Information Retrieval

Lecture 8: Automatic Summarisation

Computer Science Tripos Part II

Simone Teufel

Natural Language and Information Processing (NLIP) Group

sht25@cl.cam.ac.uk

Summarisation – an impossible task?

- Summarisation is intelligent and linguistically viable information compression
- Part of human activity in many different genres
  - TV guide: movie plot summaries
  - Blurb on back of book
  - Newsflashes
  - Subtitles
- Why do research in automatic summarisation?
  - Practical reasons: information compression needed in today's information world
  - Scientific reasons: summarisation is a test bed for current document understanding capabilities

Text Summarisation

- Compress the “most important” points of a text, express these main points in textual form
- Information reduction
- Different types of summaries
  - informative/indicative
    - informative: summary replaces full document v.
    - indicative: decision aid for question “should I read the full document?”
  - abstract/extract
    - abstract (generated text) v.
    - extract (verbatim text snippets)

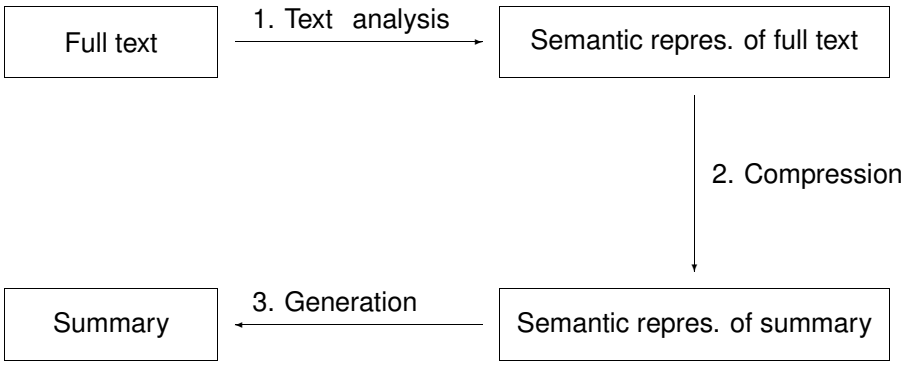
Properties of a good summary

- Considerably shorter than the input text
- Covers main points of input text
- Truth-preserving
- A good text in its own right (coherence...)
- Additional goals: flexibility (with respect to length, user, task)

Human abstracting

- Abstractors are employed at indexing/abstracting companies which produce abstract journals
- Need expert knowledge about summarising and about domain
- Several studies of human abstractors (Cremmins 1996, Endress-Niggemeyer 1995, Liddy 1991)
- Studies show that human abstractors
  - extract textual material, rework it (Cremmins, E-N)
  - only create new material from scratch when they have to, by generalisation and inference (Cremmins, E-N)
  - have a consistent building plan of a summary in their minds, but agree more on type of information to be put into summary than on the actual sentences (Liddy)
- But: Instructions for abstractors too abstract to be used for actual algorithms

Text summarisation: the deep model



Steps of the deep model:

1. Analysis of text into semantic representation
2. Manipulation (compression) of semantic representation
3. Text generation from semantic representation

How realistic is the deep model?

- Compression methods exist (step 2)
  - Summarisation model by Kintsch and van Dijk (1979), based on propositions and human memory restrictions
  - Reasoning theories, e.g. by Lehnert (1982)
- Natural and flexible text generation exists (step 3), working from semantic representation
  - McKeown et al.: Generation from basketball game statistics, weather reports
  - Moore and DiEugenio: Generation of tutor's explanations
- Bottleneck: text analysis (step 1)

Summarisation by fact extraction (Radev and McKeown 1998, CL)

Compress several descriptions about the same event from multiple news stories

MESSAGE: ID	**TST-REU-0001**
SECSOURCE: SOURCE	Reuters
SECSOURCE: DATE	March 3, 1996 11:30
PRIMSOURCE: SOURCE	
INCIDENT: DATE	March 3, 1996
INCIDENT: LOCATION	Jerusalem
INCIDENT: TYPE	Bombing
HUM TGT: NUMBER	"killed: 18" "wounded: 10"
PERP: ORGANIZATION ID	

MESSAGE: ID	**TST-REU-0002**
SECSOURCE: SOURCE	Reuters
SECSOURCE: DATE	March 4, 1996 07:20
PRIMSOURCE: SOURCE	Israel Radio
INCIDENT: DATE	March 4, 1996
INCIDENT: LOCATION	Tel Aviv
INCIDENT: TYPE	Bombing
HUM TGT: NUMBER	"killed: at least 10" "wounded: 30"
PERP: ORGANIZATION ID	

MESSAGE: ID	**TST-REU-0003**
SECSOURCE: SOURCE	Reuters
SECSOURCE: DATE	March 4, 1996 14:20
PRIMSOURCE: SOURCE	
INCIDENT: DATE	March 4, 1996
INCIDENT: LOCATION	Tel Aviv
INCIDENT: TYPE	Bombing
HUM TGT: NUMBER	"killed: at least 13" "wounded: more than 100"
PERP: ORGANIZATION ID	"Hamas"

MESSAGE: ID	**TST-REU-0004**
SECSOURCE: SOURCE	Reuters
SECSOURCE: DATE	March 4, 1996 14:30
PRIMSOURCE: SOURCE	
INCIDENT: DATE	March 4, 1996
INCIDENT: LOCATION	Tel Aviv
INCIDENT: TYPE	Bombing
HUM TGT: NUMBER	"killed: at least 12" "wounded: 105"
PERP: ORGANIZATION ID	"Hamas"

Summarisation by fact extraction (Radev and McKeown 1998, CL)

- Reason over templates
- New templates are generated by combining other templates
- The most important template, as determined by heuristics, is chosen for generation
- Rules:
  - Change of perspective: If the same source reports conflicting information over time, report both pieces of information
  - Contradiction: If two or more sources report conflicting information, choose the one that is reported by independent sources
  - Addition: If additional information is reported in a subsequent article, include the additional information
  - Refinement: Prefer more specific information over more general one (name of a terrorist group rather than the fact that it is Palestinian)
  - Agreement: Agreement between two sources is reported as it will heighten the reader's confidence in the reported fact
  - Superset/Generalization: If the same event is reported from different sources and all of them have incomplete information, report the combination of these pieces of information
  - Trend: If two or more messages reflect similar patterns over time, these can be reported in one statement (e.g. three consecutive bombings at the same location)
  - No Information: Report the lack of information from a certain source when this would be expected
- Output summary, deep-generated:

> Reuters reported that 18 people were killed in a Jerusalem bombing Sunday. The next day, a bomb in Tel Aviv killed at least 10 people and wounded 30 according to Israel Radio. Reuters reported that the radical Muslim group Hamas had claimed responsibility for the act.

- Problem: domain-specificity built into the templates

A domain-inspecific method: text extraction

- Split text in units (paragraphs or sentences or text tiles)
- Assign each unit a score of importance/“extractworthiness”, using sentential and/or relational features
  - Sentential features of a unit can be calculated in isolation, e.g. number of TF/IDF words or location
  - Relational features of a unit are calculated in context of other units, e.g. unit with highest amount of shared terms
- Extract sentences with highest score verbatim as extract

External marking of “more important” material

- Text is globally structured (rhetorical sections, anecdotal/summary beginning in journalistic writing) – location feature
- Text is locally structured (paragraph structure; headlines and sub-headlines) – paragraph structure feature
- Important concepts/terms mark important prepositions – tf/idf feature
- Certain typographic regions are good places to find important concepts: captions, title, headlines – title feature
- Sentence length is important, but the experts argue; probably genre-dependent
- Phrases mark important sections (“in this paper”, “most important”) and less important sections (hedging by auxiliaries, adverbs) – cue phrase feature

Sentential features, I

1. Concept feature (Luhn, 1958)
   - Find concepts using *tf* (nowadays: *tf*idf*), sentence score = no of frequency concepts in sentence
2. Header feature (Baxendale, 1959)
   - Find concepts in title (variation: title and headlines), sentence score = no of title concepts in sentence
3. Location feature (Edmundson, 1969)
   - Divide text into n equal sections
   - sentences in section $1 \leq i \leq n$ get sentence score = $\frac{1}{i}$
   - Always used in combination

Sentential features, II

4. Paragraph feature
   - First sentence in paragraph gets a higher score than last one, and higher than sentences in the middle
   - Always used in combination
5. Cue phrases (Paice, 1991)
6. First-sentence-in-section feature
7. Sentence length
8. Occurrence of bonus or malus word (ADAM system, Zomora (1972))
9. Occurrence of a named entity (Kupiec et al., 1995)

Combination of sentential features: manually

- Combinations of features are more robust than single features
- Manual feature combination (Edmundson):

$$Score(S) = \alpha A + \beta B + ...\omega O$$

A, B,..O: feature scores
α, β, ω: manual weights

Combination of sentential features: machine learning

- Kupiec, Pedersen, Chen: A trainable document summariser, SIGIR 1995
- Create examples of sentences that are abstract-worthy, calculate their features, using 5 well-known features ($F_1 \ldots F_5$)
- Use Naive Bayesian classifier:

$$P(s \in S|F_1, \ldots, F_k) = \frac{P(F_1,...,F_k|s \in S)P(s \in S)}{P(F_1,...,F_k)} \approx \frac{P(s \in S)\prod_{j=1}^{k} P(F_j|s \in S)}{\prod_{j=1}^{k} P(F_j)}$$

$P(s \in S\|F_1, \ldots, F_k)$:	Probability that sentence s from the source text is included in summary S, given its feature values;
$P(s \in S)$:	Probability that a sentence s in the source text is included in summary S unconditionally; compression rate of the task (constant);
$P(F_j\| s \in S)$:	probability of feature-value pair occurring in a sentence which is in the summary;
$P(F_j)$:	probability that the feature-value pair F_j (j th feature-value pair out of k feature-value pairs) occurs unconditionally;

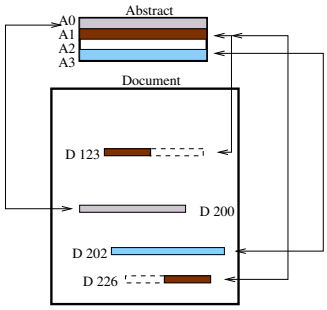
Finding the right gold standard

Subjective measures:

- Humans subjects select sentences (system developers?)

Looking for more objective measures:

- Earl: indexible sentences
- Kupiec et al: sentences with similarity to abstract sentences



Kupiec et al: gold standard

- Find best match for each abstract sentence by automatic similarity measure
- One example for a similarity measure is based on the longest common substring:

$$lcs(X,Y) = \frac{length(X) + length(Y) - edit_{i,d}(X,Y)}{2}$$

(where $edit_{i,d}$ is the minimum number of deletions and insertions needed to transform X into Y).

- Possible similarity measures are the ratio of longest common substring to the maximum length of the two sentences, or the average.
- Reject sentences with similary $<$ .5; accept sentences with similarity $>$ 0.8, hand-judge sentences with medium similarity $.5 \leq X \leq .8$

Kupiec et al's evaluation

- Corpus of 85 articles in 21 journals
- Extract as many sentences as there are gold standards in the document → precision = recall
- Very high compression makes this task harder
- Results:

Feature	Individual	Cumulative
Cue Phrases	33%	33%
Location	29%	42%
Sentence Length	24%	44%
*tf*idf*	20%	42%
Capitalization + *tf*idf*	20%	42%
Baseline	24%	

Example of an extract (Microsoft's AutoSummarize)

Distributional Clustering of English Sentences
Distributional Similarity To cluster nouns n according to their conditional verb distributions pn, we need a measure of similarity between distributions.
We will take (1) as our basic clustering model.
In particular, the model we use in our experiments has noun clusters with cluster memberships determined by p(njc) and centroid distributions determined by p(vjc).
Given any similarity measure d(n;c) between nouns and cluster centroids, the average cluster distortion is
If we maximize the cluster membership entropy
Clustering Examples
Figure 1 shows the five words most similar to the each [sic] cluster centroid for the four clusters resulting from the first two cluster splits.
Model Evaluation
1990. Statistical mechanics and phrase transitions in clustering.

Source: "*Distributional Clustering of English Sentences*" by Pereira, Tishby and Lee, ACL 1993

What are extracts good for? 21

- Extraction is the basis of all robust and reliable summarisation technology widely deployed nowadays
- It can give readers a rough idea of what this text is about
- Information analysts work successfully with them
- Task-based evaluation results:
  - Tombros et al. (1998) show slight improvement in precision and recall and larger improvement in time for a human search task
  - Mani et al. (1999) slight loss in accuracy and large advantage in time saving (50% of the time needed) for a relevance decision task

Problems with extracts 22

- Unclear to reader why particular sentence was chosen
- Coherence (syntactic, local problems)
  - Dangling anaphora
  - Unconnected discourse markers
- Cohesion (semantic discontinuities, global)
  - Concepts and agents are not introduced
  - Succession of events does not seem coherent

Fixes for coherence problems

- E.g. dangling anaphora:
  - resolve anaphora
  - recognize anaphoric use (as opposed to expletive use ("it", Paice and Husk 1987), then either
    - exclude sentences with dangling anaphora
    - include previous sentence if it contains the referent (Johnson et al. 1993; also for definite NPs) – But: length!
- There are no fixes for cohesion

Strategies for summary evaluation

1. Subjective judgements:
   How much do subjects like this summary? How coherent, well-written, etc do they find it?
2. Comparison to "gold standard" (predefined right answer):
   In how far does this summary resemble the "right answer"?
3. Task-based evaluation:
   How well can humans perform a task if they are given this summary?
4. Usability evaluation (extrinsic):
   Does the recipient of the summary have to change it? How much?

Problems in Summarisation Evaluation

1. Subjective judgements
   - Subjects can be biased
   - How to make sure they understand the same thing under ”informativeness”, for instance
2. Comparison to “gold standard”
   - by sentence co-selection, surface string similarity or “information overlap”
   - Problematic: humans do not agree on what a good summary is
   - Doubt about existence of a “gold standard”
3. Task-based evaluation
   - Probably the best evaluation around
   - Hard to define the task/set up the experiment
   - Time-consuming and expensive to do experiment
   - For final, end-of-project evaluation, not for day-to-day evaluation

Summary

- Summarisation by deep methods and problems
- Summarisation by text extraction
  - Importance features
  - Kupiec et al.’s (1995) method and training material
  - Lexical chains
- Summarisation evaluation and its problems