervised machine learning**

A corpus-based approach to SA

Idea: Mostly “supervised learning”: **corpora** of examples **annotated with sentiment** are used with machine learning algorithms to learn a classifier for each sentence/document. Corpora can be built:

- Manually: reliable, can be used as gold-standards
- From crowd-annotated resources, like Amazon Product Reviews (1-5 stars); Rotten Tomatoes, complaints.com, bitterlemons.com

Corpus: a collection of text segments (e.g. webpages, blog posts, reviews, tweets, etc) with humanly-annotated emotional indicators (e.g. positive, negative, etc).

E.g.: “If you are reading this because it is your darling fragrance, please wear it at home exclusively and tape the windows shut.” → {negative}

A corpus-based approach to SA - Corpora

Examples of corpora:

- Subjectivity corpus
 - ◊ 10,000 sentences: subjective/objective
 - ◊ Objective: IMDB plot summaries
 - ◊ Subjective: Rotten Tomatoes website.
- “Movie Review” corpus (Pang, Lee and Vaithyanathan, 2002):
 - ◊ 2,000 movie reviews (equal number of positive/negative)
 - ◊ Source: IMDB
- Many more:

<http://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html>

A corpus-based approach to SA - Features

Mostly words, but also other linguistic traits describing positive/negative examples:

- Words (unigrams)
- n-grams (sequences of n words)
- Emotions from words/phrases extracted from dictionaries
- Part-of-speech (POS) tags
- Syntactic patterns (e.g. sequences of POS tags)
- Language model scores: similarities to positive/negative corpora
- Negations

All automatically extracted from the corpus.

Two steps:

- 1 Subjectivity classifier: first run binary classifier to identify and then eliminate objective segments
- 2 Sentiment classifier with remaining segments: learn how to combine and weight different attributes to make predictions. E.g. Naive Bayes

Pre-processing of corpus similar to IR:

- Remove HTML or other tags
- Remove stopwords
- Perform word stemming/lemmatisation
- etc.

Extra reading

Bing Liu and Lei Zhang (2012). A survey on opinion mining and sentiment analysis. Kluwer Academic Publishers:

http://www.cs.uic.edu/~lzhang3/paper/opinion_survey.pdf

COM6115: Text Processing

Sentiment Analysis: Approaches and Evaluation

Chenghua Lin

Department of Computer Science
University of Sheffield

- Definition of the problem of sentiment analysis
- **Approaches to sentiment analysis**
- **Evaluation of sentiment analysis approaches**

Two approaches to SA

- Lexicon-based
 - ◊ Binary
 - ◊ Gradable
- **Corpus-based (machine learning)**

Learning Outcomes

By the end of the SA sessions, you will be able to:

- Explain the relevance of the topic
- Differentiate between objective and subjective texts
- List the main elements in a sentiment analysis system
- Provide a critical summary of the main approaches for the problem
- Explain how sentiment analysis systems are evaluated.

Text Processing

*All models are wrong
but some are useful*



George E.P. Box

Two Event Models for Naive Bayes

- Today we learn about Naïve Bayes classifier:
 - ◊ How to turn Bayes rule into a classifier
 - ◊ A supervised probabilistic model of the observed data
 - ◊ Can be used to predict the class label of new/unseen data
- Multi-variate Bernoulli Model: a document is a binary vector over the space of words
- Multinomial Model: captures word frequency information in documents

Two Event Models for Naive Bayes

A Comparison of Event Models for Naive Bayes Text Classification

Andrew McCallum^{†‡}
mccallum@justresearch.com
[†]Just Research
4616 Henry Street
Pittsburgh, PA 15213

Kamal Nigam[†]
knigam@cs.cmu.edu
[†]School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

Recent approaches to text classification have used two different first-order probabilistic models for classification, both of which make the *naive Bayes assumption*. Some use a multi-variate Bernoulli model, that is, a Bayesian Network with no dependencies between words and binary word features (*e.g.* Larkey and Croft 1996; Koller and Sahami 1997). Others use a multinomial model, that is, a model with interactions

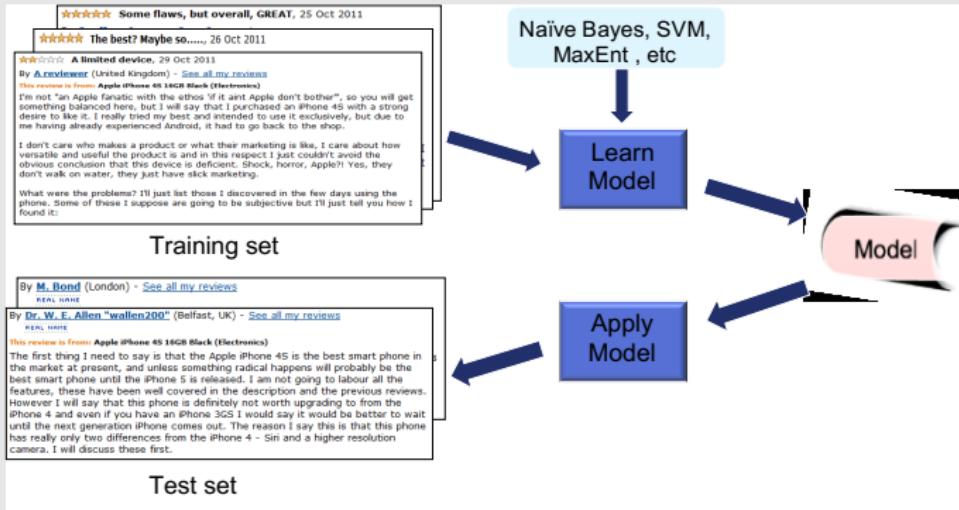
learning, especially when the number of attributes is large.

Document classification is just such a domain with a large number of attributes. The attributes of the examples to be classified are words, and the number of different words can be quite large indeed. While some simple document classification tasks can be accurately performed with vocabulary sizes less than one hundred, many complex tasks on real-world data from

Supervised Classification

- **Supervised learning:** the machine learning task of inferring a function from labeled training data
- Given:
 - ◊ **Target:** a fixed set of **classes**: $Y = y_1, y_2, \dots, y_n$, e.g. {sports, politics, ..., music}
 - ◊ **Training data:** a collection of data objects X with known classes Y , i.e. $(X, Y) = (x_1, y_1), (x_2, y_2) \dots (x_n, y_n)$. E.g {(d1, sports), (d2, sports), (d3, music) ...}.
 - ◊ **Testing data:** a description of an unseen instance, D_{new} e.g. a new document without class label information
- Goal:
 - ◊ Predict the category/class of D_{new} : $y(x) \in Y$, where $y(x)$ is a **classification function**, aka **trained model**, whose domain is X and whose range is Y .

Supervised Classifier



- Rely on syntactic or co-occurrence patterns in large text corpora

The Bayes Rule

$$p(Y | X_1, \dots, X_n) = \frac{P(X_1, \dots, X_n | Y)P(Y)}{P(X_1, \dots, X_n)}$$

Diagram illustrating the components of the Bayes Rule:

- Likelihood**: $P(X_1, \dots, X_n | Y)$
- Prior**: $P(Y)$
- Posterior**: $p(Y | X_1, \dots, X_n)$
- Normalization Constant**: $P(X_1, \dots, X_n)$

- $P(Y)$: Prior belief (probability of hypothesis Y before seeing any data)
- $P(X_1, \dots, X_n | Y)$: Likelihood (probability of the data if the hypothesis Y is true)
- $P(X_1, \dots, X_n)$: Data evidence (marginal probability of data)
- $p(Y | X_1, \dots, X_n)$: Posterior (probability of hypothesis Y after having seen the data)

The Independence Assumption

- Assume A and B are Boolean Random variables. Then
“A and B are independent”

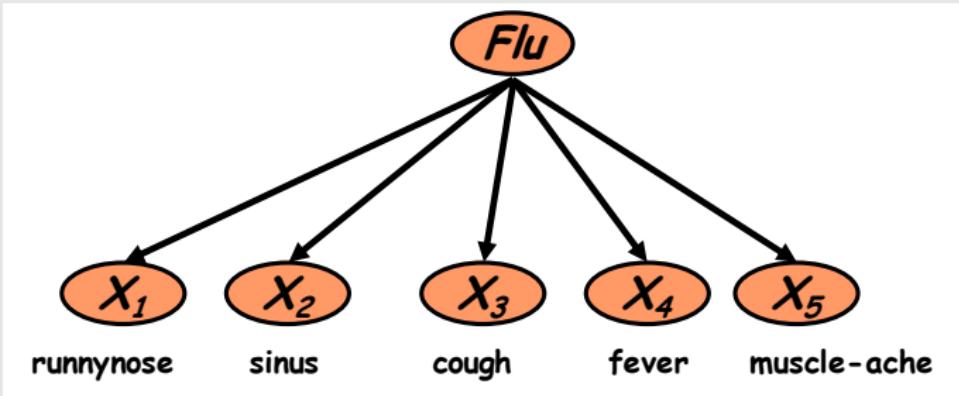
if and only if

$$P(A|B) = P(A)$$

“A and B are independent” is often notated as

$$A \perp B$$

The Independence Assumption



- Features (term presence) are *independent* of each other given the class:

$$P(X_1, \dots, X_5 | Y) = P(X_1 | Y) \bullet P(X_2 | Y) \bullet \dots \bullet P(X_5 | Y)$$

A corpus-based approach to SA - Machine Learning

Naive Bayes classifier: estimate the probability of each class given a text:

- Compute the posterior probability (Bayes rule) of each class c_i for text segment T

$$P(c_i|T) = \frac{P(T|c_i)P(c_i)}{P(T)}$$

- Assumption of independence between features (“naive” assumption)

$$P(T|c_i) = P(t_1, t_2, \dots, t_j|c_i) \approx \prod_{j=1}^n P(t_j|c_i)$$

where T is described by a number of attributes or features t_1, \dots, t_j

i.e. joint probability of the features given the class is approximated by the product of the probabilities of each feature given the class.

A corpus-based approach to SA - Machine Learning

A Naive Bayes classifier (ctd)

- **Likelihood:** product of probabilities of each feature value of segment occurring with class c_i

$$\prod_{j=1}^n P(t_j|c_i)$$

- **Prior:** probability of segment having class c_i

$$P(c_i)$$

- **Evidence:** product of probabilities of features of segment – **constant term for all classes, so can be disregarded:**

$$\prod_{j=1}^n P(t_j)$$

Final decision:

$$\operatorname{argmax}_{c_i} \prod_{j=1}^n P(t_j|c_i)P(c_i) = \operatorname{argmax}_{c_i} P(c_i) \prod_{j=1}^n P(t_j|c_i)$$

A corpus-based approach to SA - Machine Learning

A Naive Bayes classifier - a worked out example

- Corpus of movie reviews: 7 examples for **training**

Doc	Words	Class
1	Great movie, excellent plot, renowned actors	Positive
2	I had not seen a fantastic plot like this in good 5 years. Amazing!!!	Positive
3	Lovely plot, amazing cast, somehow I am in love with the bad guy	Positive
4	Bad movie with great cast, but very poor plot and unimaginative ending	Negative
5	I hate this film, it has nothing original	Negative
6	Great movie, but not...	Negative
7	Very bad movie, I have no words to express how I dislike it	Negative

A corpus-based approach to SA - Machine Learning

A Naive Bayes classifier - a worked out example (ctd)

- **Features:** adjectives (bag-of-words)

Doc	Words	Class
1	Great movie, excellent plot, renowned actors	Positive
2	I had not seen a fantastic plot like this in good 5 years. amazing !!!	Positive
3	Lovely plot, amazing cast, somehow I am in love with the bad guy	Positive
4	Bad movie with great cast, but very poor plot and unimaginative ending	Negative
5	I hate this film, it has nothing original. Really bad	Negative
6	Great movie, but not...	Negative
7	Very bad movie, I have no words to express how I dislike it	Negative

Relative frequency in corpus is the simplest approach to estimating probabilities:

Priors:

$$P(\text{positive}) = \text{count}(\text{positive})/N = 3/7 = 0.43$$

$$P(\text{negative}) = \text{count}(\text{negative})/N = 4/7 = 0.57$$

where N = total training examples

Assume standard pre-processing: tokenisation, lowercasing, punctuation removal (except special punctuation like !!!)

A corpus-based approach to SA - Machine Learning

Likelihoods:

$$P(t_j|c_i) = \frac{\text{count}(t_j, c_i)}{\text{count}(c_i)}$$

Count word t_j in class c_i / total words in that class

$P(\text{amazing} \text{positive})$	= 2/10	$P(\text{amazing} \text{negative})$	= 0/8
$P(\text{bad} \text{positive})$	= 1/10	$P(\text{bad} \text{negative})$	= 3/8
$P(\text{excellent} \text{positive})$	= 1/10	$P(\text{excellent} \text{negative})$	= 0/8
$P(\text{fantastic} \text{positive})$	= 1/10	$P(\text{fantastic} \text{negative})$	= 0/8
$P(\text{good} \text{positive})$	= 1/10	$P(\text{good} \text{negative})$	= 0/8
$P(\text{great} \text{positive})$	= 1/10	$P(\text{great} \text{negative})$	= 2/8
$P(\text{lovely} \text{positive})$	= 1/10	$P(\text{lovely} \text{negative})$	= 0/8
$P(\text{original} \text{positive})$	= 0/10	$P(\text{original} \text{negative})$	= 1/8
$P(\text{poor} \text{positive})$	= 0/10	$P(\text{poor} \text{negative})$	= 1/8
$P(\text{renowned} \text{positive})$	= 1/10	$P(\text{renowned} \text{negative})$	= 0/8
$P(\text{unimaginative} \text{positive})$	= 0/10	$P(\text{unimaginative} \text{negative})$	= 1/8
$P(\text{!!!} \text{positive})$	= 1/10	$P(\text{!!!} \text{negative})$	= 0/8

- Relative frequencies for prior ($P(c_i)$) and likelihood ($P(t_j|c_i)$) make the **model** in a Naive Bayes classifier.
- At decision (test) time, given a new segment to classify, this model is applied to find the most likely class for the segment:

$$\operatorname{argmax}_{c_i} P(c_i) \prod_{j=1}^n P(t_j|c_i)$$

A corpus-based approach to SA - Machine Learning

Given a new segment to classify (**test time**):

Doc	Words	Class
8	This was a fantastic story, good , lovely	???

Final decision

$$\operatorname{argmax}_{c_i} P(c_i) \prod_{j=1}^n P(t_j|c_i)$$

$$P(\text{positive}) * P(\text{fantastic}|\text{positive}) * P(\text{good}|\text{positive}) * P(\text{lovely}|\text{positive})$$

$$3/7 * 1/10 * 1/10 * 1/10 = 0.00043$$

$$P(\text{negative}) * P(\text{fantastic}|\text{negative}) * P(\text{good}|\text{negative}) * P(\text{lovely}|\text{negative})$$

$$4/7 * 0/8 * 0/8 * 0/8 = 0$$

So: **sentiment = positive**

A corpus-based approach to SA - Machine Learning

Given a new segment to classify (**test time**):

Doc	Words	Class
9	Great plot, great cast, great everything	???

Final decision

$$P(\text{positive}) * P(\text{great}|\text{positive}) * P(\text{great}|\text{positive}) * P(\text{great}|\text{positive})$$

$$3/7 * 1/10 * 1/10 * 1/10 = 0.00043$$

$$P(\text{negative}) * P(\text{great}|\text{negative}) * P(\text{great}|\text{negative}) * P(\text{great}|\text{negative})$$

$$4/7 * 2/8 * 2/8 * 2/8 = 0.00893$$

So: **sentiment = negative**

A corpus-based approach to SA - Machine Learning

What if the new segment to classify (**test time**) is:

Doc	Words	Class
10	Lovely plot, excellent cast, amazing everything	???

Final decision

$$P(\text{positive}) * P(\text{lovely}|\text{positive}) * P(\text{excellent}|\text{positive}) * P(\text{amazing}|\text{positive})$$

$$3/7 * 1/10 * 1/10 * 1/10 = 0.00043$$

$$P(\text{negative}) * P(\text{lovely}|\text{negative}) * P(\text{excellent}|\text{negative}) * P(\text{amazing}|\text{negative})$$

$$4/7 * 0/8 * 0/8 * 0/8 = 0$$

So: *sentiment = positive*

A corpus-based approach to SA - Machine Learning

But if the new segment to classify (**test time**) is:

Doc	Words	Class
11	Boring movie, annoying plot, unimaginative ending	???

Final decision

$$P(\text{positive}) * P(\text{boring}|\text{positive}) * P(\text{annoying}|\text{positive}) * P(\text{unimaginative}|\text{positive})$$

$$3/7 * 0/10 * 0/10 * 0/10 = 0$$

$$P(\text{negative}) * P(\text{boring}|\text{negative}) * P(\text{annoying}|\text{negative}) * P(\text{unimaginative}|\text{negative})$$

$$4/7 * 0/8 * 0/8 * 1/8 = 0$$

So: *sentiment* = ???

A corpus-based approach to SA - Machine Learning

Add smoothing to feature counts (add 1 to every count). **Likelihoods** =

$$P(t_j|c_i) = \frac{\text{count}(t_j, c_i) + 1}{\text{count}(c_i) + |V|}$$

where $|V|$ is the number of distinct attributes in training (all classes) = 12

Doc	Words	Class
12	Boring movie, annoying plot, unimaginative ending	???

Final decision

$$P(\text{positive}) * P(\text{boring}|\text{positive}) * P(\text{annoying}|\text{positive}) * P(\text{unimaginative}|\text{positive})$$

$$\frac{3/7 * ((0+1)/(10+12)) * ((0+1)/(10+12)) * ((0+1)/(10+12))}{1} = 0.000040$$

$$P(\text{negative}) * P(\text{boring}|\text{negative}) * P(\text{annoying}|\text{negative}) * P(\text{unimaginative}|\text{negative})$$

$$\frac{4/7 * ((0+1)/(8+12)) * ((0+1)/(8+12)) * ((1+1)/(8+12))}{1} = 0.000143$$

So: *sentiment = negative*

A corpus-based approach to SA - Machine Learning

Given a trained classifier that classifies arbitrary segments of text we can use it to:

- Classify **entire documents**, e.g an entire review.
- Classify **sentences** in a document (perhaps just those identified as subjective) and then compute a classification of the document by aggregating the sentiments of individual sentences, according to some function.
- Classify **sentences or phrases identified as discussing an aspect/feature** of a target object (e.g. a sentence discussing battery life of a phone) and interpret the sentiment as the sentiment of opinion holder towards the specific aspect under discussion

Questions:

- Is this a good solution? Is it robust?
- What is the role of the **prior**?
- How can we improve this solution?
 - ◊ Other **features**? Are we missing out critical information?
 - ◊ Other **algorithms**?
- What about **non-binary classification** (e.g. 5-grades of sentiment)?

Questions:

- Is this a good solution? Is it robust?
 - It's simple and will work well if data is not sparse
- What is the role of the **prior**?
 - Prior is very important esp. on biased cases
- How can we improve this solution?
 - ◊ Other **features**? Are we missing out critical information?
 - Using all words (in Naive Bayes) works well in some tasks
 - Finding subsets of words may help in other tasks
 - Using only adjectives can be limiting. Verbs like **hate**, **dislike**; nouns like **love**; words for inversion like **not**; intensifiers like **very**
 - Pre-built polarity lexicons can be helpful
 - Negation is important
 - ◊ Other **algorithms**?
 - MaxEnt & SVM tend to do better than Naive Bayes
- What about **non-binary classification** (e.g. 5-grades of sentiment)?
 - 5-class ordinal classification or regression algorithms can be used

Evaluation

How do we quantify how well our Sentiment Analysis systems work?

- Create experimental datasets (aka test corpora): i.e., text segments that have been classified by humans, e.g. positive vs negative
- Compare (positive vs negative) system to human classifications
- Compute metrics like

$$\text{Accuracy} = \frac{\# \text{ correctly classified texts}}{\# \text{ texts}}$$

$$\text{Precision Pos} = \frac{\# \text{ texts correctly classified as positive}}{\# \text{ texts classified as positive}}$$

$$\text{Recall Pos} = \frac{\# \text{ texts correctly classified as positive}}{\# \text{ positive texts}}$$

$$\text{F-measure Pos} = \frac{2 * \text{Precision Pos} * \text{Recall Pos}}{\text{Precision Pos} + \text{Recall Pos}}$$

Same for **negative** class.

Baseline: most frequent class in the training set.

Conclusions

- Naïve Bayes classifier:
 - ◊ Really easy to implement and often works well
 - ◊ Often a good first thing to try
- Actually, the Naïve Bayes assumption is almost never true
- Still, Naïve Bayes often performs surprisingly well even when its assumption does not hold
- SA is an exciting topic, many applications, huge market for systems, particularly in focused domains.
- Promising results with simple techniques, but many interesting research challenges to be addressed for high accuracy.

Extra reading

Bing Liu and Lei Zhang (2012). A survey on opinion mining and sentiment analysis. Kluwer Academic Publishers:

http://www.cs.uic.edu/~lzhang3/paper/opinion_survey.pdf

Bing Liu (2012). Sentiment Analysis and Opinion Mining. Morgan and Claypool Publishers. Draft on line at: <https://www.cs.uic.edu/~liub/FBS/SentimentAnalysis-and-OpinionMining.pdf>

Article on SemEval in Wikipedia:

<https://en.wikipedia.org/wiki/SemEval>.

COM6115: Text Processing

Natural Language Generation

Chenghua Lin

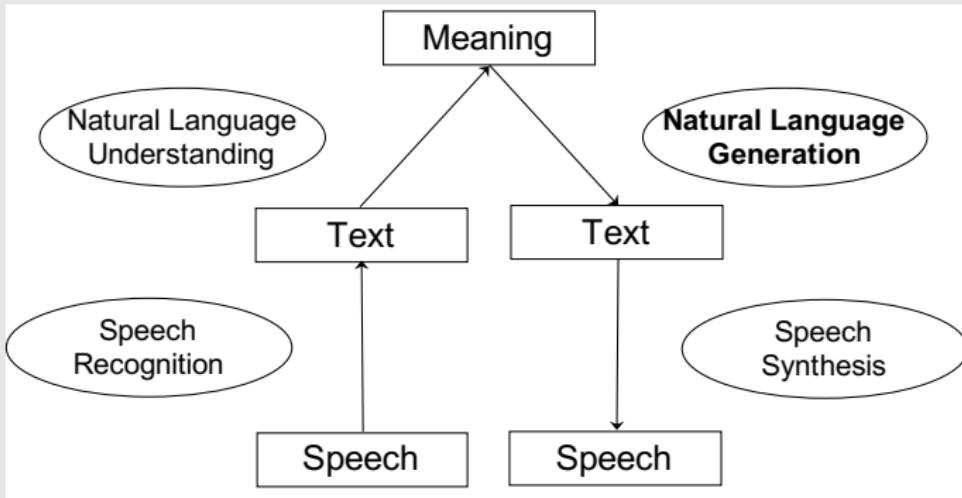
Department of Computer Science
University of Sheffield

What is NLG?

Background read: “*Building Natural Language Generation Systems*”
– Reiter and Dale

- Defined as the task of generating text (or speech) from non-linguistic input.
 - ◊ Data: numbers, RDF triples, etc
 - ◊ Output is documents, reports, explanations, help messages, and other kinds of texts
- Requires
 - ◊ Knowledge of language
 - ◊ Knowledge of the domain

Language Technology



First Example: Weather Forecasts

- Input: numerical weather predictions
 - ◊ From supercomputer running a numerical weather simulation
- Output: textual weather forecast
 - ◊ Users prefer some NLG texts over human texts!
 - ◊ More consistent, better word choice

Simple example: Point weather forecast

London Heathrow Airport [Change table layout](#)

Tue 4 Mar Wed 5 Mar Thu 6 Mar Fri 7 Mar Sat 8 Mar

06:00 Wed 05 Mar 2014 - 06:00 Thu 06 Mar 2014

Sunshine from mid-morning and into the afternoon. Staying dry, but becoming cloudier from early evening and into Thursday. It is likely to feel milder than on Tuesday with a maximum temperature during the afternoon in the region of 11C and a minimum temperature overnight of around 6C. Light winds throughout.

UK local time	Warnings for Greater London	Weather	Precip. (%)	Temp. (°C)	Feels like (°C)	Wind speed & direction (mph)	Wind gusts (mph)	Visibility	Humidity (%)	UV index	Daily air quality index [BETA]
0000	No warnings		<5				No gusts	Moderate	90		
0300	No warnings		<5				No gusts	Moderate	92		

Example 1: Met Office NLG System

- Input:
 - ◊ Weather prediction data of temperature, wind speed and direction, precipitation and visibility, etc.;
 - ◊ Daily summary weather prediction data of average daily and nightly values for parameters as above; and
 - ◊ Seasonal averages (lows, highs and mean) for temperature.
- Output: weather forecast texts
- NLG system vs. manual report writing
 - ◊ Volume: satellite cloud data is gathered at a speed of 158M per second
 - ◊ Time: NLG system (< 30 secs) vs. human expert (hours)

Example 1: Met Office NLG System

Weather and climate change > www.metoffice.gov.uk

Apps Suggested Sites Web Slice Gallery http://spe.sysu.edu.i... iServe Browser OpenRDF Workbench W seed DiscOU Sentiment Analysis Django: Passing arg...

 Met Office Email alerts | Contact us Search

Weather Climate Learning Research Products News Holiday weather Get ready for winter

This website uses cookies. [Read about how we use cookies.](#)



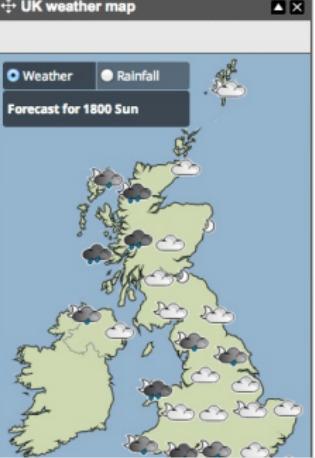
Aberdeen
Find a weather forecast
Enter place name or postcode 
Nearby locations  Recent locations (3) 

Five day forecast

Day	Weather	Temperature (°C)	Wind (mph)
Sun	   	Max.  Min. 	12

UK weather map

Weather Rainfall Forecast for 1800 Sun



News

Landmark report on climate change finalised 

E97m supercomputer makes UK world-leader in weather and climate science  27 Oct 2014
A new E97m Met Office supercomputer will cement the UK's position as a world leader in weather and climate prediction.

Cray announced as supplier for Met Office supercomputer

Winter Weather Planning  Updated: 31 Oct 2014

Data Input

```
96,122,1,5,2.00,200,-14.41,-3.668,-1.431,.345,1023,15.41,15.82,20.07,-11.1,-2.878,104.2,28,153.6,53.19,0,16.26
96,122,1,5,2.25,215,-10.72,-3.241,-1.35,.152,1023,15.3,15.78,20.07,-11.42,-2.762,105,.208,98.2,.822,0,17.05
96,122,1,5,2.50,230,-8.37,-1.282,-.904,2.15,1022,15.3,15.71,20.05,-11.66,-3.206,104.4,2,141.6,42.96,0,17.7
96,122,1,5,2.75,245,-12.81,-2.11,-1.067,2.119,1022,15.33,15.79,19.99,-11.15,-3.093,104.8,.2,186.5,11.32,0,17.81
96,122,1,5,3.00,300,-13.68,-3,-1.35,1.075,1022,15.36,15.79,19.96,-10.63,-3.005,104.6,.402,285.8,61.45,0,18.47
96,122,1,5,3.25,315,-10.2,-2.457,-1.13,.-.73,1022,15.32,15.66,19.92,-11.17,-3.263,103.6,.304,354.7,36.29,0,19.03
96,122,1,5,3.50,330,-9.33,-1.353,-.942,.902,1022,15.21,15.62,19.9,-10.95,-2.903,104.3,.313,302.2,34.69,0,19.16
96,122,1,5,3.75,345,-7.29,.-.285,.-.76,2.048,1022,15.24,15.63,19.87,-10.68,-3.27,104,.252,313,29.7,0,19.61
96,122,1,5,4.00,400,-6.822,-.365,-.653,1.531,1022,15.25,15.63,19.83,-9.93,-3.316,104,.331,274.2,52.98,0,20.42
96,122,1,5,4.25,415,-8.78,-.65,-.747,1.602,1023,15.35,15.66,19.79,-9.77,-2.656,103.3,.253,247.7,10.99,0,21.08
96,122,1,5,4.50,430,-8.73,-.641,-.741,1.785,1023,15.46,15.81,19.75,-9.16,-2.782,103.7,.2,295,29.15,0,21.3
96,122,1,5,4.75,445,-11.45,-2.671,-1.03,-.456,1022,15.46,15.82,19.74,-8.81,-2.464,103.7,.2,355.3,23.98,0,21.65
96,122,1,5,5.00,500,-13.12,-4.3,-1.306,1.359,1022,15.42,15.75,19.76,-9.39,-2.49,103.4,.2,20.67,.188,0,21.83
96,122,1,5,5.25,515,-13.62,-4.621,-1.344,-.842,1022,15.32,15.67,19.81,-9.47,-2.703,103.7,.2,20.65,.183,0,21.98
96,122,1,5,5.50,530,-13.8,-3.534,-1.325,.943,1022,15.23,15.61,19.86,-10.92,-3.384,103.9,.2,20.65,.183,0,22.14
96,122,1,5,5.75,545,-14.7,-3.748,-1.419,.385,1022,15.06,15.47,19.9,-11.62,-2.868,104.4,.2,341.6,18.6,0,22.36
96,122,1,5,6.00,600,-13.61,-2.315,-1.287,2.038,1022,14.98,15.42,19.9,-12.37,-3.092,104.7,.2,298.6,5.173,0,22.54
96,122,1,5,6.25,615,-14,-2.894,1.293,.669,1022,14.92,15.36,19.88,-12.48,-3.808,104.7,.591,320.3,21.07,0,22.87
```

Example 1: Met Office NLG System

Forecast summary

Regional UK 5 days UK 6-30 days

Regional forecast for Grampian

Rain edging northwards during Monday morning. Becoming drier later.

This Evening and Tonight:

Dry this evening and for most of the night with some clear spells. Becoming cold with a few mist or fog patches forming as winds fall light. Showers will spread up into southern Aberdeenshire by morning. Minimum Temperature 2C.

Monday:

Dry, bright start, soon becoming cloudy. Showers or longer spells of over southern Aberdeenshire will edge northwards to all parts during the morning. Becoming drier from south during afternoon. Maximum Temperature 10C.

Outlook for Tuesday to Thursday:

Frequent showers Tuesday, wintry on hills later, as winds turn more northerly. Some showers early Wednesday, hill snow, then dry and bright. Frost overnight then bright start Thursday, rain later.

Issued at: 1600 on Sun 02 Nov 2014

Weather map

Sun 1600

Aberlour
Aberdeen
Dochry
Dundee

Weather forecast map for Aberdeen

Location Details

Aberdeen

Location: 57.1498, -2.0927

Altitude: 19m above mean sea level

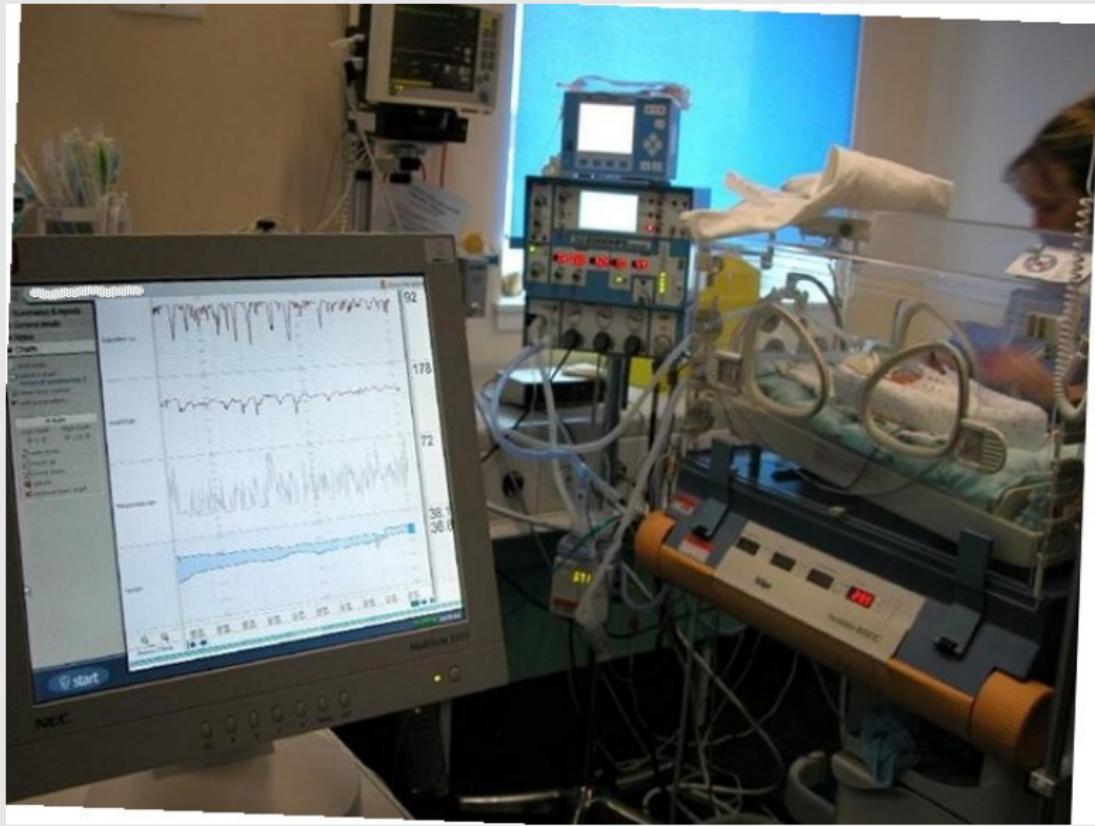
Video forecast

Sunday's Forecast

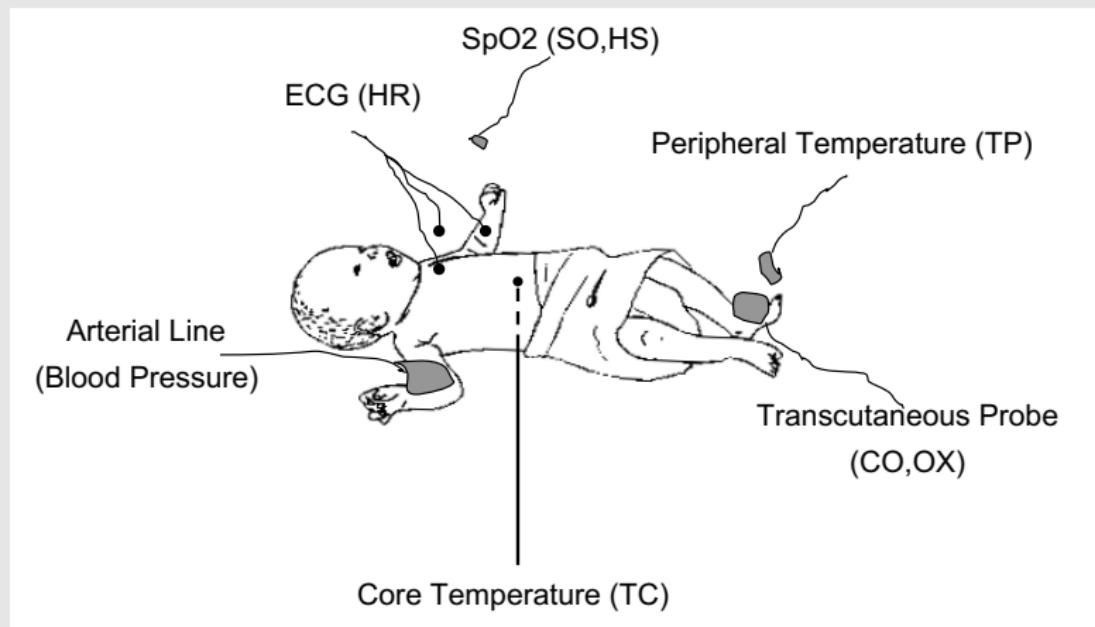
Example 2: BabyTalk

- Goal: Summarise clinical data about premature babies in neonatal ICU
- Input: sensor data; records of actions/observations by medical staff
- Output: multi-para texts, summarise
 - ◊ BT45: 45 mins data, for doctors
 - ◊ BT-Nurse: 12 hrs data, for nurses
 - ◊ BT-Family: 24 hrs data, for parents

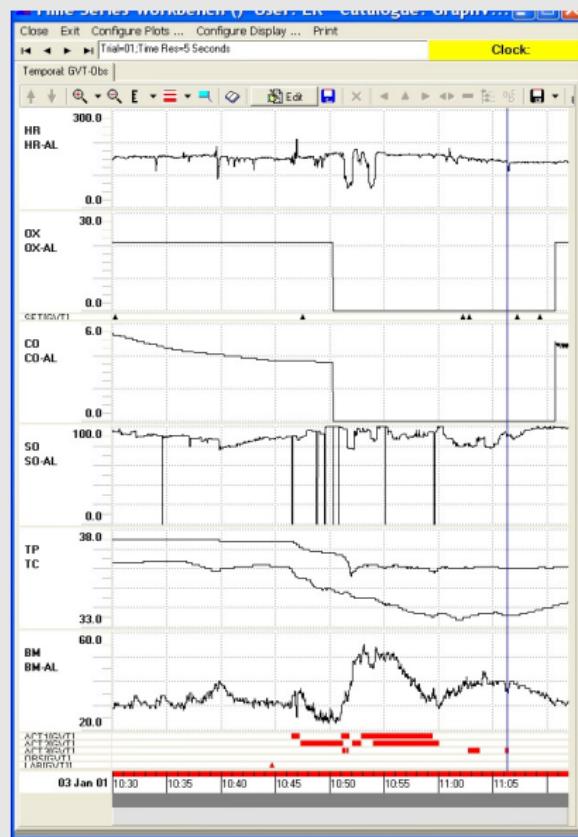
Neonatal ICU



Baby Monitoring



Input: Sensor Data



Input: Action Records

FullDescriptor	Time
SETTING;VENTILATOR;FiO2 (36%)	10.30
MEDICATION;Morphine	10.44
ACTION;CARE;TURN/CHANGE POSITION;SUPINE	10.46-10.47
ACTION;RESPIRATION;HAND-BAG BABY	10.47-10.51
SETTING;VENTILATOR;FiO2 (60%)	10.47
ACTION;RESPIRATION;INTUBATE	10.51-10.52

BT45 text (extract)

Computer-generated text

- By 11:00 the baby had been hand-bagged a number of times causing 2 successive bradycardias. She was successfully reintubated after 2 attempts. The baby was sucked out twice.
At 11:02 FIO₂ was raised to 79%.

BT-Nurse text (extract)

Respiratory Support

Current Status

Currently, the baby is on CMV in 27 % O₂. Vent RR is 55 breaths per minute. Pressures are 20/4 cms H₂O. Tidal volume is 1.5.

SaO₂ is variable within the acceptable range and there have been some desaturations.

...

Events During the Shift

A blood gas was taken at around 19:45. Parameters were acceptable. pH was 7.18. CO₂ was 7.71 kPa. BE was -4.8 mmol/L.

...

BT-Family text (extract)

John was in intensive care. He was stable during the day and night. Since last week, his weight increased from 860 grams (1 lb 14 oz) to 1113 grams (2 lb 7 oz). He was nursed in an incubator.

Yesterday, John was on a ventilator. The mode of ventilation is Bilevel Positive Airway Pressure (BiPAP) Ventilation. This machine helps to provide the support that enables him to breathe more comfortably. Since last week, his inspired Oxygen (FiO₂) was lowered from 56 % to 21 % (which is the same as normal air). This is a positive development for your child.

During the day, Nurse Johnson looked after your baby. Nurse Stevens cared for your baby during the night.

Example 3: Summary of the NBA Game statistics

- Goal: Generate documents to summarise the game statistics of NBA.
 - ◆ In addition to capturing the writing style, a generation system should select record content, express it clearly, and order it appropriately.
- Input: structured data table capturing statistics info of games
- Output: a game summary document

Exam 3: NBA - Training Corpus

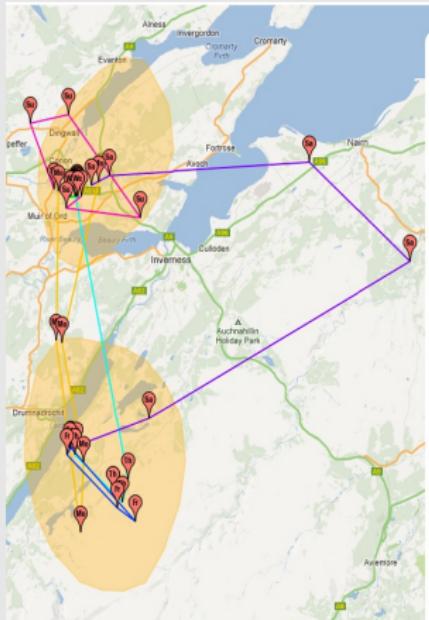
TEAM	WIN	LOSS	PTS	FG.PCT	RB	AS ...
Heat	11	12	103	49	47	27
Hawks	7	15	95	43	33	20

PLAYER	AS	RB	PT	FG	FGA	CITY ...
Tyler Johnson	5	2	27	8	16	Miami
Dwight Howard	4	17	23	9	11	Atlanta
Paul Millsap	2	9	21	8	12	Atlanta
Goran Dragic	4	2	21	8	17	Miami
Wayne Ellington	2	3	19	7	15	Miami
Dennis Schroder	7	4	17	8	15	Atlanta
Rodney McGruder	5	5	11	3	8	Miami
Thabo Sefolosha	5	5	10	5	11	Atlanta
Kyle Korver	5	3	9	3	9	Atlanta
...						

The Atlanta Hawks defeated the Miami Heat , 103 - 95 , at Philips Arena on Wednesday . Atlanta was in desperate need of a win and they were able to take care of a shorthanded Miami team here . Defense was key for the Hawks , as they held the Heat to 42 percent shooting and forced them to commit 16 turnovers . Atlanta also dominated in the paint , winning the rebounding battle , 47 - 34 , and outscoring them in the paint 58 - 26. The Hawks shot 49 percent from the field and assisted on 27 of their 43 made baskets . This was a near wire - to - wire win for the Hawks , as Miami held just one lead in the first five minutes . Miami (7 - 15) are as beat - up as anyone right now and it 's taking a toll on the heavily used starters . Hassan Whiteside really struggled in this game , as he amassed eight points , 12 rebounds and one blocks on 4 - of - 12 shooting ...

"An example data-record and document pair from the ROTOWIRE dataset. We show a subset of the game's records (there are 628 in total), and a selection from the gold document. The document mentions only a select subset of the records, but may express them in a complicated manner."

Exam 3: Blogging Birds



Eyes to the Skies

Wyvis Moray Millie Ussie

Millie's journeys from 2013-04-08 to 2013-04-14

This week Millie was feeling restless. She predominantly flew around Easter Kinkell and Farraline and made several excursions clocking up about 264 kms. During this week, Millie's foraging patterns have been varied and she roosted in many woodlands on the move.

On Monday morning amid overcast conditions she was observed flying past the Beauly Firth to reach Teavarran 17.0 km away from where she started. On Tuesday and Wednesday she did not travel much and stayed in the Rootfield area. On Thursday morning she was observed flying down to Farraline, passing Loch Ness and Loch Ruthven. In the afternoon she was spotted in rough grassland near Easter Aberchalter, maybe feeding on small mammals, before retiring to the roost in woodland near Errigoe.

[read more....](#)

Select the week:

April 2013						
Mo	Tu	We	Th	Fr	Sa	Su
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4

Click here for daily blog:
[Mo](#) [Tu](#) [We](#) [Th](#) [Fr](#) [Sa](#) [Su](#)

Bio:

2012 female bird, which fledged from a nest near Culbokie. Named after the scientific name for the Red kite *Milvus milvus*.

Blogging Birds

Wyvis's Journeys (20/09/12 to 26/09/12) , Female bird born in 2012

DOW	Hour	Habitat	Significant Weather	Temp (C)	Visibility (m)	Wind Speed (mph)	Location	Features	Distance Flown	Other Kites
Friday	8	coniferous woodland	overcast	13.0	24000	3.0	East Croachy	Loch Ruthven,	0.0	
	10	rough grassland	heavy rain	13.9	5000	2.0	Torness	Loch Ruthven,	4.0	
	12	rough grassland	heavy rain	16.0	3600	1.0	Torness	Loch Ruthven,	2.0	
	14	rough grassland	heavy rain	16.0	3600	1.0	Torness	Loch Ruthven,	2.0	Merida
	16	improved grassland	overcast	18.4	45000	5.0	Torness		3.0	

Other NLG projects

- Automatic journalism
 - ◊ <http://www.bbc.co.uk/news/technology-34204052>
- Assistive technology: help people with learning disabilities, blind people, deaf people, ...
- Education: computerised tutoring systems, feedback on assessments
- Image labelling
- Agent and dialogue systems (e.g., Siri, Cortana)
- Etc, etc

Image labelling example

Describes without errors	Describes with minor errors	Somewhat related to the image	Unrelated to the image
 A person riding a motorcycle on a dirt road.	 Two dogs play in the grass.	 A skateboarder does a trick on a ramp.	 A dog is jumping to catch a frisbee.
 A group of young people playing a game of frisbee.	 Two hockey players are fighting over the puck.	 A little girl in a pink hat is blowing bubbles.	 A refrigerator filled with lots of food and drinks.
 A herd of elephants walking across a dry grass field.	 A close up of a cat laying on a couch.	 A red motorcycle parked on the side of the road.	 A yellow school bus parked in a parking lot.

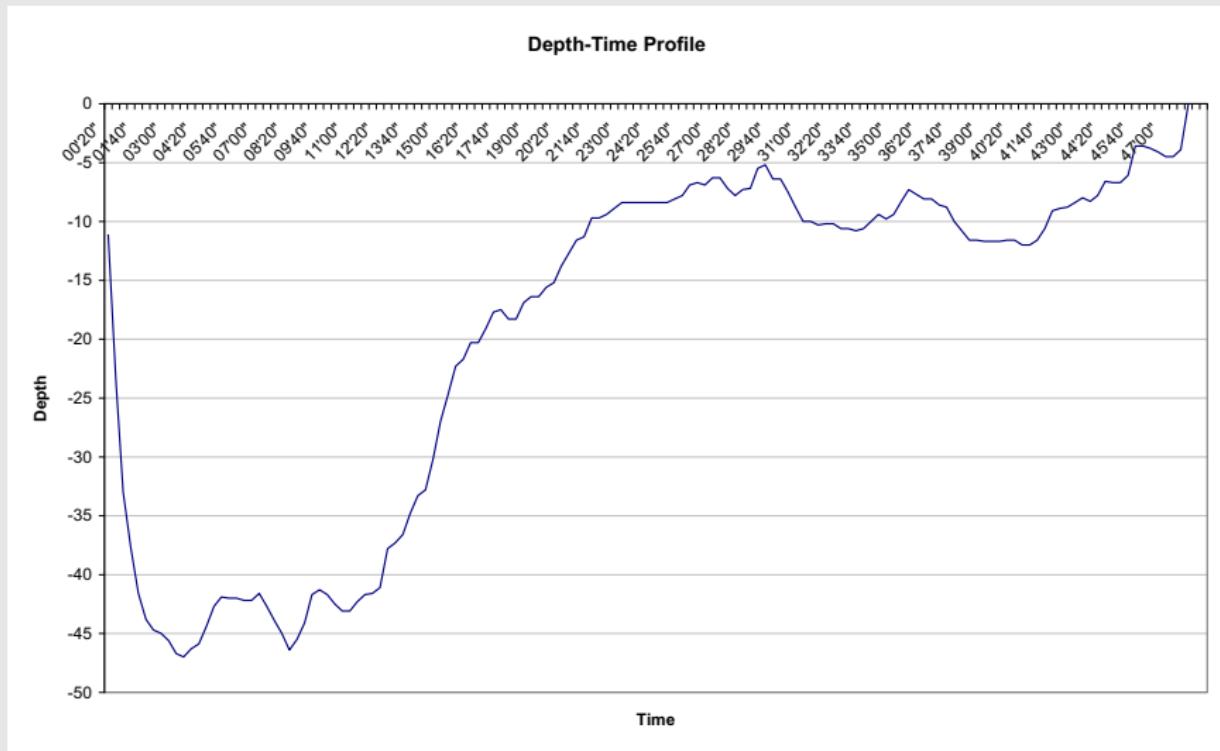
How do NLG Systems Work?

- Usually three stages
 - ◊ Not including data analysis
- **Document planning:** decide on content and structure of text
- **Microplanning:** decide how to linguistically express text (which words, sentences, etc to use)
- **Realisation:** grammatical details
 - ◊ E.g. *children* vs. *childs*, *an apple* vs. *a apple*

Scubatext example

- Demo system for scuba divers
- Input is *dive computer data*
 - ◊ Depth-time profile of scuba dive
- Output is feedback to diver
 - ◊ Mistakes, what to do better next time
 - ◊ Encouragement of things done well

Scuba - input



Scuba – output

- “Risky dive with some minor problems. Because your bottom time of 12 min exceeds no-stop limit by 4 min this dive is risky. But you performed the ascent well. Your buoyancy control in the bottom zone was poor as indicated by ‘saw tooth’ patterns.”

Scuba: data analytics

- Look for trends and patterns in data
 - ◊ Trends: e.g., depth increases fairly steadily over first 3 minutes
 - ◊ Patterns: e.g., sawtooth between 3 and 15 minutes
- Will not further discuss here

Document Planning

- Content selection: of the zillions of things I could say, which should I say?
 - ◊ Depends on what is important
 - ◊ What makes good narrative
 - ◊ What is easy to say
- Structure: How should I organise this content as a text?
 - ◊ What order do I say things in?
 - ◊ Rhetorical structure?

Scuba: content

- Probably focus on patterns indicating dangerous activities
 - ◊ E.g., most important thing to mention
- How much should we say about these?
 - ◊ Detail? Explanations?
- Encourage/praise good diving
 - ◊ Positive feedback is important

Scuba: structure

- Mention most dangerous thing first?
 - ◊ Or should we just order by time?
 - ◊ Start with overview?
- Linking words (cue phrases)
 - ◊ Also, but, because, ...

Microplanning

- Lexical/syntactic choice: Which words and linguistic structures to use?
- Aggregation: How should information be distributed across sentences and paras
- Reference: How should the text refer to objects and entities?

SCUBA: microplanning

- Lexical/syntactic choice:
 - ◊ *risky* vs. *dangerous* vs. *unwise* vs. ...
 - ◊ *performed the ascent* vs. *ascended* vs ...
 - ◊ *12 min* vs. *720 sec* vs. *714.56 sec*
- Aggregation: 1 sentence or 2 sentences?
 - ◊ “Because your bottom time of 12 min exceeds no-stop limit by 4 min this dive is risky, but you performed the ascent well.”

Scuba: Microplanning

- Aggregation (continued)
 - ◊ Phrase merging
 - “Your first ascent was fine. Your second ascent was fine” vs.
 - “Your first and second ascents were fine.”
 - ◊ Reference
 - Your ascent vs.
 - Your first ascent vs.
 - Your ascent from 33m at 3 min

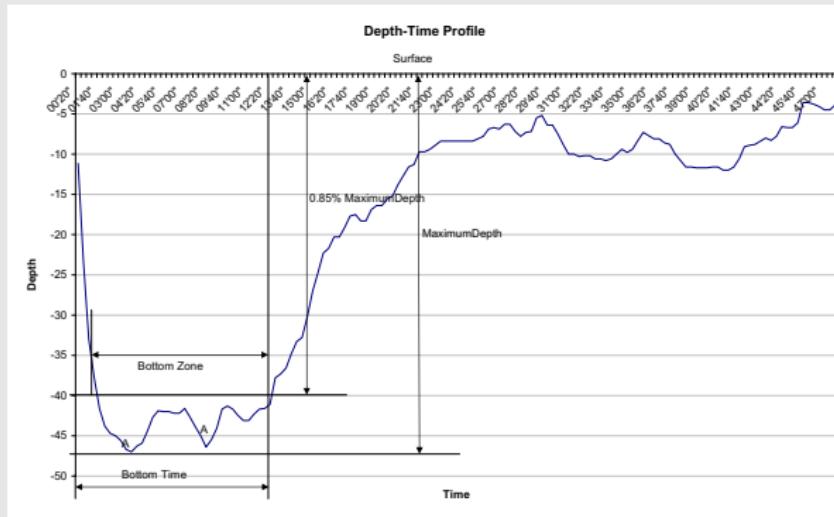
- Grammars (linguistic): Form legal English sentences based on decisions made in previous stages
 - ◊ Obey sub-languages, i.e., language of a restricted domain, particularly a technical domain.
 - ◊ genre constraints, e.g., scientific writing vs. social media text, etc.
- Structure: Form legal HTML, RTF, or whatever output format is desired

Scuba: Realisation

- Simple linguistic processing
 - ◊ Capitalise first word of sentence
 - ◊ Subject-verb agreement
 - Your first ascent was fine
 - Your first and second ascents were fine
- Structure
 - ◊ Inserting line breaks in text
 - ◊ Add HTML markups, eg, <P>

- Speech output
- Text and visualisations
 - ◊ Produce separately, OR
 - ◊ Tight integration
 - E.g., text refers to graphic, OR
 - graphs has text annotations

Combined (Preferred)



Risky dive with some minor problems. Because your bottom time of 12.0min exceeds no-stop limit by 4.0min this dive is risky. But you performed the ascent well. Your buoyanc control in the bottom zone was poor as indicated by 'saw tooth' patterns marked 'A' on the depth-time profile.

Building NLG Systems

- Knowledge and corpus analysis
- Evaluation

Building NLG Systems: Knowledge

- Need knowledge
 - ◊ Which patterns most important?
 - ◊ What order to use?
 - ◊ Which words to use?
 - ◊ When to merge phrases?
 - ◊ Etc.
- Where does this come from?

Knowledge Sources

- Imitate a *corpus* of human-written texts
 - ◊ Most straightforward
 - ◊ Manually examine
 - ◊ Use learning if corpus is large enough
- Ask domain experts
 - ◊ Experts bad at explaining what they do
 - ◊ Better at critiquing what system does
- Experiments with users
 - ◊ Very nice in principle, but a lot of work

Scuba: Corpus

- See which patterns humans mention in the corpus, and have the system mention these
- See the words used by humans, and have the system use these as well
- etc.

Evaluation

- Does system help people?
 - ◊ Do divers dive more safely when they use Scuba NLG system
- Do people like the texts
 - ◊ Do divers consider Scuba to be useful?
- Comparison to human texts
 - ◊ Are Scuba texts similar to corpus texts

NLG vs NLP

- Producing rather than understanding language
- Focus on content and AI issues as well as linguistic issues
- Increasing uptake of statistical and deep learning techniques

COM6115: Text Processing

Natural Language Generation 2

Chenghua Lin

Department of Computer Science
University of Sheffield

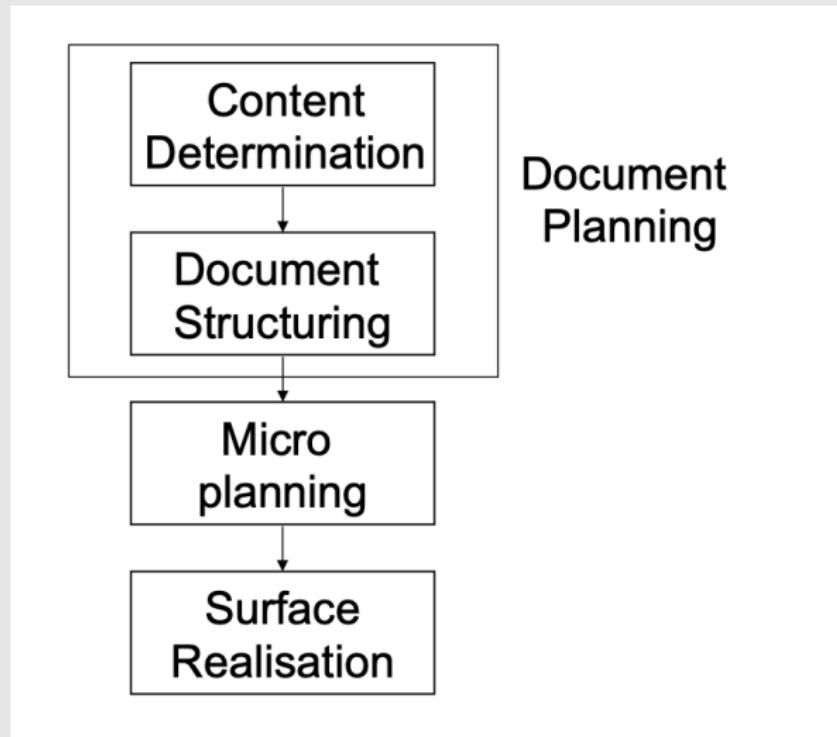


There are three rules for writing a novel.
Unfortunately, no one knows what they are.

(W. Somerset Maugham)

[izquotes.com](#)

The Architectural View



Document Planning

- Problem: Usually the output text can only communicate a small portion of the input data
 - ◊ Which bits should be communicated?
 - ◊ How should information be ordered and structured?
- First stage of NLG
- Goals:
 - ◊ **Decide on content:** to determine what information to communicate
 - ◊ **Decide on rhetorical structure:** to determine how to structure the information to make a coherent text

How to Choose Content

Feature	P	R	F1	Acc	P	R	F1	Acc	P	R	F1	Acc	P	R	F1	Acc
	Academic				Conversation				Fiction				News			
GloVe	65.2	67.5	66.3	92.1	58.4	62.6	60.4	94.1	60.1	55.6	57.8	91.4	69.3	64.9	67.0	90.1
ELMo	65.1	74.1	69.3	92.4	67.6	65.1	66.4	95.2	62.3	68.4	65.2	92.3	72.6	73.4	73.0	91.6
BERT ₁₇	67.3	71.7	69.4	92.7	70.9	63.0	67.7	95.7	70.3	65.9	68.1	93.5	74.0	71.1	72.6	91.6
GE	66.9	74.6	70.5	92.8	63.3	69.3	66.1	94.9	65.8	65.5	65.7	92.8	73.1	74.5	73.8	91.8
GB ₁₇	64.7	77.2	70.4	92.5	68.1	67.5	67.8	95.7	70.3	67.6	68.9	93.7	74.3	71.5	72.9	91.7
EB ₁₇	71.8	72.3	72.0	93.5	69.9	66.3	68.1	95.8	72.9	64.8	68.6	93.6	76.1	70.5	73.2	91.9
GEB ₁₇	72.7	72.0	72.3	93.8	74.0	64.9	69.1	95.9	75.9	67.1	71.2	94.3	77.7	71.4	74.4	92.4
	Verb				Adjective				Noun				Adverb			
GloVe	60.2	57.2	58.7	84.9	54.9	42.2	47.7	90.1	59.1	50.5	54.5	88.6	49.4	49.4	49.4	93.0
ELMo	62.7	70.3	66.3	86.6	46.7	54.9	50.5	88.5	61.5	58.6	60.0	89.5	57.6	51.9	54.6	94.0
BERT ₁₇	63.3	72.2	67.5	89.9	54.7	49.1	51.8	90.2	66.8	51.7	58.3	90.0	66.7	45.5	54.1	94.7
GE	62.4	68.9	65.5	86.4	56.9	58.7	57.8	90.8	62.4	59.9	61.1	90.1	53.7	56.5	55.1	93.6
GB ₁₇	64.7	69.1	66.8	87.1	58.4	53.8	56.0	90.9	65.0	57.7	61.1	90.1	61.3	49.4	54.7	94.3
EB ₁₇	66.9	69.0	67.9	87.8	53.7	53.2	53.4	90.1	73.4	49.5	59.1	90.8	63.3	49.4	55.5	94.5
GEB ₁₇	71.6	67.4	69.4	88.9	62.8	53.5	57.8	91.6	69.9	54.5	61.3	90.7	69.1	49.4	57.6	95.0

Table 3: Word embedding feature analysis on different genres and PoS of VUA-all-POS development set.

Content Determination

- The most important aspect of NLG!
 - ◊ If we get content right, users may not be too fussed if language isn't perfect
 - ◊ If we get content wrong, users will be unhappy even if language is perfect
- Also the most domain-dependent aspect
 - ◊ Based on domain, user, tasks more than general knowledge about language

How to Choose Content

- Theoretical approach: deep reasoning based on deep knowledge of user, task, context, etc
- Pragmatic approach: write schemas which try to imitate human-written texts in a corpus
- Statistical approach: use learning techniques to learn content rules from corpus

Theoretical Approach

- Deduce what the user needs to know, and communicate this
- Based on in-depth knowledge
 - ◊ User (knowledge, task, etc)
 - ◊ Context, domain, world
- Use AI reasoning engine
 - ◊ e.g. applies logical rules to the knowledge base to deduce new information
- Not feasible in practice
 - ◊ Lack knowledge about user
 - ◊ Lack knowledge of context
 - ◊ Very hard to maintain knowledge base, e.g., new users, new regulations, etc.

Statistical Approach

- Statistical/learning techniques (including deep learning)
 - ◊ Parse corpus, align with source data, use machine learning algorithms to learn content selection rules/schemas/cases
 - Modelling the coherence of discourse, Barzilay and Lapata, 2005
 - NBA boxscore-data, Wiseman et al, 2017
 - E2E Challenge, Dušek et al, 2019
- Worth considering if large corpora available

Pragmatic Approach: Schema

- Analyse corpus texts (after aligning them to data), and manually infer content and structure rules.
- Typically based on imitating patterns seen in human-written texts (i.e., corpus)
 - ◆ Revised based on user feedback
- Specify structure as well as content

- Cue phrases: linguistic expressions such as that may explicitly mark the structure of a discourse.
- Rhetorical Relations (RR) describe how the parts of a text are linked to each other, and can be expressed via cue phrases.
 - ◊ Best cue phrase for RR depends on context
 - Your first ascent was a bit rapid. However, your second ascent was fine.
 - Your first ascent was a bit rapid, but your second ascent was fine.
 - ◊ Also readers like cue phrases to be varied, not same one used again and again
 - Eg, don't overuse "for example"

Text Structure

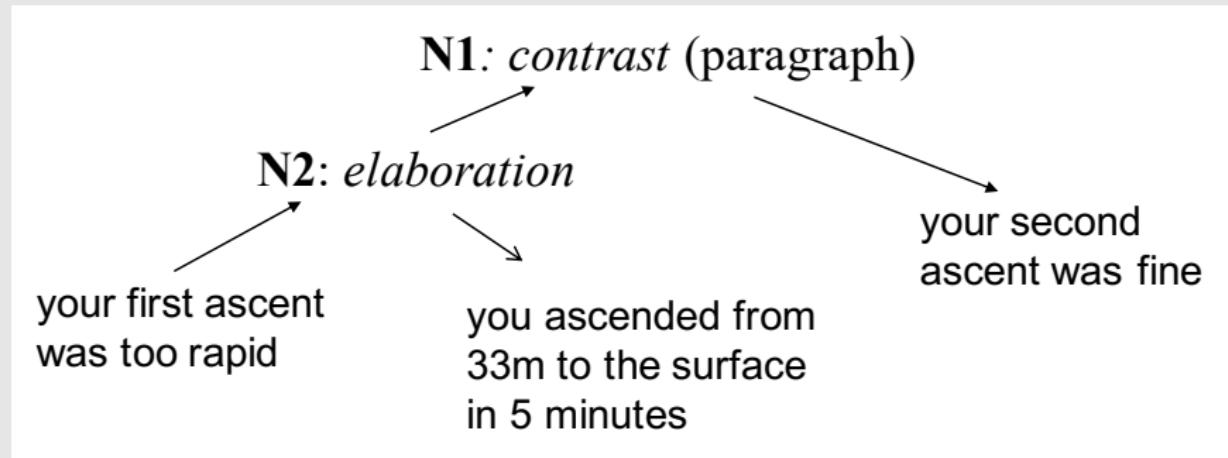
Rhetorical Relations: describe how the parts of a text are linked to each other. The common ones are:

- CONCESSION (although, despite)
- CONTRAST (but, however)
- ELABORATION (usually no cue)
- EXAMPLE (for example, for instance)
- REASON (because, since)
- SEQUENCE (and, also)

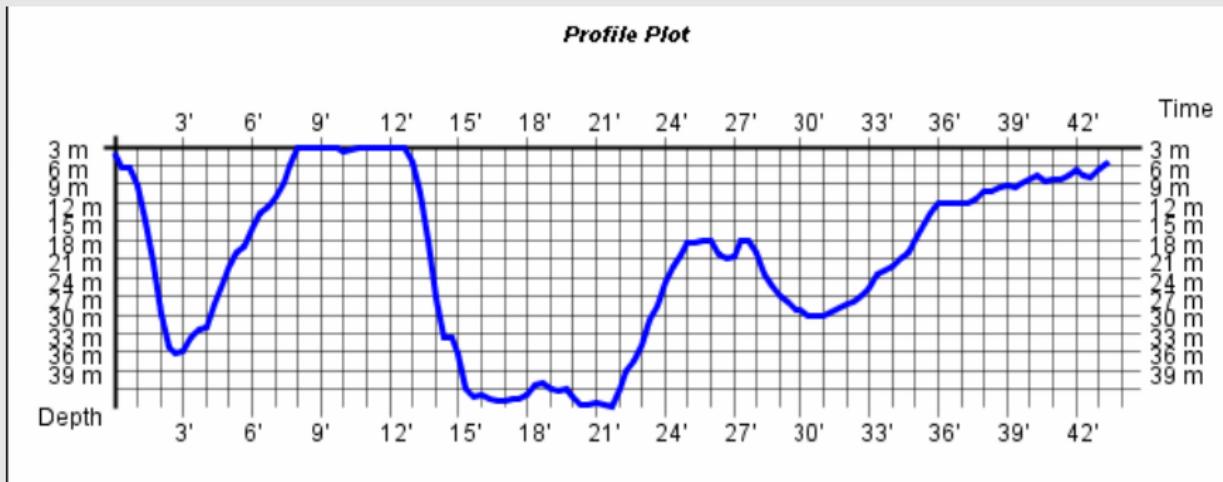
Research community does not agree; many different sets of rhetorical relations proposed.

Example

- Your first ascent was too rapid; you ascended from 33m to the surface in 5 minutes. However, your second ascent was fine.



Scubatext Example: Input



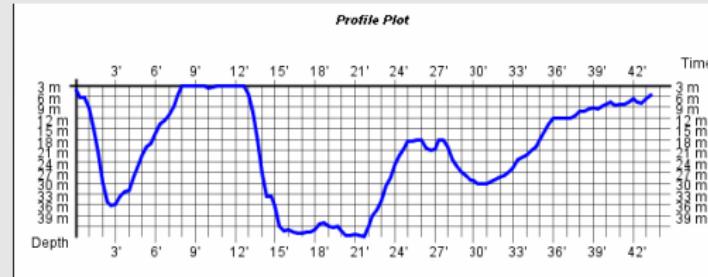
Input Segments

diveNo	segNo	iTime	iValue	fTime	fValue
1460	1	0	1.3	60	6.3
1460	2	60	6.3	140	32.2
1460	3	140	32.2	480	0
1460	4	480	0	760	0
1460	5	760	0	920	38.9
1460	6	920	38.9	1300	41.6
1460	7	1300	41.6	1500	15.5
1460	8	1500	15.5	1860	27.2
1460	9	1860	27.2	2160	9.2
1460	10	2160	9.2	2600	2.7

Corresponding Corpus Text

- Your first ascent was a bit rapid; you ascended from 33m to the surface in 5 minutes, it would have been better if you had taken more time to make this ascent. You also did not stop at 5m, we recommend that anyone diving beneath 12m should stop for 3 minutes at 5m. Your second ascent was fine.

Align corpus text with data



Input: 1460 3 140 32.2 480 0

Output: (representation of)

Your first ascent was a bit rapid; you ascended from 33m to the surface in 5 minutes, it would have been better if you had taken more time to make this ascent.

Input: 1460 10 2160 9.2 2600 2.7

Output: (representation of)

Your second ascent was fine.

Possible Content Rules

- Describe segments that end (near) 0
 - ◊ And that don't start at 0
 - ◊ Also segment at end of dive
- Give additional info about such segments whose slope is too high
 - ◊ Explain risk
 - ◊ Say what should have happened

Possible Ordering Rules

- Break up dive into sections
- For each section, start with most important safety issue (or say dive was fine if no safety issue)
- Then add less important safety issues
- Then say something about what was done well
- ...

More Examples

- We have just looked at one example here!
- Need to repeat process for at least 20-30 examples, which cover spread of possible cases (including special cases)
- Merge rules and deal with conflicts
 - ◊ Often causes by different corpus authors writing differently;
 - ◊ may give priority to one particular author, and imitate his style

Pseudocode example

```
Schema ScubaSchema
for each ascent A in data set
    if ascent is too fast
        add unsafeAscentSchema(A)
    else
        add safeAscentSchema(A)
set rhetorical relation
```

Creating Schemas

- Usually just written as code in Java or other standard programming languages
- Creating schemas is an **art**, no solid methodology (yet)
- Problems
 - ◊ Corpus texts likely to be inconsistent
 - Especially if several authors wrote texts
 - ◊ Some cases not covered in the corpus
 - Unusual cases, boundary cases

Advanced: User-Adaptation

- Texts should depend on
 - ◊ User's personality
 - ◊ User's domain knowledge (how much do we need to explain)
 - ◊ User's vocabulary (can we use technical terms in the text)
 - ◊ User's task (what does he need to know)
- Hard to get this information...

Personality and Perspectives

- Text can communicate perspectives, e.g.,
 - ◊ Smoking is killing you
 - ◊ If you keep on smoking, your health may keep on getting worse
 - ◊ If you stop smoking, your health is likely to improve
 - ◊ If you stop smoking, you'll feel better
- How to choose between these?
- Depends on personality of reader
 - ◊ Some people react better to positive messages, others to negative messages
 - ◊ Some react better to short direct messages, others want these weakened ("may", "is likely to")
 - ◊ Hard to predict...

Conclusion

- Content determination is the first and most important aspect of NLG
 - ◊ What information should we communicate?
- Mostly based on imitating what is observed in human-written texts
 - ◊ Using schemas, written in Java
- Also decide on structure
 - ◊ Tree structure, rhetorical relations

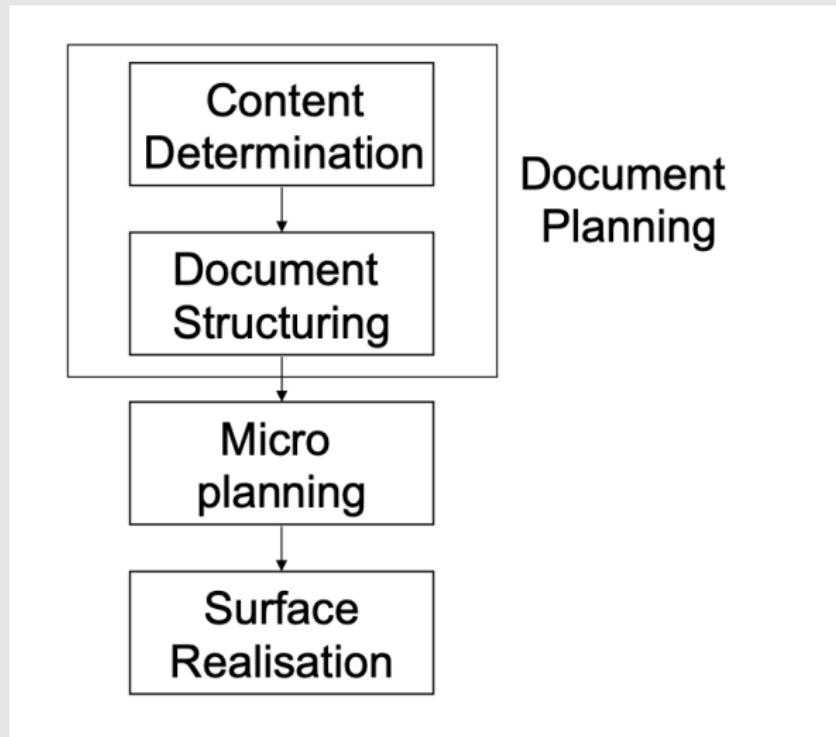
COM6115: Text Processing

Natural Language Generation 3

Chenghua Lin

Department of Computer Science
University of Sheffield

The Architectural View



Microplanning

- Second stage of NLG
 - ◊ Choosing language to express content
- Several subtasks
 - ◊ Lexical choice: Which words to use
 - ◊ Reference: How to refer to objects
 - ◊ Aggregation: How/when combine phrases into sentences

Microplanning

- Problem: There are zillions of ways of expressing a message in words
 - ◊ John sold the book to Mary
 - ◊ Mary bought the book from John
 - ◊ John sold the book. Mary bought it
 - ◊ Etc, etc
- Which one should we use?

Approaches

- Theoretical
 - ◊ Define what “best” means, make microplanning choices that optimise it
 - ◊ Hard to do in practice because we don’t have good models of the effects of choices
- Pragmatic
 - ◊ Imitate corpus
 - Use statistical learning if corpus large enough
 - ◊ Problem: sometimes corpus texts may not be very good from a microplanning perspective

Lexical choice

- Lexical choice: the task of choosing the right words or lemmas to express the contents of the message
- I.e., which word should be used to communicate a concept?
 - ◊ Buy vs sell
 - ◊ Ascended vs rose vs surfaced
 - ◊ Too fast vs too rapidly
 - ◊ Recommend vs suggest
 - ◊ etc

Issues that affect lexical choice

- Frequency (affects readability)
 - ◊ lie vs prevarication
- Formality:
 - ◊ Error vs howler
- Focus, expectations
 - ◊ not many, few, a few, only a few [students failed the exam]
- Technical terms
 - ◊ (statistics) standard error, not
 - ◊ standard mistake
- Convention
 - ◊ Temperature falls, Wind speed eases

Corpus-based Approach: Example

Statistics-Based Lexical Choice for NLG from Quantitative Information

Motivation

- NLG systems express information in human language

Forecasted numeric data			Forecast Text
Wind Direction (azimuth)	Wind Speed (knots)	Gust (knots)	
2	9	11	
92	20	30	
130	4	5	

Motivation

- Systems need to “know” what expressions are most suitable for expressing a given piece of information.

wd=130	→	“SE”
wd=92	→	“E”
ws=4	→	“8 OR LESS”



Motivation

- Systems need to “know” what expressions are most suitable for expressing a given piece of information.



Our Goal

- To develop a statistical algorithm for lexical choice for quantitative information, which can
 - ◊ Detect the relationship between data dimensions (aka. attributes) and words
 - ◊ Does not rely on hand-crafted rules;
 - ◊ Predict both when and which word(s) should be used;
 - ◊ One word can refer to multiple dimensions.

$$P(\text{"muggy"} \mid \text{ws}=20, \text{temp}=35, \text{humid}=97, \dots)$$

Methodology

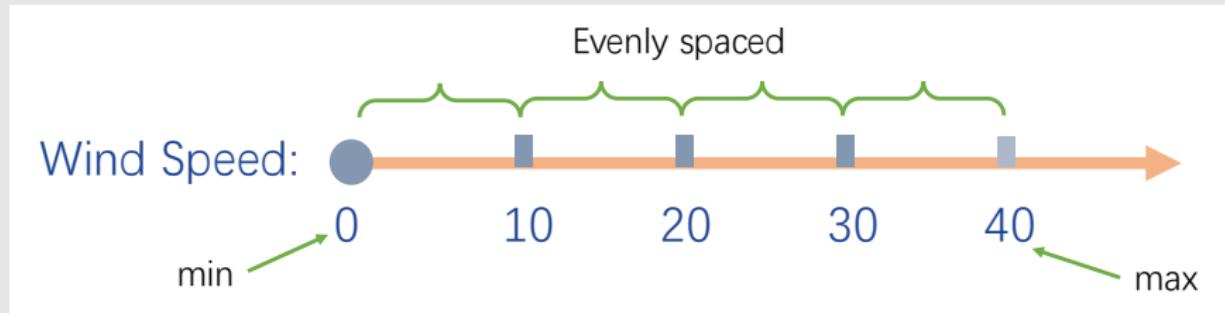
- Each data record consists of attribute-value pairs.
- E.g., dir=2,ws=9,gusts=11, where the attributes are “dir”, “ws”, and “gusts”.

Forecasted numerical data		
Wind Direction (azimuth)	Wind Speed (knots)	Gust (knots)
2	9	11
92	20	30
130	4	5

Representing Data in Vector

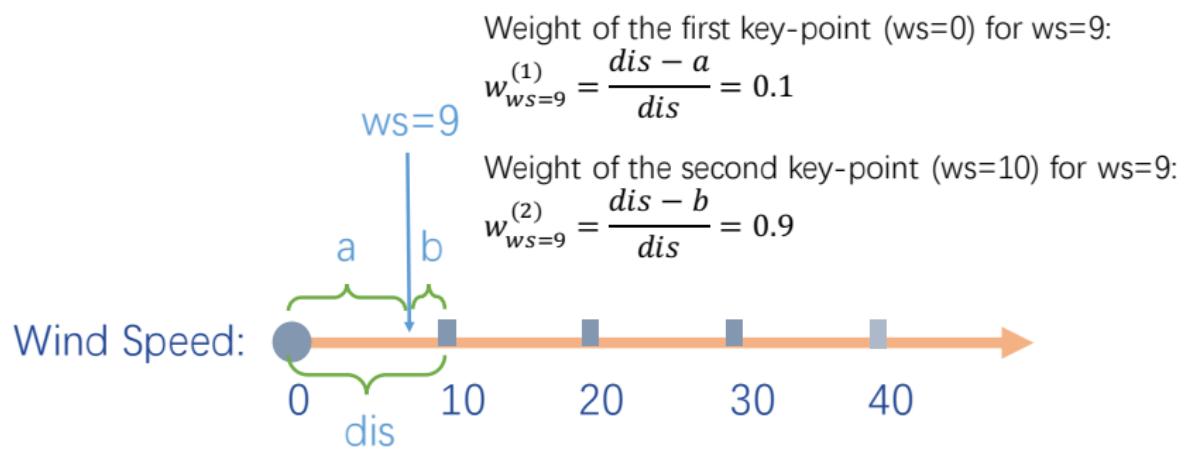
We represent each attribute (e.g. wind speed) as a combination of some weighted **key-points**.

- The key-points are derived by:
 - ◊ Taking the min and max values of the attribute (from training data)
 - ◊ Key-points are evenly spaced between the min and max values
- The number of the key-points for an attribute are fixed



Data Representation

An example of deriving the key point weights for the attribute value ws=9 (i.e., wind speed dimension).



Data Representation

An example of deriving the key point weights for the data record $\text{ws}=9$ (i.e., wind speed dimension).

- In this way, an attribute value (e.g. $\text{ws}=9$) can be represented by a key point weight vector
- I.e. the weight vector of $\text{ws}=9$ is [0.1, 0.9, 0, 0, 0].



Representing Data in Vector

Similarly, a data record (i.e., a set of attribute-value pairs) can be represented by multiple groups of key-points, e.g.:

$$ws = 9 \rightarrow [0.1, 0.9, 0, 0, 0]$$

$$dir = 2 \rightarrow [0.97, 0.03, 0, 0, 0]$$

Thus, to represent a set of attribute-value pairs, we concatenate the individual weight vectors, e.g.:

$$\{ws = 9, dir = 2\} \rightarrow [0.1, 0.9, 0, 0, 0, 0.97, 0.03, 0, 0, 0]$$

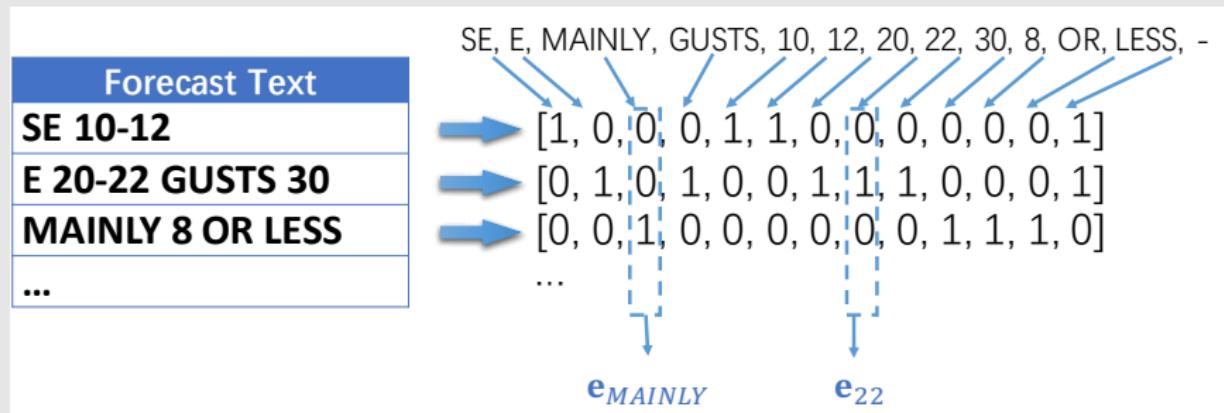
Representing Data in Vector

The entire data-text corpus can then be represented with a vector matrix (K), whose row corresponds to the weight vector of a data record.

$$\begin{array}{l} \text{Wind speed} \qquad \qquad \qquad \text{Wind direction} \\ \overbrace{\qquad\qquad\qquad}^{\text{Wind speed}} \qquad \qquad \qquad \overbrace{\qquad\qquad\qquad}^{\text{Wind direction}} \\ \{ws = 9, dir = 2, \dots\} \quad \{ws = 20, dir = 130, \dots\} \rightarrow K = \begin{bmatrix} 0.1 & 0.9 & 0 & 0 & 0 & 0.97 & 0.03 & 0 & 0 & 0 & \dots \\ 0 & 0 & 1 & 0 & 0 & 0 & 0.66 & 0.44 & 0 & 0 & \dots \\ 0.8 & 0.2 & 0 & 0 & 0 & 1 & 0.86 & 0.14 & 0 & 0 & \dots \\ \dots & \dots \end{bmatrix} \\ \{ws = 2, dir = 90, \dots\} \\ \dots \end{array}$$

Representing Text

- We use a column vector (namely e_i) to represent the text of a data record. Each element of e_i indicates whether a word appears in the data record, e.g.:



Representing Words in Vector

- So far, data are represented by weight vectors, whose values can be calculated using key points.
- We represent words in the corpus using the same weight vectors whose values are unknown.

$$\{ws = 9, dir = 2\} \rightarrow v_{data} = [0.1, 0.9, 0, 0, 0, 0.97, 0.03, 0, 0, 0]$$

$$"muggy" \rightarrow v_{muggy} = [?, ?, ?, ?, ?, ?, ?, ?, ?, ?]$$

Methodology

Task: To estimate v_i for word i given a data-to-text corpus as input.

Assumption: v_i and v_d should be close to each other in the vector space if word i appears in data record d

$$\frac{v_{d1} \cdot v_i}{\|v_{d1}\| \|v_i\|} = appear(i, d1)$$

$$\frac{v_{d2} \cdot v_i}{\|v_{d2}\| \|v_i\|} = appear(i, d2)$$

...

NB: $appear(i, d1) = 1$ if word i appears in data record d and 0 otherwise.

Methodology

Our task is to find the weight vector v_i for each word i , such that the similarity of v_i and v_d is close to $appear(i, d)$ as much as possible for each data record (d).

$$v_i = \min_{v_i} \sqrt{\sum_d (sim(v_i, v_d) - appear(i, d))^2}$$

Methodology

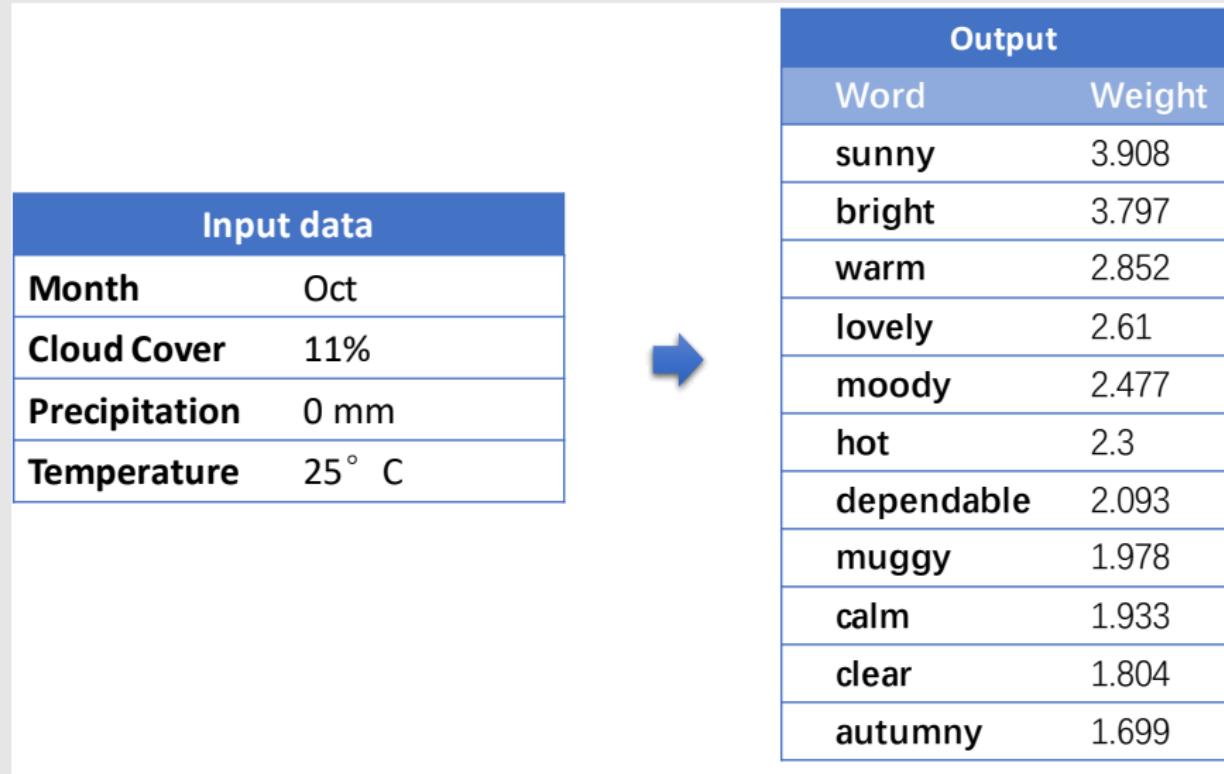
- Finding v_i equivalent to finding the optimal solution the following equation using least squares

$$\mathbf{K}' \cdot \frac{v_i}{\|v_i\|} = \mathbf{e}_i$$

$$opt\left(\frac{v_i}{\|v_i\|}\right) = (\mathbf{K}'^T \mathbf{K}')^{-1} \mathbf{K}'^T \mathbf{e}_i$$

- Once v_i is solved, we can then estimate the most appropriate words for for unseen data .

Results



- Which phrase should be used to identify an object?
- Referring expression generation: the task of selecting the content (and, to some extent, the form) of referential noun phrases in text.
 - ◊ Look at the big dog
 - ◊ Look at Fido
 - ◊ Look at it

Types of reference

- Pronoun – it, them, him, you,...
- Name – Dr Adam Smith, Adam Smith, Adam, Dr Smith
- Definite NP – the big black dog, the big dog, the black dog, the dog

Suggestion

- Use pronoun if possible
 - ◊ Referent mentioned recently
 - ◊ Pronoun is not ambiguous
- Else use name if possible
 - ◊ Shortest form which is unambiguous and stylistically allowed
- Else use definite NP
 - ◊ Shortest one, prefer basic-level words
- Only use forms seen in corpus

Aggregation

- Aggregation: the task of merging distinct representations into a single, more concise representation
- When/how should we combine phrases?
 - ◊ Your first ascent was fine. Your second ascent was fine.
 - ◊ Your first ascent was fine, and your second ascent was fine.
 - ◊ Your first ascent and your second ascent were fine.
 - ◊ Your first and second ascents were fine.

Suggestions on Aggregation

- Generally use the deepest one we can
 - ◊ Your first ascent was safe, and your second ascent was safe.
 - ◊ Your first ascent and your second ascent were safe.
 - ◊ Your first and second ascents were safe.
- Depends on how similar phrases are.
- Depends on genre (corpus)

Microplannng

- Decide how to best express a message in language
 - ◊ Essential for producing “nice” texts
- Imitating corpus works to some degree, but not perfectly
 - ◊ Currently more of an art than a science
- Key is better understanding of how linguistic choices affected readers
 - ◊ Our SumTime weather-forecast generator microplans better than human forecasters

Realisation

- Third (last) NLG stage
- Creating linear text from (typically) structured input; ensuring syntactic correctness
- Take care of details of language
 - ◊ Syntactic details
 - Eg Agreement (the dog runs vs the dogs run)
 - ◊ Morphological details
 - Eg, plurals (dog/dogs vs box/boxes)
 - ◊ Presentation details
 - Eg, fit to 80 column width

Realisation

- Problem: There are lots of finicky details of language which most people developing NLG systems don't want to worry about
- Solution: Automate this using a realiser

Syntax

- Sentences must obey the rules of English grammar
 - ◊ Specifies which order words should appear in, extra function words, word forms
- Many aspects of grammar are somewhat bizarre
- Just tell realiser verb, tense, whether negated, and it will figure out the verb group
 - ◊ (watch, future) -> will watch
 - ◊ (watch, past, negated) -> did not watch
 - ◊ Etc
- Similarly automate other “obscure” encodings of information

Morphology

In linguistics, morphology is the study of words, how they are formed, and their relationship to other words in the same language. E.g.,

- Variations of a root form of a word, e.g., prefixes, suffixes
- Inflectional morphology - same core meaning
 - ◊ plurals, past tense, superlatives, e.g., dog, dogs
 - ◊ part of speech unchanged
- Derivational morphology - change meaning
 - ◊ prefix *re* means do again: reheat, resist
 - ◊ suffix *er* means one who: teacher, baker
 - ◊ part of speech changed

- Calculates morphological variants automatically
 - ◊ (dog, plural) -> dogs
 - ◊ (box, plural) -> boxes
 - ◊ (child, plural) -> children
 - ◊ etc
- Automatically insert appropriate punctuation for a structure
- Many possible output formats
 - ◊ Simple text
 - ◊ HTML
 - ◊ MS Word

Realiser systems

- simpleNLG – relatively limited functionality, but well documented, fast, easy to use, tested
 - ◊ Most popular, easy-to-use, programmatically controllable and extendable realisation engine.
 - ◊ Has adapted into many (western) languages: French, German, Mandarin ...
- KPML – lots of functionality but poorly documented, buggy, slow
- openCCG – somewhere in between
- Many more

Realiser

- creates linear text from (typically) structured input; ensuring syntactic correctness
- automates the finicky details of language
 - ◊ So NLG developer doesn't have to worry about these
 - ◊ One of the advantages of NLG

COM6115: Text Processing

Introduction to Information Extraction

Chenghua Lin

Department of Computer Science
University of Sheffield

Overview of Lectures on IE

- Introduction to Information Extraction
 - ◊ Definition + Contrast with IR
 - ◊ Example Applications
 - ◊ Overview of Tasks and Approaches
 - ◊ Evaluation
 - ◊ A Brief History of IE
- Named Entity Recognition
 - ◊ Task
 - ◊ Approaches: Rule-based; Supervised Learning
 - ◊ Entity Linking
- Relation Extraction
 - ◊ Task
 - ◊ Approaches: Rule-based; Supervised learning; Bootstrapping; Distant Supervision

Introduction to Information Extraction: Outline

- Definition + Contrast with IR
- Example Applications
- Overview of Tasks
- Overview of Approaches
- Evaluation
- A Brief History of IE

- Definition: the **Information Extraction** (IE) task:

From each text in a set of unstructured natural language texts identify information about predefined classes of **entities**, **relationships** or **events** and record this information in a structured form by either:

- ◊ Annotating the source text, e.g. using XML tags; or
 - ◊ Filling in a data structure separate from the text, e.g a template or database record or “stand-off annotation”
- For example: from financial newswire stories identify those dealing with management succession events and from these extract details of organisations and persons, the post being assumed or vacated, etc.

Definition (cont)

- IE may also be described as:
 - ◊ The activity of populating a structured information repository (database) from an unstructured, or free text, information source
 - ◊ The activity of creating a semantically annotated text collection (cf. "The Semantic Web")
- The resulting structured data source is then used for some other purpose:
 - ◊ searching or analysis using conventional database queries;
 - ◊ data-mining;
 - ◊ generating a summary (perhaps in another language);

Example

Who's News: @ Burns Fry Ltd.

04/13/94

WALL STREET JOURNAL (J), PAGE B10 <CO>

BURNS FRY Ltd. (Toronto) – Donald Wright, 46 years old, was named executive vice president and director of fixed income at this brokerage firm. Mr. Wright resigned as president of Merrill Lynch Canada Inc., a unit of Merrill Lynch & Co., to succeed Mark Kassirer, 48, who left Burns Fry last month. A Merrill Lynch spokeswoman said it hasn't named a successor to Mr. Wright, who is expected to begin his new position by the end of the month.

Example

Who's News: @ Burns Fry Ltd.

04/13/94

WALL STREET JOURNAL (J), PAGE B10 <CO>

BURNS FRY Ltd. (Toronto) – Donald Wright, 46 years old, was named executive vice president and director of fixed income at this brokerage firm.

Mr. Wright resigned as president of Merrill Lynch Canada Inc., a unit of Merrill Lynch & Co., to succeed Mark Kassirer, 48, who left Burns Fry last month. A Merrill Lynch spokeswoman said it hasn't named a successor to Mr. Wright, who is expected to begin his new position by the end of the month.

- identify persons (red)

Example

Who's News: © Burns Fry Ltd.

04/13/94

WALL STREET JOURNAL (J), PAGE B10 (CO)

BURNS FRY Ltd. (Toronto) – Donald Wright, 46 years old, was named executive vice president and director of fixed income at this brokerage firm.

Mr. Wright resigned as president of Merrill Lynch Canada Inc., a unit of Merrill Lynch & Co., to succeed Mark Kassirer, 48, who left Burns Fry last month. A Merrill Lynch spokeswoman said it hasn't named a successor to Mr. Wright, who is expected to begin his new position by the end of the month.

- identify persons (red)
- identify organisations (blue)

Example

Who's News: @ Burns Fry Ltd.

04/13/94

WALL STREET JOURNAL (J), PAGE B10 <CO>

BURNS FRY Ltd. (**Toronto**) – **Donald Wright**, 46 years old, was named executive vice president and director of fixed income at this brokerage firm.

Mr. Wright resigned as president of **Merrill Lynch Canada Inc.**, a unit of **Merrill Lynch & Co.**, to succeed **Mark Kassirer**, 48, who left **Burns Fry** last month. A **Merrill Lynch** spokeswoman said it hasn't named a successor to **Mr. Wright**, who is expected to begin his new position by the end of the month.

- identify persons (red)
- identify organisations (blue)
- identify locations (green)

Example

Who's News: @ Burns Fry Ltd.

04/13/94

WALL STREET JOURNAL (J), PAGE B10 <CO>

BURNS FRY Ltd. (**Toronto**) – **Donald Wright**, 46 years old, was named executive vice president and director of fixed income at this brokerage firm.

Mr. Wright resigned as president of **Merrill Lynch Canada Inc.**, a unit of **Merrill Lynch & Co.**, to succeed **Mark Kassirer**, 48, who left Burns Fry **last month**. A **Merrill Lynch** spokeswoman said it hasn't named a successor to **Mr. Wright**, who is expected to begin his new position by **the end of the month**.

- identify persons (red)
- identify organisations (blue)
- identify locations (green)
- identify times (cyan)

Example

Who's News: @ Burns Fry Ltd.

04/13/94

WALL STREET JOURNAL (J), PAGE B10 <CO>

BURNS FRY Ltd. (Toronto) – Donald Wright, 46 years old, was named executive vice president and director of fixed income at this brokerage firm.

Mr. Wright resigned as president of Merrill Lynch Canada Inc., a unit of Merrill Lynch & Co., to succeed Mark Kassirer, 48, who left Burns Fry last month. A Merrill Lynch spokeswoman said it hasn't named a successor to Mr. Wright, who is expected to begin his new position by the end of the month.

- identify persons (red)
- identify organisations (blue)
- identify locations (green)
- identify times (cyan)
- identify company positions (purple)

Example

Who's News: @ Burns Fry Ltd.

04/13/94

WALL STREET JOURNAL (J), PAGE B10 <CO>

BURNS FRY Ltd. (Toronto) – Donald Wright, 46 years old, was named executive vice president and director of fixed income at this brokerage firm.

Mr. Wright resigned as president of Merrill Lynch Canada Inc., a unit of Merrill Lynch & Co., to succeed Mark Kassirer, 48, who left Burns Fry last month. A Merrill Lynch spokeswoman said it hasn't named a successor to Mr. Wright, who is expected to begin his new position by the end of the month.

- identify persons (red)
- identify organisations (blue)
- identify locations (green)
- identify times (cyan)
- identify company positions (purple)
- identify succession events (underlined)

Contrast with Information Retrieval

Information Retrieval

- Task:
 - ◊ Given: a document collection and a user query
 - ◊ Return: a (ranked) list of documents relevant to the user query
- Strengths:
 - ◊ Can search huge document collections very rapidly
 - ◊ Insensitive to genre and domain of the texts
 - ◊ Relatively straightforward to implement
 - challenges scaling to huge, dynamic document collections, e.g. the Web
- Weaknesses
 - ◊ Documents are returned not information/answers, so
 - user must further read texts to extract information
 - output is unstructured so limited possibilities for direct data mining/further processing

Contrast with Information Retrieval

Information Extraction

- Task:
 - ◊ Given: a document collection and a predefined set of entities, relations and/or events
 - ◊ Return: a structured representation of all mentions of the specified entities, relations and/or events
- Strengths:
 - ◊ Extracts facts from texts, not just texts from text collections
 - ◊ Can feed other powerful applications (databases, semantic indexing engines, data mining tools)
- Weaknesses
 - ◊ Systems tend to be genre/domain specific and porting to new genres and domains can be time-consuming/requires expertise
 - ◊ Limited accuracy
 - ◊ Computationally demanding, so performance issues on very large collections

Example Applications

- Scrapping web pages to build structured databases of job postings, apartment rentals, seminar announcements, etc.
- Assisting biomedical database curators by extracting biomedical entities and relations from the scientific literature prior to entry in a human-maintained database (e.g. Flybase)
- Assisting companies in competitor intelligence gathering, e.g. management or researcher succession events, new product or project announcements, etc.

Introduction to Information Extraction: Outline

- Definition + Contrast with IR
- Example Applications
- Overview of Tasks
- Overview of Approaches
- Evaluation
- A Brief History of IE

Overview of Tasks: Entity Extraction

Entity Extraction/Named Entity Recognition

- **Task:** for each textual mention of an entity of one of a fixed set of types identify its **extent** and its **type**

Cable and Wireless today announced ... Extent: 0-3; Type = ORG

IBM and Microsoft today announced ... Extent: 0-1; Type = ORG
Extent: 2-3 Type = ORG

John Lewis hired ... Extent: 0-2; Type = ORG

Theresa May hired ... Extent: 0-2; Type = PER

- Types of entities which have been addressed by IE systems include:
 - ◊ Named individuals
 - Organisations, persons, locations, books, films, ships, restaurants ...
 - ◊ Named Kinds
 - Proteins, chemical compounds/drugs, diseases, aircraft components ...
 - ◊ Times
 - temporal expressions – dates, times of day
 - ◊ Measures
 - monetary expressions, distances/sizes, weights ...

Overview of Tasks: Entity Extraction – Coreference

- Multiple references to the same entity in a text are rarely made using the same string:
 - ◊ Pronouns – Tony Blair ... he
 - ◊ Names/definite descriptions – Tony Blair ... the Prime Minister
 - ◊ Abbreviated forms – Theresa May ... May; United Nations ... UN
 - ◊ Orthographic variants – alpha helix ... alpha-helix ... α -helix ... a-helix
- Different textual expressions that refer to the same real world entity are said to **corefer**.
- Clearly IE systems are more useful if they can recognise which text mentions are coreferential.
- **Coreference Task:** link together all textual references to the same real world entity, regardless of whether the surface form is a name or not

Overview of Tasks: Relation Extraction

Relation Extraction

- **Task:** identify all assertions of relations, usually binary, between entities identified in entity extraction
- May be divided into two subtasks:
 - ◊ **Relation detection:** find pairs of entities between which a relation holds
 - ◊ **Relation classification:** for pairs of entities between which a relation holds, determine what the relation is
- Examples
 - ◊ LOCATION_OF holding between
 - ORGANISATION and GEOPOLITICAL_LOCATION
 - medical INVESTIGATION and BODY_PART
 - ◊ EMPLOYEE_OF holding between PERSON and ORGANISATION
 - ◊ PRODUCT_OF holding between ARTIFACT and ORGANISATION
 - ◊ INTERACTION holding between PROTEIN and PROTEIN

Overview of Tasks: Relation Extraction

Relation Extraction is challenging for several reasons:

- The same relation may be expressed in many different ways:
 - ◊ Synonyms: [Microsoft]_{ORG} is based/headquartered in [Redmond]_{LOC}
 - ◊ Syntactic variations:
 - [Microsoft]_{ORG}, the software giant and . . . , is based in [Redmond]_{LOC}
 - [Redmond]_{LOC-based} [Microsoft]_{ORG} . . .
 - [Redmond]_{LOC}'s [Microsoft]_{ORG} . . . ; [Microsoft]_{ORG} of [Redmond]_{LOC}
 - [Redmond]_{LOC} software giant [Microsoft]_{ORG} . . .
- Discovering relations frequently depends upon being able to follow coreference links.

Dirk Ruthless of MegaCorp made a stunning announcement today. In September he will be stepping down as Chief Executive Officer to spend more time with his pet piranhas.

To determine the corporate position of Dirk Ruthless we must correctly resolve the pronominal anaphor “he” in the second sentence with “Dirk Ruthless” in the first

Overview of Tasks: Event Detection

Event Extraction

- **Task:** identify all reports of event instances, typically of a small set of classes
- May be divided into two subtasks:
 - ◊ **Event detection:** find mentions of events in text
 - ◊ **Event classification:** assign detected events to one of a set of classes
- Examples
 - ◊ Rocket/missile launches
 - ◊ Management succession events
 - ◊ Joint venture/product announcements
 - ◊ Terrorist attacks
- Events may be simply viewed as relations. However they are typically complex relations that
 - ◊ Are temporally situated and often of relatively short duration
 - ◊ Involve multiple role players (frequently > 2)
 - ◊ Are often expressed across multiple sentences

Introduction to Information Extraction: Outline

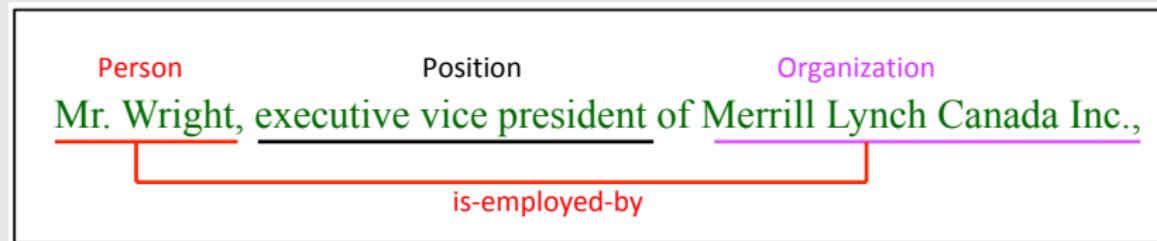
- Definition + Contrast with IR
- Example Applications
- Overview of Tasks
- **Overview of Approaches**
- Evaluation
- A Brief History of IE

Overview of Approaches:

Approaches to IE may be placed into four categories:

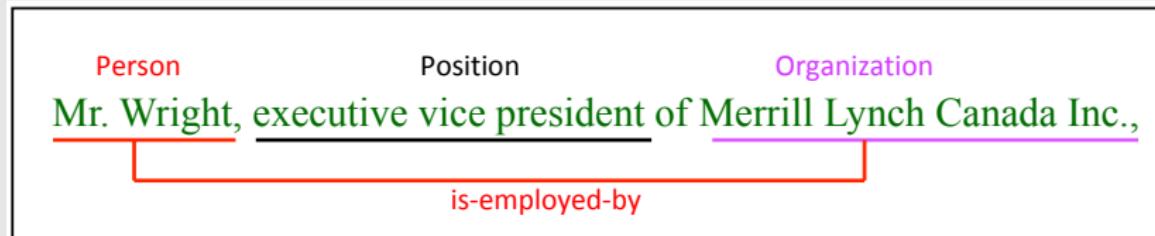
- Knowledge Engineering Approaches
- Supervised Learning Approaches
- Bootstrapping Approaches
- Distant Supervision Approaches

Knowledge Engineering Approaches



- Such systems use manually authored rules and can be divided into
 - “deep” – linguistically inspired “language understanding” systems
 - “shallow” – systems engineered to the IE task, typically using pattern-action rules
- Pattern: ‘‘Mr. \$Uppercase-initial-word’’
Action: add-entity(person(“Mr. \$Uppercase-initial-word”))
- Pattern: \\$Person, \$Position of \\$Organization”
Action: add-relation(is-employed-by(\$Person,\$Organization))

Supervised learning approaches



- Systems are given texts with manually annotated entities + relations
- For each entity/relation create a training instance
 - ◊ k words either side of an entity mention
 - ◊ k words to the left of entity 1 and to the right of entity 2 plus the words in between
- Training instances represented in terms of features
 - ◊ words, parts of speech, orthographic characteristics, syntactic info
- Systems may learn
 - ◊ patterns that match extraction targets
 - ◊ Classifiers that classify tokens as beginning/inside/outside a tag type
- Learning techniques include: covering algorithms, HMMs, SVMs

Bootstrapping Approaches

- A technique for relation extraction that requires only minimal supervision
- Systems are given
 - ◊ seed tuples (e.g. `< Microsoft, Redmond >`)
 - ◊ seed patterns (e.g. `[X]ORG is located in [Y]Loc`)or both.
- System searches in large corpus for
 - ◊ occurrences of seed tuples and then extracts a pattern that matches the context of the seed tuple
 - ◊ matches of seed patterns from which it harvests new tuples
- New tuples are assumed to stand in the required relation and are added to the tuple store
- Process iterates until convergence
- See later lecture

Distant Supervision Approaches

- Sometimes also called “weakly labelled” approaches
- Assumes a (semi-)structured data source, such as
 - ◊ Wikipedia infoboxes (e.g. PERSON BORN_IN LOCATION/DATE)
 - ◊ Freebase or Wikidata
 - ◊ Flybase or the Yeast Protein Database, (e.g. PROTEIN IS_LOCATED_IN SUBCELLULAR_LOCATION)

which contains tuples of entities standing in the relation of interest and, ideally, a pointer to a source text

- Tuples from data source are used to label
 - ◊ the text with which they are associated, if available
 - ◊ documents from the web, if not
- Labelled data is used to train a standard supervised named entity or relation extraction system
- See later lecture

Introduction to Information Extraction: Outline

- Definition + Contrast with IR
- Example Applications
- Overview of Tasks
- Overview of Approaches
- Evaluation
- A Brief History of IE

Evaluation

- Correct answers, called **keys**, are produced manually for each extraction task (filled templates or SGML annotated texts)
- Scoring of system results, called **responses**, against keys is done automatically.
- At least some portion of the answer keys are multiply produced by different humans so that **interannotator agreement** figures can be computed.
- Principal metrics – borrowed from information retrieval – are:
 - ◊ **Precision** (how much of what system returns is correct)
 - ◊ **Recall** (how much of what is correct system returns)
 - ◊ **F-measure** (a weighted combination of precision and recall)

Introduction to Information Extraction: Outline

- Definition + Contrast with IR
- Example Applications
- Overview of Tasks
- Overview of Approaches
- Evaluation + Shared Task Challenges
- A Brief History of IE

A Brief History of IE

- 1960s** The first published work on information extraction (though it was not called this at the time)
- 1970s** A significant precursor was the psychologist Roger Schank's work on scripts and story understanding
- 1980s** Saw the emergence of some commercial systems targetted at financial transactions and newswires
Message Understanding Conference 1 (MUC-1) – in 1987
- 1990s** MUC ran 7 times until 1998 and significantly advanced the field.
Machine learning approaches to IE began to appear
- 2000s** ACE (Automatic Content Extraction) the successor programme to MUC ran 1999-2008; succeeded by TAC (Text Analytics Conference) (2008-present); BioCreative (IE in the biomedical domain) began (2004-present); work on IE in other languages began (e.g. Spanish, Japanese, Chinese, Arabic)
- 2010s** TAC is going, particularly the **knowledge base population** track
Currently there are a number of IE systems on the market and a large and on-going research effort in the field

COM6115: Text Processing

Information Extraction: Named Entity Recognition

Chenghua Lin

Department of Computer Science
University of Sheffield

Overview of Lectures on IE

- Introduction to Information Extraction
 - ◊ Definition + Contrast with IR
 - ◊ Example Applications
 - ◊ Overview of Tasks and Approaches
 - ◊ Evaluation + Shared Task Challenges
 - ◊ A Brief History of IE
- Named Entity Recognition
 - ◊ Task
 - ◊ Approaches to NER
 - ◊ Entity Linking
- Relation Extraction
 - ◊ Task
 - ◊ Approaches: Rule-based; Supervised learning; Bootstrapping; Distant Supervision

Named Entity Recognition: Outline

- Named Entity Recognition Task
- Approaches to NER
 - ◊ Knowledge-engineering approaches to NER
 - ◊ Supervised learning approaches to NER
- Entity Linking

Named Entity Recognition Task: Recap

- **Task:** for each textual mention of an entity of one of a fixed set of types identify its **extent** and its **type**

Cable and Wireless today announced ... Extent: 0-3; Type = ORG

IBM and Microsoft today announced ... Extent: 0-1; Type = ORG
Extent: 2-3 Type = ORG

John Lewis hired ... Extent: 0-2; Type = ORG

Theresa May hired ... Extent: 0-2; Type = PER

- Types of entities which have been addressed by IE systems include:

- ◊ Named individuals
 - Organisations, persons, locations, books, films, ships, restaurants ...
- ◊ Named Kinds
 - Proteins, chemical compounds/drugs, diseases, aircraft components ...
- ◊ Times
 - temporal expressions – dates, times of day
- ◊ Measures
 - monetary expressions, distances/sizes, weights ...

Entity Extraction – Coreference: Recap

- Multiple references to the same entity in a text are rarely made using the same string:
 - ◊ Pronouns – They ... he
 - ◊ Names/definite descriptions – Tony Blair ... the Prime Minister
 - ◊ Abbreviated forms – Theresa May ... May; United Nations ... UN
 - ◊ Orthographic variants – alpha helix ... alpha-helix ... α -helix ... a-helix
- Different textual expressions that refer to the same real world entity are said to **corefer**.
- Clearly IE systems are more useful if they can recognise which text mentions are coreferential.
- **Coreference Task:** link together all textual references to the same real world entity, regardless of whether the surface form is a name or not

Named Entity Recognition: Outline

- Named Entity Recognition Task
- **Approaches**
 - ◊ Knowledge-engineering approaches to NER
 - ◊ Supervised learning approaches to NER
- Entity Linking

Overview of Approaches to NER

As with IE in general approaches to NER may be placed into four categories:

- ◊ Knowledge Engineering Approaches
- ◊ Supervised Learning Approaches
- ◊ Bootstrapping Approaches
- ◊ Distant Supervision Approaches

For reasons of time, we will consider the first two only.

Knowledge Engineering Approaches to NER

- Such systems typically use
 - ◊ named entity lexicons and
 - ◊ manually authored pattern/action rules or regular expression
- Dominant approach in the 1990s and still in use in many IE systems today.
- One such NER system, developed for participation in MUC-6, is described in Wakao et al. (1996) – will use as an example.
- The Wakao et al. system recognizes organisation, person and location names and time expressions in newswire texts
- System has three main stages:
 - ◊ Lexical processing
 - ◊ NE parsing
 - ◊ Discourse interpretation

Knowledge Engineering Approaches to NER

Step 1: Lexical Processing

- Many rule-based NER systems made extensive use of specialised lexicons of proper names, such as **gazetteers** – lists of place names
- The Wakao et al. system has specialised lexicons for
 - ◊ Organisations (2600 entries)
 - ◊ Locations (2200 entries)
 - ◊ Person names (500 entries)
 - ◊ Company designators (e.g. Plc, Corp, Ltd – 94 entries)
 - ◊ Person titles (e.g. Mr, Dr, Reverend – 160 titles)
- Why not use even larger gazetteers?
 - ◊ e.g. Gazetteer of British Place Names claims it “provides an exhaustive Place Name Index to Great Britain, containing over 50,000 entries”
- Reasons:
 - ◊ Many NEs occur in multiple categories – the larger the lexicons the greater ambiguity, e.g.,
 - Ford – company vs Ford – person vs Ford – place
 - ◊ the listing of names is never complete, so need some mechanism to type unseen NEs in any case

Knowledge Engineering Approaches to NER

Step 1: Lexical Processing (cont)

Principal lexical processing sub-steps in the Wakao et al. system are:

- Tokenisation, sentence splitting, morphological analysis
- Part-of-speech tagging – tags known proper name words and unknown uppercase-initial words as proper names (NNP, NNPS)
- Name List/Gazetteer Lookup and Tagging (organisations, locations, persons, company designators, person titles)
- Trigger Word Tagging – certain words in multi-word names function as trigger words, permitting classification of the name
 - ◊ e.g. *Airlines* in *Wing and Prayer Airlines*
 - ◊ system has trigger words for various orgs, gov't institutions, locations

Example:

Norwich Investment Bank plc. today announced ... →

Norwich_{NNP/LOC} Investment_{NNP} Bank_{NNP/ORG-TRIGGER} plc._{NN/CDG}
today_{RB} announced_{VBD} ...

Knowledge Engineering Approaches to NER

Step 2: NE Parsing

- After lexical processing the next step in the Wakao et al. system is NE parsing.
- The system has 177 hand-produced rules for proper names: 94 for organisation; 54 for person; 11 for location; 18 for time expressions.
- The rule `ORGAN NP --> NAMES_NP '&' NAMES_NP` means:
If an unclassified proper name (`NAMES_NP`) is followed by '`&`' and another unclassified proper name, then it is an organisation name.
E.g. [Marks & Spencer](#) and [American Telephone & Telegraph](#)

Knowledge Engineering Approaches to NER

Step 3: Discourse Interpretation – Coreference Resolution

- When the name class of an antecedent (anaphor) is known then establishing coreference allows the name class of the anaphor (antecedent) to be established.
 - ◊ E.g., Alice is my sister. She is CEO of MS. (Antecedent – Alice; anaphor – she)
- An unclassified PN may be co-referential with a variant form of a classified PN, e.g.
 - ◊ Ford – Ford Motor Co.
 - ◊ CAA – Creative Artists Agency

In such cases the unclassified PN may be inferred to have the same class as the classified PN.

Wakao et al. use 45 heuristics of this type for organisation, location, and person names.

Knowledge Engineering Approaches to NER: Evaluation of Wakao et al.

- Evaluated on MUC-6 NE evaluation set – a blind test set of 30 Wall Street Journal Articles containing:
 - ◊ 449 organisation names
 - ◊ 373 person names
 - ◊ 110 location names
 - ◊ 111 time expressions
- Results were:

Proper Name Class	Recall	Precision
Organisation	91%	91%
Person	90%	95%
Location	88%	89%
Time	94%	97%
Overall	91%	93%

- Best system results on this evaluation were F-measure = 96.42%
 - ◊ Human results were 96.68%

Knowledge Engineering Approaches to NER: Strengths and Weaknesses

Strengths

- High performance – only several points behind human annotators
- Transparent – easy to understand what system is doing/why

Weaknesses

- Porting to another domain requires substantial rule re-engineering
- Acquisition of domain-specific lexicons
- Rule writing requires high levels of expertise

Named Entity Recognition: Outline

- Named Entity Recognition Task
- Approaches
 - ◊ Knowledge-engineering approaches to NER
 - ◊ Supervised learning approaches to NER
- Entity Linking

Supervised learning approaches to NER

- Supervised learning approaches aim to address the portability problems inherent in knowledge engineering NER
 - ◊ Instead of manually authoring rules, systems learn from annotated examples
 - ◊ Moving to new domain requires only annotated data in the domain – can be supplied by domain expert without need for expert computational linguist
- A wide variety of supervised learning techniques have been tried, including
 - ◊ Hidden Markov models
 - ◊ Decision Trees
 - ◊ Maximum Entropy
 - ◊ Support Vector Machines
 - ◊ Conditional Random Fields
 - ◊ AdaBoost
 - ◊ Deep Learning

Supervised learning approaches to NER: Sequence Labelling

- Systems may learn
 - ◊ **patterns** that match extraction targets
 - ◊ **classifiers** that label tokens as beginning/inside/outside a tag type
- Most work in recent years has followed the latter approach – called **sequence labelling**.
- In sequence labelling for NER, each token is given one of three label types:
 - ◊ **B_{Type}** if the token is at the **beginning** of a named entity of type = *Type* (here, e.g., $Type \in \{ORG, PER, LOC\}$).
 - ◊ **I_{Type}** if the token is **inside** a named entity of type = *Type*
 - ◊ **O** if the token is **outside** any named entity

For obvious reasons this scheme is called BIO or sometimes IOB sequence labelling

Supervised learning approaches to NER: Sequence Labelling – Example

- Suppose we have the sentence

[_{ORG} American Airlines], a unit of [_{ORG} AMR Corp.], immediately matched the move, spokesman [_{PER} Tim Wagner] said.

(Jurafsky and Martin, 2nd ed., p. 730)

- In BIO encoding this example looks like this →
- Given labelled sequences like this example sentence as training data, the task of the supervised learner to learn to predict the labelling of a new, unlabelled example.

Words	Label
American	B _{ORG}
Airlines	I _{ORG}
,	O
a	O
unit	O
of	O
AMR	B _{ORG}
Corp.	I _{ORG}
,	O
immediately	O
matched	O
the	O
move	O
,	O
spokesman	O
Tim	B _{PERS}
Wagner	I _{PERS}
said	O
.	O

Supervised learning approaches to NER: Features for Sequence Labelling

- Given a BIO-type encoding, each training instance (token) is typically represented as a set of **features**.
- Features can be not only characteristics of the token itself but of neighbouring tokens as well
 - ◊ usually consider tokens in a window of e.g. ± 2 or 3 tokens either side of the training instance

Supervised learning approaches to NER: Features for Sequence Labelling

- Given a BIO-type encoding, each training instance (token) is typically represented as a set of **features**.
- Features can be not only characteristics of the token itself but of neighbouring tokens as well
 - usually consider tokens in a window of e.g. ± 2 or 3 tokens either side of the training instance
- Features commonly used for NER sequence labelling include:

Feature	Explanation
Lexical items	The token to be labeled
Stemmed lexical items	Stemmed version of the target token
Shape	The orthographic pattern of the target word
Character affixes	Character-level affixes of the target and surrounding words
Part of speech	Part of speech of the word
Syntactic chunk labels	Base-phrase chunk label
Gazetteer or name list	Presence of the word in one or more named entity lists
Predictive token(s)	Presence of predictive words in surrounding text
Bag of words/Bag of N-grams	Words and/or N -grams occurring in the surrounding context

(Jurafsky and Martin, 2nd ed., p. 731)

Supervised learning approaches to NER: Features for Sequence Labelling (cont)

- For case sensitive languages like English the orthographic pattern of a token carries significant information.
- Commonly used “shape” features include:

Shape	Example
Lower	cummings
Capitalized	Washington
All caps	IRA
Mixed case	eBay
Capitalized character with period	H.
Ends in digit	A9
Contains hyphen	H-P

(Jurafsky and Martin, 2nd ed., p. 731)

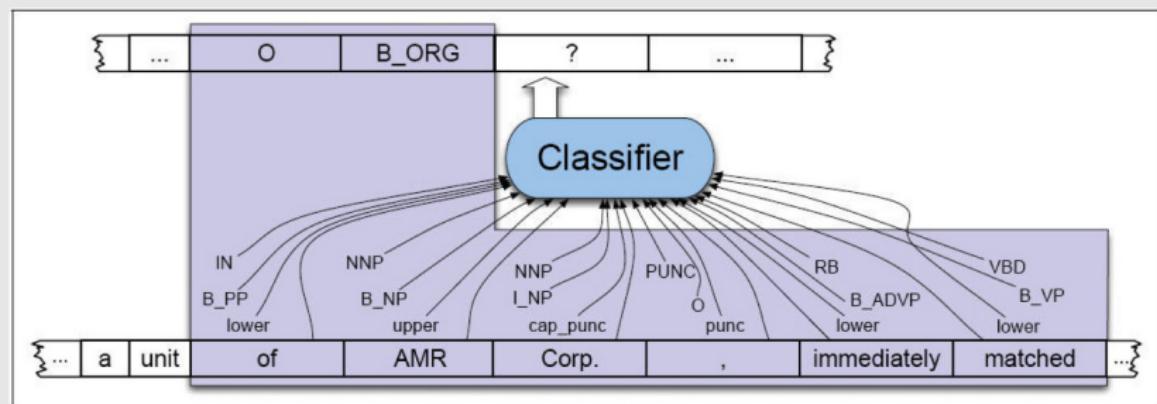
Supervised learning approaches to NER: Features for Sequence Labelling (cont)

- After a model has been learned, then at classification time the classifier extracts features from
 - ◊ the input string
 - ◊ its left predictions

Supervised learning approaches to NER: Features for Sequence Labelling (cont)

- After a model has been learned, then at classification time the classifier extracts features from
 - the input string
 - its left predictions

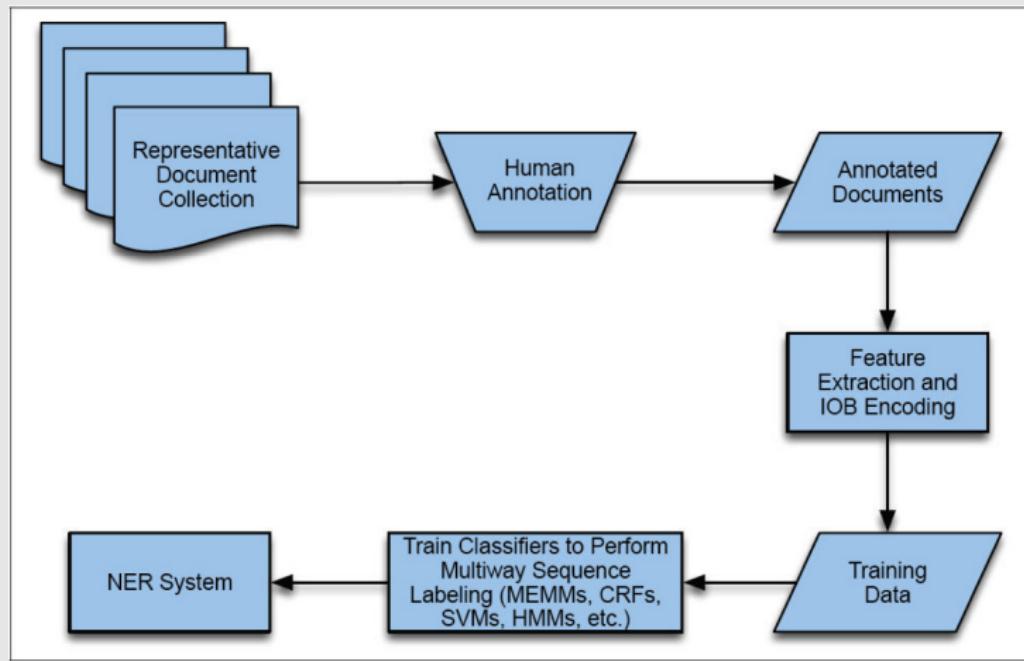
The available features for classification are those shown in the shaded area in the following figure:



(Jurafsky and Martin, 2nd ed., p. 733)

Supervised learning approaches to NER: Sequence Labelling Overview

- The following diagram recaps the main steps in the sequence labelling approach to NER.



(Jurafsky and Martin, 2nd ed., p. 722)

Supervised learning approaches to NER: Carreras et al. (2003)

- One implementation of the BIO-based sequence labelling for NER is described by Carreras et al. (2003)
 - ◊ Achieved highest score in the CONLL 2003 NER shared task challenge.
- Notable aspects of their approach include:
 - ◊ They divided the problem into two parts
 - **NE detection**: in a first pass over the text BIO tags are assigned without regard to type – i.e. boundaries are found for all NE's regardless of whether they are organisations, persons, locations, etc.
 - **NE classification**: in a second pass the NE's detected in the first pass are assigned a class (organisation, person, location, etc.)
 - Two pass approach has the advantage that training data for **all** NE classes can be used for the NE detection task
 - ◊ They used the **Adaboost** classifier
 - ◊ They used all features mentioned above plus some additional ones, e.g.
 - Type pattern of consecutive words in context – functional (f), capitalized (C), lowercased (l), punctuation mark (.), quote (‘), quote (’), other (x) – e.g. word type pattern for the phrase **John Smith payed 3 euros** is CClxl.

Supervised learning approaches to NER: Carreras et al. (2003)

- Overall performance on NE Detection:
 - ◊ 91.93% precision/94.02% recall on English test set
 - ◊ 85.85% precision/72.61% recall on German test set
 - Note: all common nouns are capitalised in German
- Overall best performance on NE Classification, assuming perfect Detection:
 - ◊ 95.14% accuracy for English
 - ◊ 85.14% accuracy for German
- Overall performance for NE Detection + Classification:
 - ◊ 84.05% precision/85.96% recall on English test set
 - ◊ 75.47% precision/63.82% recall on German test set
- Looking at different entity classes, LOC and PER score consistently higher than ORG and MISC

Named Entity Recognition: Outline

- Named Entity Recognition Task
- Approaches
 - ◊ Knowledge-engineering approaches to NER
 - ◊ Supervised learning approaches to NER
- Entity Linking

Entity Linking

- One important application of IE is **knowlege base population (KBP)** – facts are gathered from open access web sources and used to build a structured information repository.
- For KBP to work, not only must entities be detected, they must be linked to the appropriate entry in the KB, if facts are to be correctly assembled.

Entity Linking

- One important application of IE is **knowlege base population (KBP)** – facts are gathered from open access web sources and used to build a structured information repository.
- For KBP to work, not only must entities be detected, they must be linked to the appropriate entry in the KB, if facts are to be correctly assembled.
- This leads to the **Entity Linking Task**: Given a text with a recognised NE mention in that text and a knowledge base (KB), such as Wikipedia, link the NEs to the matching entry in the KB if there is one, else create an entry.
- Is this task difficult? – yes!!
 - ◊ Wikipedia contains over 200 entries for **John Smith**
 - ◊ There are at least 1,716 places called **San José** (or **San Jose**); 41 **Springfield**'s in the US
 - ◊ **Ashoka Restaurant**, **ABC Taxis**, ...

Entity Linking (cont)

- Many approaches have been developed.
- Simple approach: given a text T containing an NE mention m and using Wikipedia as a KB
 - 1 index all pages in the KB using an information retrieval system
 - 2 build a query from T (e.g. use the sentence/paragraph/whole text) containing m and search the KB
 - 3 from the ranked list of KB pages returned by step 2 pick the high ranked page whose name matches m and return it

Problem: doesn't work very well

- More successful approaches consider disambiguating all NEs jointly
 - ◊ Intuition: in disambiguating a text mentioning **Ashoka** and **Sheffield**, the **Ashoka** mentioned is likely to be in **Sheffield**, while the **Sheffield** is likely to be one containing an **Ashoka** restaurant.
 - ◊ See, e.g., Alhelbawy and Gaizauskas (2014)

Conclusion

- Named Entity Recognition (NER) is a core IE technology that is now relatively mature and at “usable” performance levels
- NER aims to detect and classify all mentions of named entities of a given set of entity types within a given text
- Techniques used have included:
 - ◊ knowledge engineering approaches
 - ◊ supervised learning approaches

References

- Alhelbawy, A. and Gaizauskas, R. Collective Named Entity Disambiguation using Graph Ranking and Clique Partitioning Approaches. In Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, 1544–1555, 2014.
- Carreras, X., Marquez, L. and Padro, L. A Simple Named Entity Extractor using AdaBoost. Proceedings of CoNLL-2003 , 152–155, 2003.
- Jurafsky, D and Martin, J. H. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition. 2nd ed. Pearson Inc. 2009. See Chapter 22.1 “Named Entity Recognition”.

References

- Wakao, T., Gaizauskas, R. and Wilks, Y. Evaluation of an Algorithm for the Recognition and Classification of Proper Names. In Proceedings of the 16th International Conference on Computational Linguistics (COLING96), 418-423, 1996.

COM6115: Text Processing

Information Extraction: Relation Extraction

Chenghua Lin

Department of Computer Science
University of Sheffield

Overview of Lectures on IE

- Introduction to Information Extraction
 - ◊ Definition + Contrast with IR
 - ◊ Example Applications
 - ◊ Overview of Tasks and Approaches
 - ◊ Evaluation + Shared Task Challenges
 - ◊ A Brief History of IE
- Named Entity Recognition
 - ◊ Task
 - ◊ Approaches to NER
 - ◊ Entity Linking
- Relation Extraction
 - ◊ Task
 - ◊ Approaches: Knowledge Engineering; Supervised learning; Bootstrapping; Distant Supervision

Relation Extraction: Outline

- Relation Extraction Task
- Approaches to Relation Extraction
 - ◊ Knowledge-engineering approaches to NER
 - ◊ Supervised learning approaches to NER
 - ◊ Bootstrapping Approaches to NER
 - ◊ Distant Supervision Approaches to NER

Relation Extraction Task: Recap

- **Task:** given a text T and a set of relations \mathbf{R} , identify all assertions of relations from \mathbf{R} in T , holding between entities identified in entity extraction.
- Note:
 - ◊ relations in \mathbf{R} are usually binary
 - ◊ the entity types of arguments of relations in \mathbf{R} are assumed to be a subset of those identified in the entity extraction process
- May be divided into two subtasks:
 - ◊ **Relation detection:** find pairs of entities between which a relation holds
 - ◊ **Relation classification:** for pairs of entities between which a relation holds, determine what that relation is

Relation Extraction Task: Examples

- Examples

- ◊ LOCATION_OF holding between
 - ORGANISATION and GEOPOLITICAL_LOCATION
 - medical INVESTIGATION and BODY_PART
 - GENE and CHROMOSOME_LOCATION
- ◊ EMPLOYEE_OF holding between PERSON and ORGANISATION
- ◊ PRODUCT_OF holding between ARTIFACT and ORGANISATION
- ◊ IS_EXPOSED_TO holding between ORGANIZATION and RISK
- ◊ IS_ASSOCIATED_WITH holding between DRUG and SIDE_EFFECT
- ◊ INTERACTION holding between PROTEIN and PROTEIN

Relation Extraction Task: Challenges

Relation Extraction is challenging for several reasons:

- The same relation may be expressed in many different ways:
 - ◊ Canonical: [Microsoft]_{ORG} is located in [Redmond]_{LOC}
 - ◊ Synonyms: [Microsoft]_{ORG} is based/headquartered in [Redmond]_{LOC}
 - ◊ Syntactic variations:
 - [Microsoft]_{ORG}, the software giant and . . . , is based in [Redmond]_{LOC}
 - [Redmond]_{LOC}-based [Microsoft]_{ORG} . . .
 - [Redmond]_{LOC}'s [Microsoft]_{ORG} . . . ; [Microsoft]_{ORG} of [Redmond]_{LOC}
 - [Redmond]_{LOC} software giant [Microsoft]_{ORG} . . .

Relation Extraction Task: Challenges

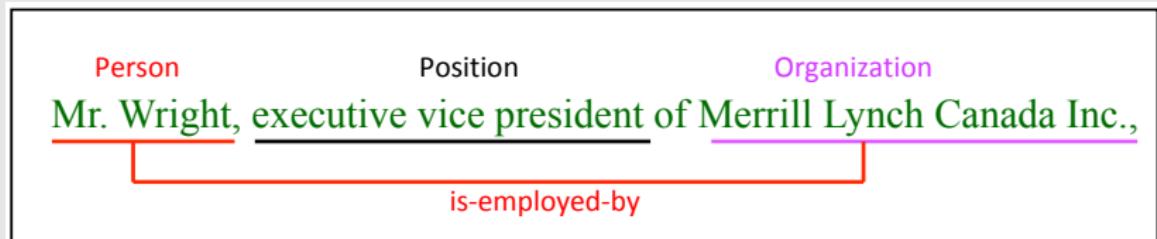
Relation Extraction is challenging for several reasons (cont):

- The information required may be spread across multiple sentences and discovering relations may depend upon following coreference links.
Dirk Ruthless of MegaCorp made a stunning announcement today. In September he will be stepping down as Chief Executive Officer to spend more time with his pet piranhas.
 - ◊ To determine the corporate position of Dirk Ruthless we must correctly resolve the pronominal anaphor “he” in the second sentence with “Dirk Ruthless” in the first
- The information to be extracted may be implied by the text, rather than explicitly asserted, and extracting it may require **inference**
 - ◊ E.g. in the previous example we are not told explicitly that Dirk Ruthless **is** CEO of MegaCorp
 - ◊ To determine this requires knowing (*inter alia*) that stepping down from a position presupposes being in the position prior to stepping down

Relation Extraction: Outline

- Relation Extraction Task
- Approaches to Relation Extraction
 - ◊ Knowledge-engineering approaches
 - ◊ Supervised learning approaches
 - ◊ Bootstrapping Approaches
 - ◊ Distant Supervision Approaches

Knowledge Engineering Approaches



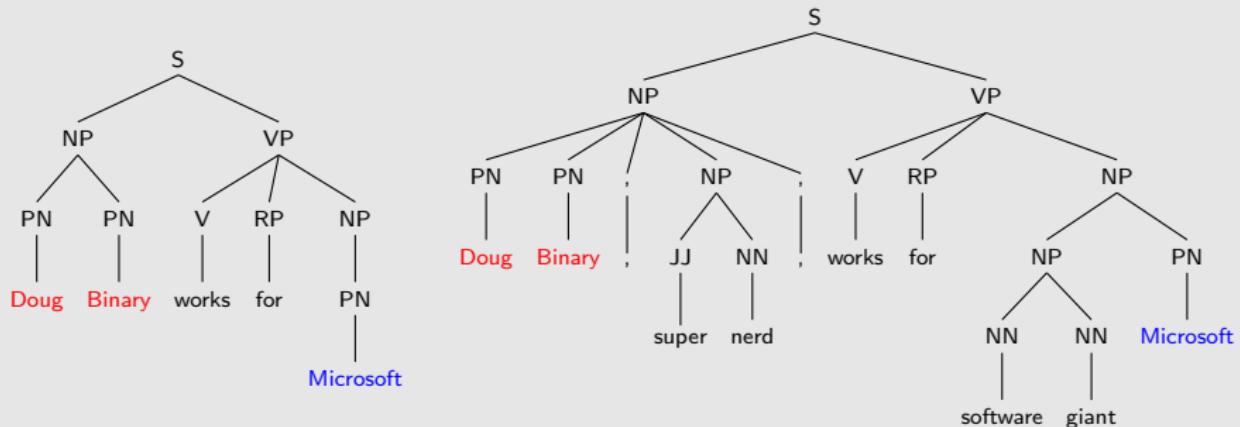
Such systems use manually authored rules and can be divided into

- “shallow” – systems engineered to the IE task, typically using pattern-action rules

Pattern: \\${\$Person}, \${\$Position} of \${\$Organization}

Action: add-relation(is-employed-by(\$Person,\$Organization))

Knowledge Engineering Approaches (cont)



- “deep” – linguistically inspired “language understanding” systems
 - ◊ typically parse input using broad coverage NL parser to identify key grammatical relations, like **subject** and **object**
 - ◊ use transduction rules to extract relations of interest from parser output
 - ◊ extraction rules over parser output allow a wider set of expressions to be captured than with regex's over words and NE tags alone
 - Example shows how multiple surface forms share underlying syntactic structure: here both have form SUBJECT = PER, OBJECT = ORG and VERB = *works for*

Knowledge Engineering Approaches (cont)

- Strengths
 - ◊ High precision
 - ◊ System behaviour is human-comprehensible
- Weaknesses
 - ◊ The writing of rules has no end
 - ◊ New rules needed for every new domain (pattern action rules for shallow approaches; transduction rules for deep approaches)

Relation Extraction: Outline

- Relation Extraction Task
- Approaches to Relation Extraction
 - ◊ Knowledge-engineering approaches
 - ◊ Supervised learning approaches
 - ◊ Bootstrapping Approaches
 - ◊ Distant Supervision Approaches

Supervised learning approaches

- First question to be asked: **What is to be learned?**
- Answer 1: **rules** that
 - ◊ Match to all and only relation bearing sentences
 - ◊ Capture substrings within the matched text that correspond to relation arguments
- Answer 2: **binary classifier** that when applied to a sentence containing instances of the entity types between which the relation holds
 - ◊ Returns 1 if the relation holds in this instance
 - ◊ Returns 0 if the relation does not hold in this instance

As with NER can be divided into detection and classification stages:

- ◊ Classifier 1 (binary) determines whether a given sentence expresses any of a set of relations of interest (**relation detection**)
- ◊ Classifier 2 (multi-way) determines, for positive outputs from Classifier 1, which relation holds (**relation classification**)
- Rule learning approach popular in late 1990's/early 2000's; since then most work focusses on classifier approach – we'll look at the 2nd only

Supervised learning approaches: Classifier Learning

In classification approaches to relation extraction:

- Assume entities to be related already tagged
- Use any algorithm for learning binary classifiers to learn to distinguish instances (typically sentences) where
 - ◊ entities co-occur and relation holds (positive instances)
 - ◊ entities co-occur and relation does not hold (negative instances)
- Key issue: what **features** do we use to represent the instances?
Features used fall into 3 broad classes:
 - ◊ Features of the named entities
 - ◊ Features from the words in the text, usually words from 3 locations
 - words between the two NE candidate arguments
 - words in a fixed window to the left of the 1st candidate
 - words in a fixed window to the right of the 2nd candidate
 - ◊ Features about the entity pair within the sentence, e.g.
 - how far the entities are apart (in words or constituents)
 - whether other NE's occur between them
 - features from the syntactic structure of the sentence

Classifier Learning – Example

- Suppose we have the sentence

[*ORG* American Airlines], a unit of [*ORG* AMR Corp.], immediately matched the move, spokesman [*PER* Tim Wagner] said.

(Jurafsky and Martin, 2nd ed., p. 730)

- Then features extracted for this example when classifying the tuple:
< American Airlines, Tim Wagner >

Entity-based features

Entity ₁ type	ORG
Entity ₁ head	<i>airlines</i>
Entity ₂ type	PERS
Entity ₂ head	<i>Wagner</i>
Concatenated types	ORGPERS

Word-based features

Between-entity bag of words	{ <i>a, unit, of, AMR, Inc., immediately, matched, the, move, spokesman</i> }
Word(s) before Entity ₁	NONE
Word(s) after Entity ₂	<i>said</i>

Syntactic features

Constituent path	$NP \uparrow NP \uparrow S \uparrow S \downarrow NP$
Base syntactic chunk path	$NP \rightarrow NP \rightarrow PP \rightarrow NP \rightarrow VP \rightarrow NP \rightarrow NP$
Typed-dependency path	<i>Airlines</i> \leftarrow_{subj} <i>matched</i> \leftarrow_{comp} <i>said</i> \rightarrow_{subj} <i>Wagner</i>

(Jurafsky and Martin, 2nd ed., p. 738)

Supervised learning approaches

- Strengths:
 - ◊ No need to write extensive/complex rule sets for each domain
 - ◊ Same system straightforwardly adapts to any new domain, provided training data is supplied
- Weaknesses:
 - ◊ Quality of relation extraction dependent on quality and quantity of training data, which can be difficult and time consuming to generate
 - ◊ Developing feature extractors can be difficult and they may be noisy (e.g. parsers) reducing overall performance

Relation Extraction: Outline

- Relation Extraction Task
- Approaches to Relation Extraction
 - ◊ Knowledge-engineering approaches
 - ◊ Supervised learning approaches
 - ◊ Bootstrapping Approaches
 - ◊ Distant Supervision Approaches

Bootstrapping approaches

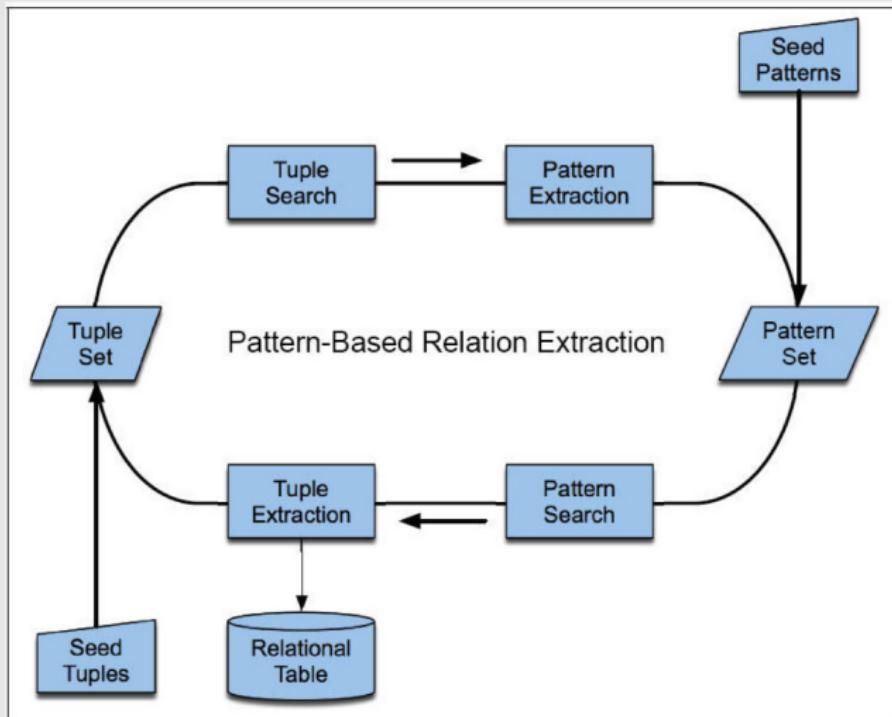
- Motivation: reduce number of manually labelled examples needed to build a system
- Key idea: start with a document collection \mathcal{D} and either :
 - 1 set of trusted tuples \mathbf{T} (e.g. pairs of entities known to stand in the relation of interest)
 - 2 set of trusted patterns \mathbf{P} (i.e. patterns known to extract pairs of entities in the given relation with high accuracy)

Then, if

- 1 then find tuples from \mathbf{T} in sentences \mathbf{S} in \mathcal{D} , extract patterns from context of sentences in \mathbf{S} , add patterns to \mathbf{P} and then use \mathbf{P} to find new tuples in \mathcal{D} and add to \mathbf{T} ; repeat until convergence
- 2 then match patterns from \mathbf{P} in sentences \mathbf{S} in \mathcal{D} , extract tuples from pattern matches in sentences in \mathbf{S} , add tuples to \mathbf{T} and then use tuples in \mathbf{T} to find new patterns in \mathcal{D} and add to \mathbf{P} ; repeat until convergence

Bootstrapping approaches

- Diagrammatically, this can be shown as follows:



(Jurafsky and Martin, 2nd ed., p. 740)

Bootstrapping approaches – DIPRE

- One early system employing this approach was **DIPRE** – Dual Iterative Pattern Relation Expansion – proposed by Sergie Brin (1999)
- Aim: to extract useful relational tuples from the Web, of the form (PERSON, BOOK_TITLE) – e.g. (Leo Tolstoy, War and Peace)
- Method:
 - ◊ Exploit “duality of patterns and relations”
 - Good tuples help find good patterns
 - Good patterns help find good tuples
 - ◊ Starting with user-supplied tuples, iteratively
 - Use these tuples to find patterns
 - Use the patterns to find more tuples

Bootstrapping approaches – DIPRE (cont)

The main loop in DIPRE is as follows:

1 $R' \leftarrow \text{Sample}$

R' is an approximation of the target relation (a set of tuples);

Sample is a small user-supplied sample (e.g. 5 author-title pairs)

2 $O \leftarrow \text{FindOccurrences}(R', D)$

Find all occurrences of tuples of R' in D

3 $P \leftarrow \text{GenPatterns}(O)$

Generate patterns based on the set of occurrences – want patterns to have low error rate and, ideally, high coverage (can compensate for latter with large database (e.g. the Web))

4 $R' \leftarrow M_D(P)$

Update R' with the set of tuples from documents in D that matched by patterns in P

5 If R' is large enough return ; else go to 2.

Bootstrapping approaches – DIPRE (cont)

Brin reports an experiment with finding (author,title) pairs on the web

- **Patterns** are defined as 5-tuples:
(order, urlprefix, prefix, middle, suffix)
 - ◊ If order is true an (author, title) pair matches the pattern if there is a document in the collection (web)
 - whose URL matches urlprefix*
 - which contains text which matches the RE *prefix, author, middle, title, suffix*
 - more detailed RE's are given for author and title
 - ◊ If order is false title and author are switched
- **Occurrences** are defined as 7-tuples:
(author, title, order, url, prefix, middle, suffix)
 - ◊ Order records the order the author and title occurred in the text
 - ◊ URL is the URL of the document the occurrence was found in
 - ◊ Prefix is the m characters (in tests m=10) preceding the author (or title)
 - ◊ Middle is text between author and title
 - ◊ Suffix is m characters following title (or author)

Bootstrapping approaches – DIPRE (cont)

- An algorithm for generating a pattern given a set of occurrences is described
 - ◊ Algorithm insists *order* and *middle* of all occurrences is the same – they form part of the generated pattern
 - ◊ Additionally pattern contains
 - longest matching prefix of the *url* of all the occurrences
 - longest matching suffix of the *prefix* of all the occurrences
 - longest matching prefix of the *suffix* of all the occurrences
 - See Brin (1999) for details
- Patterns are assessed for specificity and rejected if their specificity is too low, i.e. if they are too general
 - ◊ Specificity of a pattern is defined in terms of the product of the lengths of the pattern's *middle*, *urlprefix*, *prefix* and *suffix*
 - ◊ For a pattern p , $\text{specificity}(p) \times n$ must exceed some threshold t , where n is the number of books with occurrences supporting the pattern p

Bootstrapping approaches – DIPRE Experiment

- Used 24 million web pages + 5 seed tuples

Author	Title
Isaac Asimov	The Robots of Dawn
David Brin	Startide Rising
James Gleick	James Gleick
Charles Dickens	Great Expectations
William Shakespeare	The Comedy of Errors

- Yielded 199 occurrences and generated 3 patterns
- These 3 patterns produced 4047 unique (author, title) pairs
- A search over 5 million web pages yielded 3972 occurrences of these books – stopped at this point due to computational constraints
- These occurrences produced 105 patterns which in turn produced 9369 (author, title) pairs – some had bad authors and were rejected
- Using these working pairs in a final iteration resulted in 9988 occurrences, then 346 patterns and then 15257 unique books
- Manual inspection of 20 from the final list showed 19 were bonafide books and 1 was an article

Bootstrapping approaches

- Strengths:
 - ◊ Need for manually labelled training data is eliminated
- Weaknesses:
 - ◊ Can suffer from **semantic drift** – when an erroneous pattern introduces erroneous tuples, which in turn lead to erroneous patterns
 - Introduction of confidence measures for patterns and tuples can mitigate against this problem to some extent
 - ◊ Works well when significant redundancy in assertion of specific tuples and in use of specific patterns to express a relation
 - True for some domains/relations and text collections, not for others
 - ◊ Issues when multiple relations hold between the same pair of entities
 - e.g. suppose someone is born, is educated and dies in the same location, then a sentence containing occurrences of person name and location name could be expressing any of three relations

Relation Extraction: Outline

- Relation Extraction Task
- Approaches to Relation Extraction
 - ◊ Knowledge-engineering approaches
 - ◊ Supervised learning approaches
 - ◊ Bootstrapping Approaches
 - ◊ Distant Supervision Approaches

Distant Supervision Approaches

- As with bootstrapping approaches, **distant supervision** approaches aim to reduce/eliminate the need for manually labelled training data
- Key idea:
 - ◊ Suppose we have a large document collection \mathcal{D} plus a structured data source (e.g. a database) \mathcal{R} that contains
 - many instances of a relation of interest in, e.g., a relational table
 - optionally, for each relation instance a link to a document in \mathcal{D} providing evidence for the relation
 - ◊ Then we can
 - search for sentences in \mathcal{D} containing the entity pairs that occur in relation instances (tuples) in \mathcal{R}
 - label these sentences as positive occurrences of the relation instance
 - use the labelled sentences as training data to train a standard supervised relation extractor

Distant Supervision approaches (cont)

- One well-known approach using distant supervision is described by Mintz et al. (2009)
- Mintz et al. use **Freebase** as their structured data source

Relation name	Size	Example
/people/person/nationality	281,107	John Dugard, South Africa
/location/location/contains	253,223	Belgium, Nijlen
/people/person/profession	208,888	Dusa McDuff, Mathematician
/people/person/place_of_birth	105,799	Edwin Hubble, Marshfield
/dining/restaurant/cuisine	86,213	MacAyo's Mexican Kitchen, Mexican
/business/business_chain/location	66,529	Apple Inc., Apple Inc., South Park, NC
/biology/organism_classification_rank	42,806	Scorpaeniformes, Order
/film/film/genre	40,658	Where the Sidewalk Ends, Film noir
/film/film/language	31,103	Enter the Phoenix, Cantonese
/biology/organism_higher_classification	30,052	Calopteryx, Calopterygidae
/film/film/country	27,217	Turtle Diary, United States
/film/writer/film	23,856	Irving Shulman, Rebel Without a Cause
/film/director/film	23,539	Michael Mann, Collateral
/film/producer/film	22,079	Diane Eskenazi, Aladdin
/people/deceased_person/place_of_death	18,814	John W. Kern, Asheville
/music/artist/origin	18,619	The Octopus Project, Austin
/people/person/religion	17,582	Joseph Chartrand, Catholicism
/book/author/works_written	17,278	Paul Auster, Travels in the Scriptorium
/soccer/football_position/players	17,244	Midfielder, Chen Tao
/people/deceased_person/cause_of_death	16,709	Richard Daintree, Tuberculosis
/book/book/genre	16,431	Pony Soldiers, Science fiction
/film/film/music	14,070	Stavisky, Stephen Sondheim
/business/company/industry	13,805	ATS Medical, Health care

Source:
Mintz et al. (2009)

Table 2: The 23 largest Freebase relations we use, with their size and an instance of each relation.

Distant Supervision approaches – Mintz et al. (cont)

- Freebase was a free on-line database of structured semantic data
 - ◊ data derived from, e.g. Wikipedia infoboxes + other open access sources
 - ◊ after filtering Mintz et al. derived 1.8 million instances of 102 relations connecting 940,000 entities
 - ◊ Freebase no longer available – bought by Google and now forms part of Google Knowledge Graph (partly free, partly paid access)
 - ◊ Similar current sources are [DBpedia](#) and [WikiData](#)
- Mintz et al. use a dump of the text from Wikipedia as their document collection
 - ◊ dump consists of \approx 1.8 million articles, averaging 14.3 sentences/article
 - ◊ used 800,000 articles for training and 400,000 for testing

Distant Supervision approaches – Mintz et al. (cont)

- **Distant supervision assumption:** if two entities participate in a relation, any sentence that contains those two entities might express that relation.
 - ◊ So, tag all sentences containing the two entity mentions as mentions of the relation
- Same relation may be expressed in different ways in different sentences. E.g.

[Steven Spielberg]'s film [Saving Private Ryan] is loosely based on the brothers' story.

Allison co-produced the Academy Award- winning [Saving Private Ryan], directed by [Steven Spielberg]...

- ◊ So, combine features from multiple mentions to get a richer feature vector
- ◊ Use multiclass logistic regression as a learning framework
- ◊ At test time features are combined from all occurrences of a given entity pair in the test data and the most likely relation (or none) is assigned

- Also need **negative instances** – an ‘unrelated’ relation!
 - ◊ to get these, randomly select entity pairs that do not appear in any Freebase relation and extract features for them
 - ◊ Could be related – i.e. wrongly omitted from Freebase – but effect of these rare occurrences should be low
- Mintz et al. evaluate their approach
 - ◊ humans evaluate highest ranked 100 and 1000 results per relation for 10 relations
 - ◊ average precision for best feature combinations just under 70% (69% for top 10; 68% for top 1000)
 - ◊ these results are competitive for knowledge engineering and “normal” supervised learning systems, which struggle to get over 75% on similar tasks

Distant Supervision approaches: Strengths and Weaknesses

- Strengths:
 - ◊ Need for manually labelled training data is eliminated
 - ◊ Can very rapidly get extractors for a wide range of relations
- Weaknesses:
 - ◊ Precision still lags behind best knowledge-engineered/directly supervised learning approaches
 - ◊ Only works if a good supply of structured data is available for the relation(s) of interest

Conclusion

- Relation extraction aims to detect and classify all mentions of a given set of relations holding between specified entity types within a given text
- Relation extraction is a core IE technology that is stubbornly difficult, due to the highly variable ways relations can be expressed in natural language
- Techniques used have included:
 - ◊ Knowledge engineering approaches
 - ◊ Supervised learning approaches
 - ◊ Bootstrapping Approaches
 - ◊ Distant Supervision Approaches
- Open challenges include:
 - ◊ improving precision and recall
 - ◊ handling: relations expressed over > 1 sentences; textual entailment
 - ◊ improving bootstrapping techniques so as to minimise “semantic drift”
 - ◊ developing relation extractors for languages other than English

References

- Agichtein, E. and L. Gravano. Snowball: extracting relations from large plain-text collections. In Proceedings of the 5th ACM Conference on Digital Libraries , 85-94, 2000.
- Brin, S. Extracting patterns and relations from the World Wide Web. In WebDB Workshop at 6th International Conference on Extending Database Technology, EDBT'98 , 172–183, 1998.
- Jurafsky, D and Martin, J. H. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition. 2nd ed. Pearson Inc. 2009. See Chapter 22.2 “Relation Detection and Classification”.
- Mintz, Mike, Steven Bills, Rion Snow and Dan Jurafsky. Distant supervision for relation extraction without labeled data. In Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2 , 1003–1011, 2009.