Keyword Extraction using Word Co-occurrence
TIR 2010, Bilbao 31 August 2010

Christian Wartena (Novay)
Rogier Brussee (Univ. of Applied Sciences Utrecht, presenter)
Wout Slakhorst (Novay)
Problem description

- Keywords used for organising and retrieval of documents (including non textual ones)
- Problem:
  - Determine keywords automatically

- Operational problem:
  - Define relevance measure of terms
  - Select collection of terms based on relevance
    - Here, just rank
Keywords, world knowledge, informativity

- Relevance of term as keyword depends on:
  - **Importance** of term for the *document*
  - **Discriminative power** of term within *document collection*
  - **A priori criteria**
    - in a thesaurus
    - right word class,
    - non stopword,
    - ...
World knowledge from statistics

• Problem: What can we do if we **do** have access to large document collection?
  – assuming it is a natural document collection

• Importance in the doc collection is (hopefully) a proxy for the importance of terms in “the world”.
  – Importance w.r.t. everything

• Statistics of the collection becomes a source of world knowledge
  – OK to use broad external world knowledge
    • E.g. word class of terms
Predicting the term distribution

• **keyword** is short summary of content of a document

• Use **term distribution** of the document as proxy for the content
  – Bag words model.
  – Distributional hypothesis (Harris 1954)

• Good keywords should **predict** the term distribution of the document
Everything is a distribution

• **Term distribution** of a document:
  – $q_d(t)$ is the term distribution of $d$
  – “The fraction of term occurrences found in $d$, matching $t$”

• **Document distribution** of a term
  – $Q_z(d)$ is the document distribution of $z$
  – “The fraction of term occurrences matching $z$, found in $d$”

• **Background distribution** of the corpus
  – $q(t)$ is the fraction of term occurrences matching $t$
Co-occurrence distribution of a term

- Co-occurrence distribution of a term

\[
\bar{p}_z(t) = \sum_d Q_z(d)q_d(t)
\]

- Average distribution of terms co-occurring with \( t \)
Co-occurrence of tags
“average tag cloud”
Co-occurrence of tags
“average tag cloud”
Co-occurrence of tags
“average tag cloud”

\[ Q(d_2|z) \]

\[ q(t_3|d_1) \quad q(t_3|d_2) \]

\[ q(t_1|d_1) \quad q(t_2|d_1) \]

\[ Q(d_1|z) \]

\[ q(t_4|d_2) \quad q(t_5|d_2) \]
Relevance measure for terms:

- Relevance measure for term $z$
  - Importance: __
    - Closeness of $p_z$ to document distribution $q_d$
  - Specifity __
    - Awayness of $p_z$ from background $q$

- $\rightarrow$ need to specify distance measure!
Different distance measures for distributions

- **Kullback Leibler divergence** $D(p||q)$
  - #bits per term saved by compression on a term stream using true distribution $p$ instead of estimate $q$.
  - Infinite if $p$ is not divisible by $q$!

- **Jensen Shannon divergence** $JSD(p,q)$
  - #bits per term saved by compression using streams distributed like $p$ and $q$ separately instead of mixture

- **Naive correlation coefficient** $r(p,p';q)$
  - Cosine similarity of $(p-q)$ and $(p'-q)$
Relevance measures for terms

- Only weigh closeness of term to document distribution

\[ jsd(z, d) = JSD(p_z, q_d) \]

- Weigh closeness of term to document and awayness to corpus

\[ \Delta(z, d) = D(p_z || q_d) - D(p_z || q) = \sum_t p_z(t) \log \left( \frac{q_d(t)}{q(t)} \right) \]

- Correlate differences

\[ r(z, d) = r(p_z, q_d ; q) \]
Evaluation

• Use 11000 ACM abstracts with keywords.
  – #keywords = 1—10, av = 4.5
  – 27336 distinct keywords,
  – 21634 used only once,
  – 2 used more than 100 times.
  – 21642, consists of more than one word.

• UIMA and GATE based pipeline
Multiword detection

• Imperative to detect multiwords as candidate terms!
  – Algorithm: detect superabundant combinations taking word class into account using t-test (see Manning and Schütze)
  – detection algorithm identified 4817 multiwords.
  – Results sensitive to multiword extraction algorithm ☹, but all methods evaluated suffer ☺.
  – Only 52% of articles has a keyword that is selected as a candidate term after preprocessing. 52% is optimal!
  – Selected terms may be perfectly acceptable keywords
Evaluation BBC dataset

- 2879 BBC Program descriptions (Many very short)
  - #keywords = 1 -- 22 keywords, av = 2.9
  - 1748 distinct keywords,
  - 898 used once
  - 8 used more than a 100 times,
  - 792 keywords consist of multi word.

- Multiword detection algorithm found 168 multiwords.

- 57% of articles has a keyword selected as a candidate term
11000 ACM abstracts
2879 BBC abstracts
Conclusion

• Using co-occurrence data improves on tf-idf
• Slightly naive correlation coefficient works best.
• There is room for improvement
  – Christian Wartena has recently gotten good results with recommendation by using some clustering, and with doc retrieval on keywords (CLEF).
  – Good multiword detection is really important.