MyMedia

Dynamic Personalization of Multimedia

Thesaurus based Keyword Extraction

Luit Gazendam (Novay)
Christian Wartena (Novay)
Rogier Brussee (Univ. of Applied Sciences Utrecht)

Problem description

- Keywords used for organising and retrieval 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,
    - …

Example : TF.IDF

- Consider TF.IDF
  - Input:
    - document collection $\{d_1 \ldots d_N\}$
    - Term collection $\{t_1 \ldots t_n\}$
    - Count's
      - $n(d,t)$ = # occurences of term t in document d
      - df(t) = # documents containing t (doc frequency)
      - N = # documents

tf.idf(d,t) = $n(d,t)$ log(N/df(t))

$n(d,t)$: weighs importance of term in the document

N/df(t) : weighs importance of term in the doc collection

World knowledge from thesaurus structure

- Problem: What can we do if we do not have access to large document collection ?
  - or there is no natural document collection

- Importance in the doc collection is really a proxy for importance of terms in “the world”.
  - Importance w.r.t. everything
    - ever written, English web, Information retrieval literature
- Thesauri are alternative sources of world knowledge
  - also required by many archives

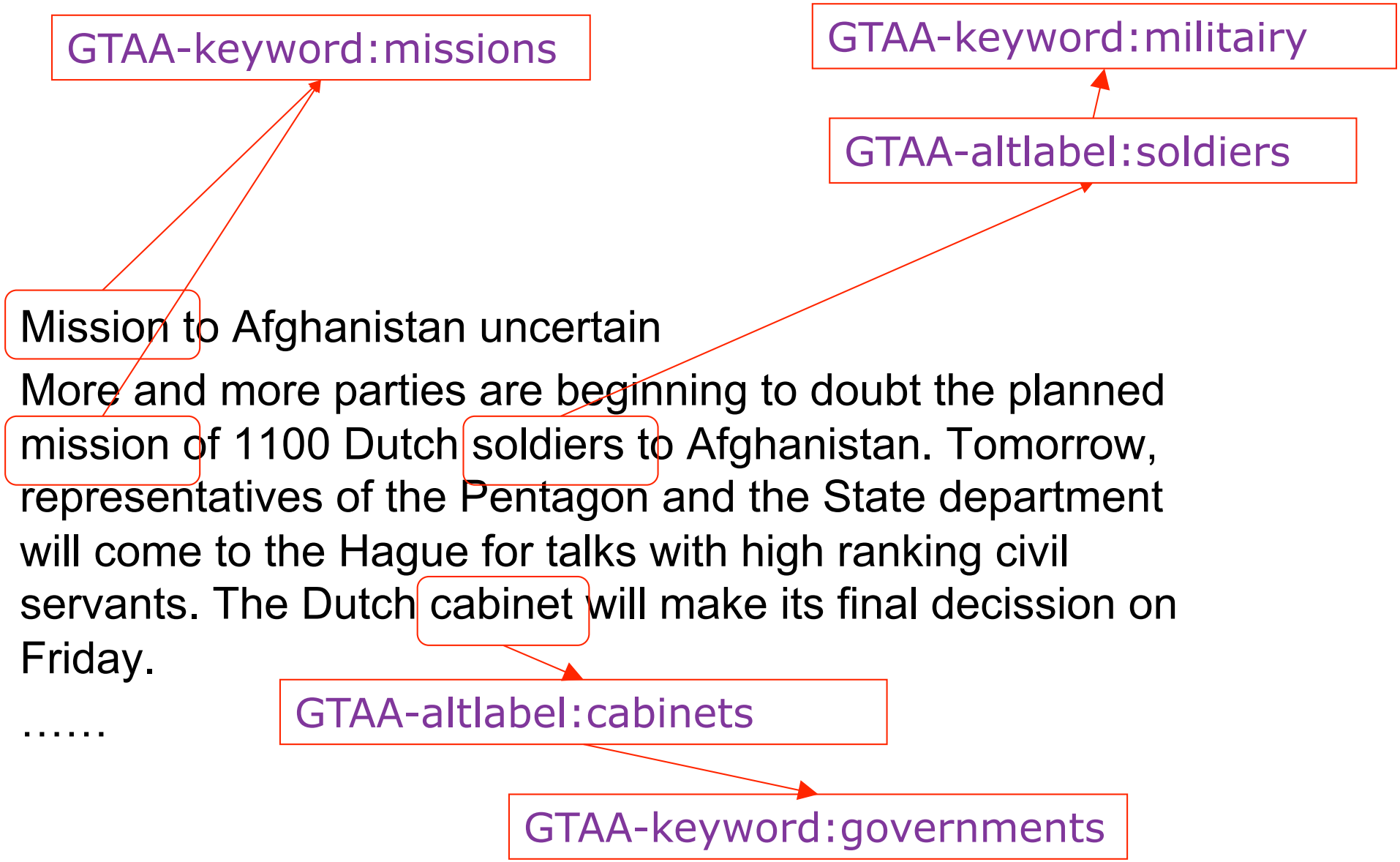
Document

Mission to Afghanistan uncertain

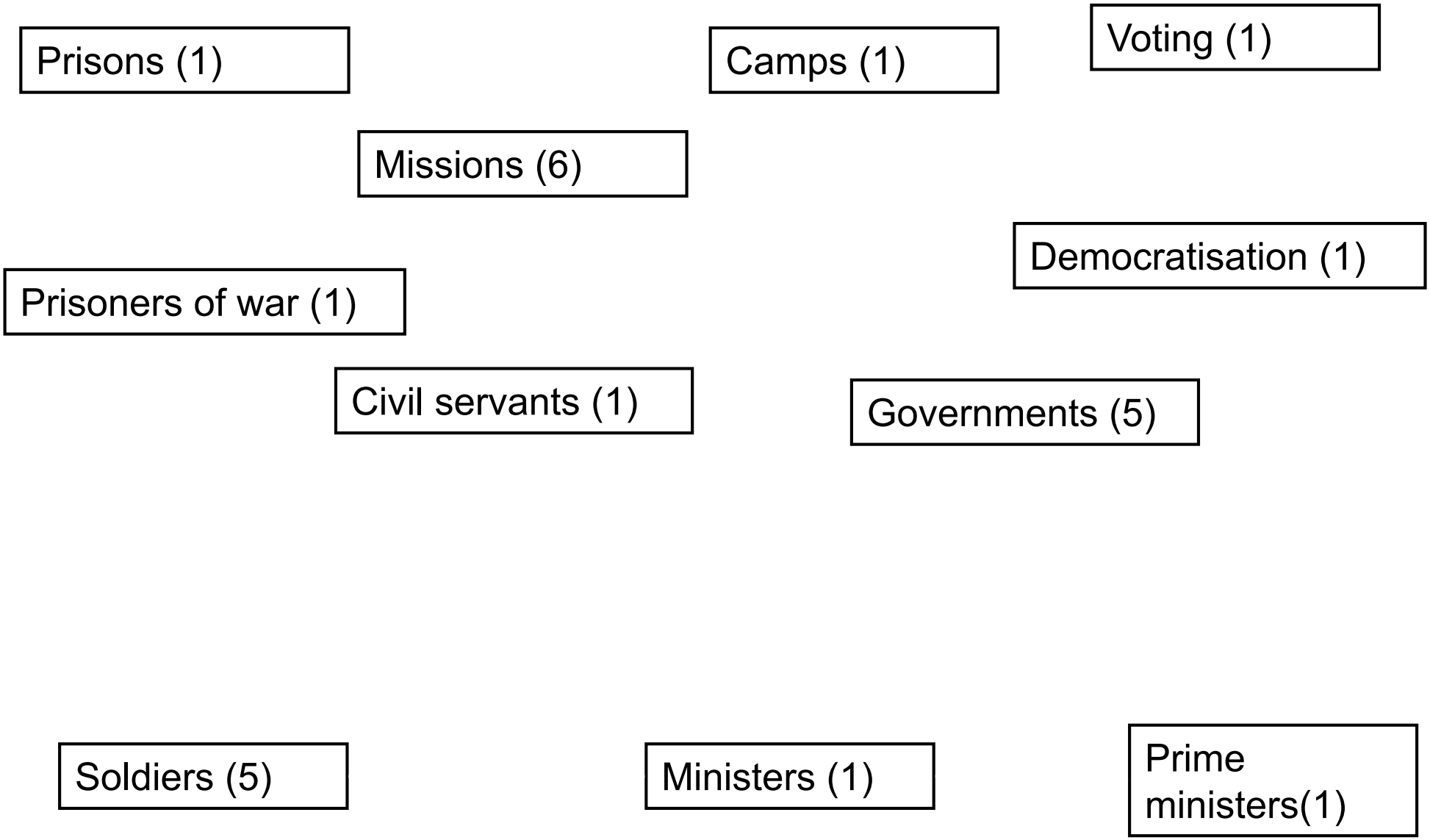
More and more parties are beginning to doubt the planned mission of 1100 Dutch soldiers to Afghanistan. Tomorrow, representatives of the Pentagon and the State department will come to the Hague for talks with high ranking civil servants. The Dutch cabinet will make its final decission on Friday.

Document analysis: find thesaurus terms

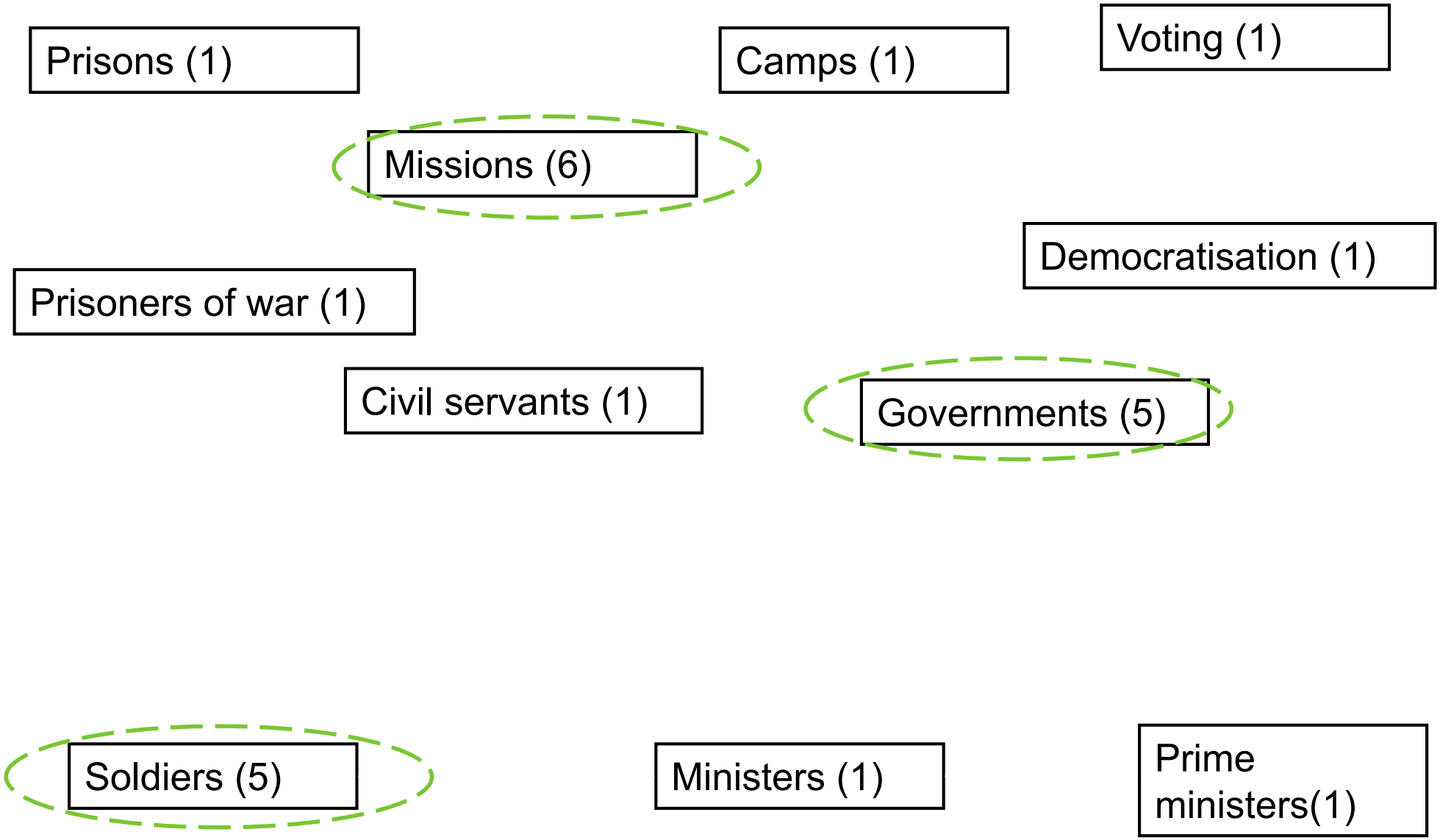
(Apolda semantic annotation tool)



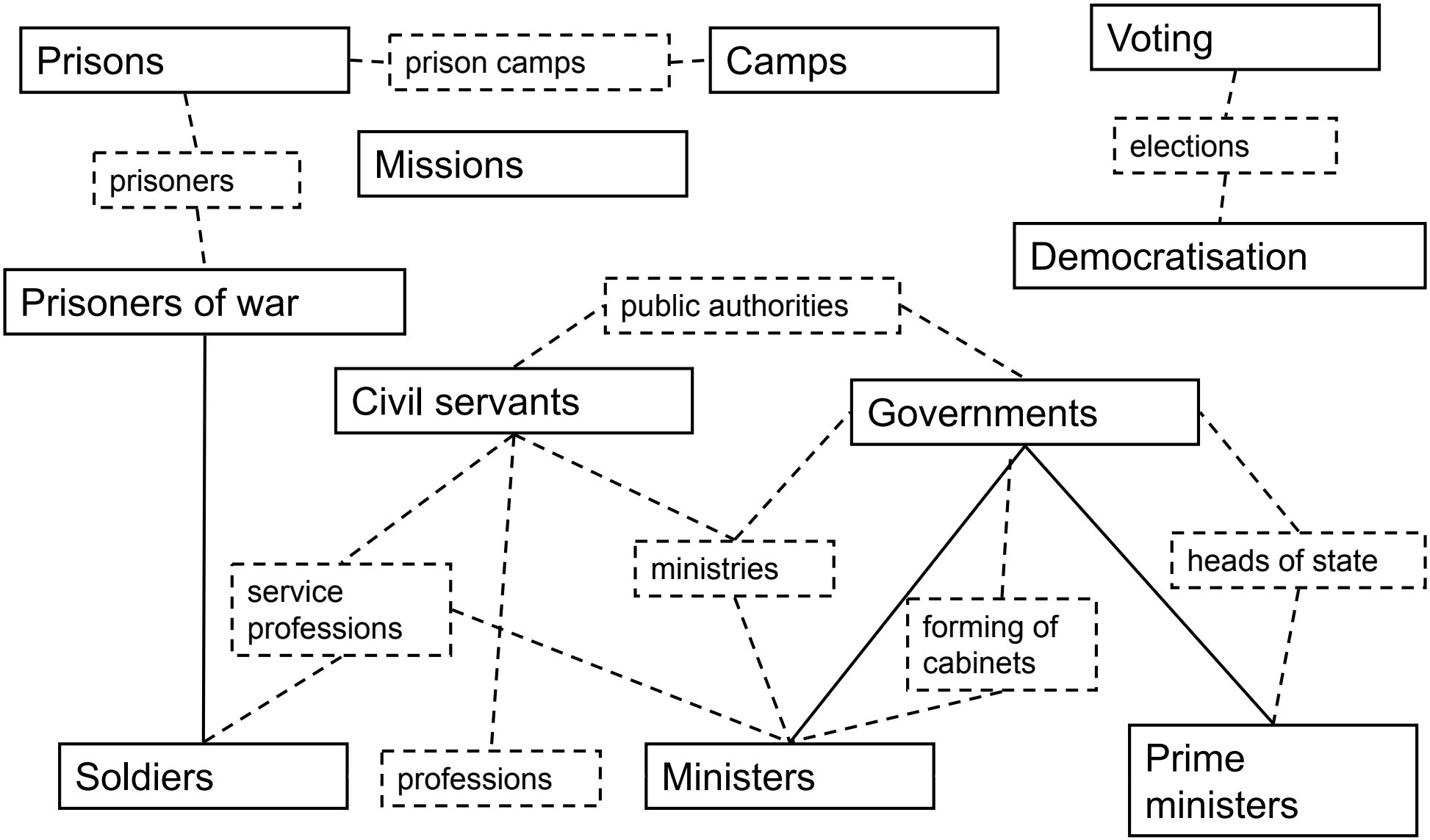
Count lexical representations in a doc.



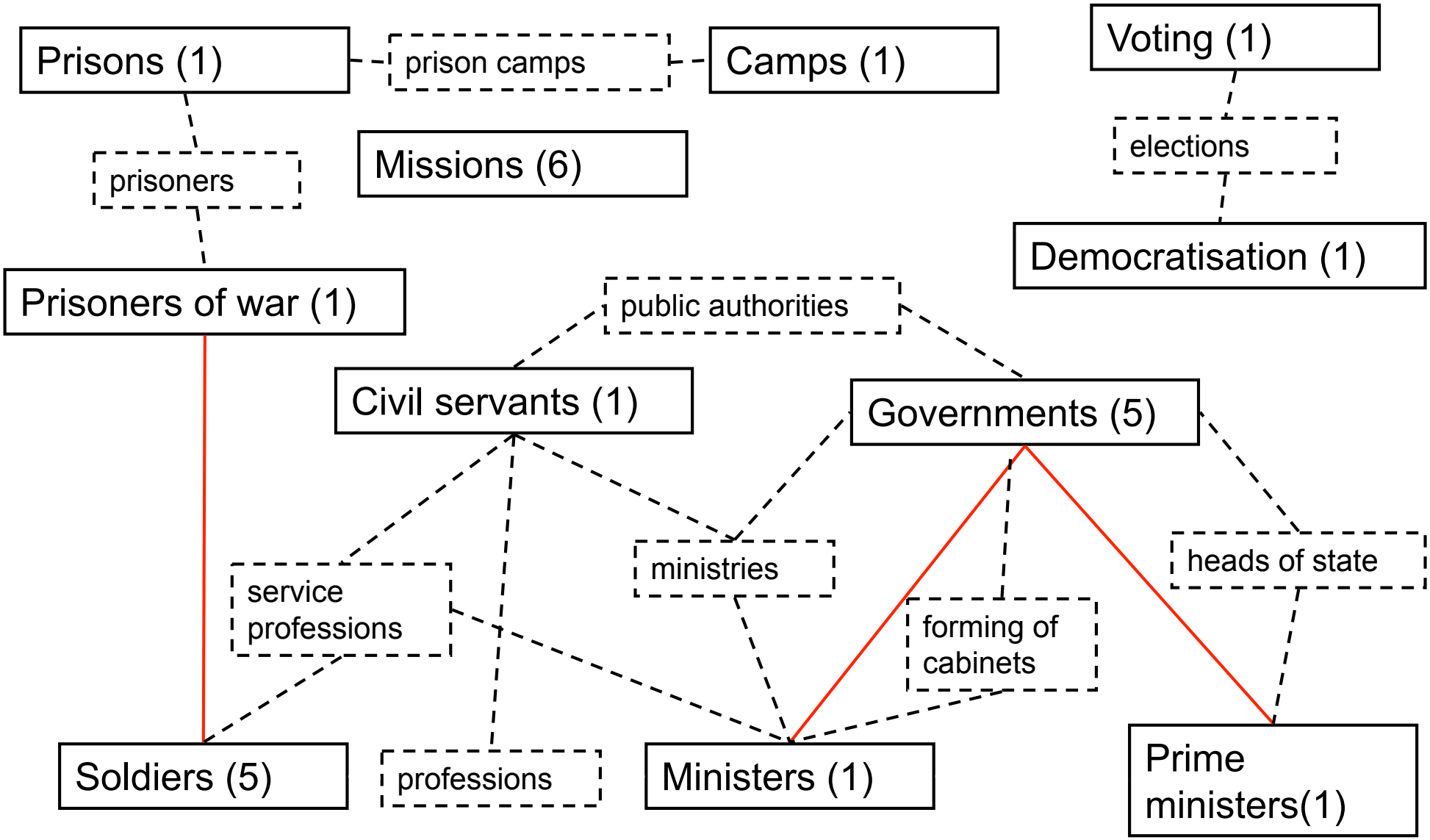
Ranking: frequency of terms (?).



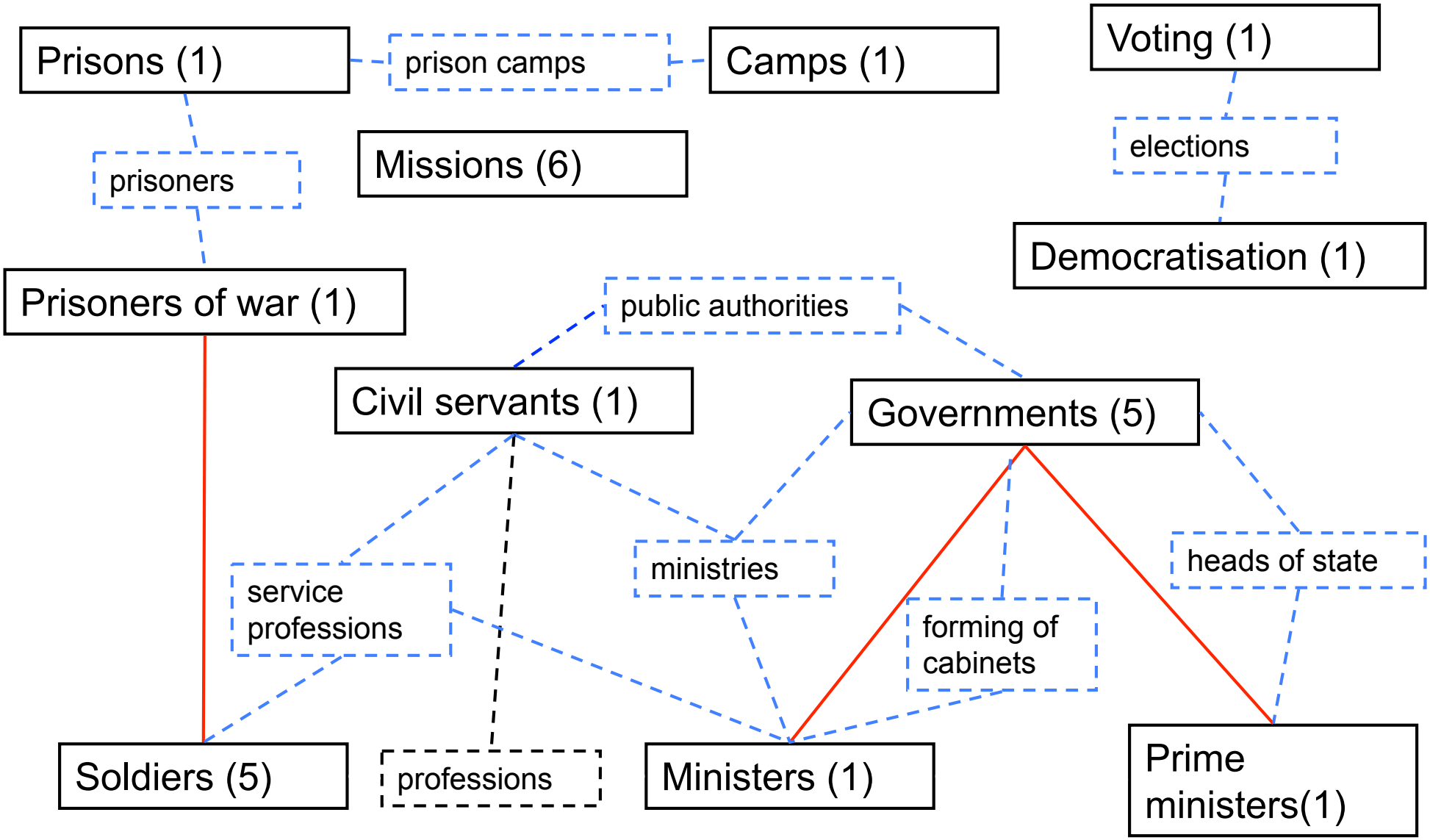
Thesaurus structure (USE, BT, NT, RT).



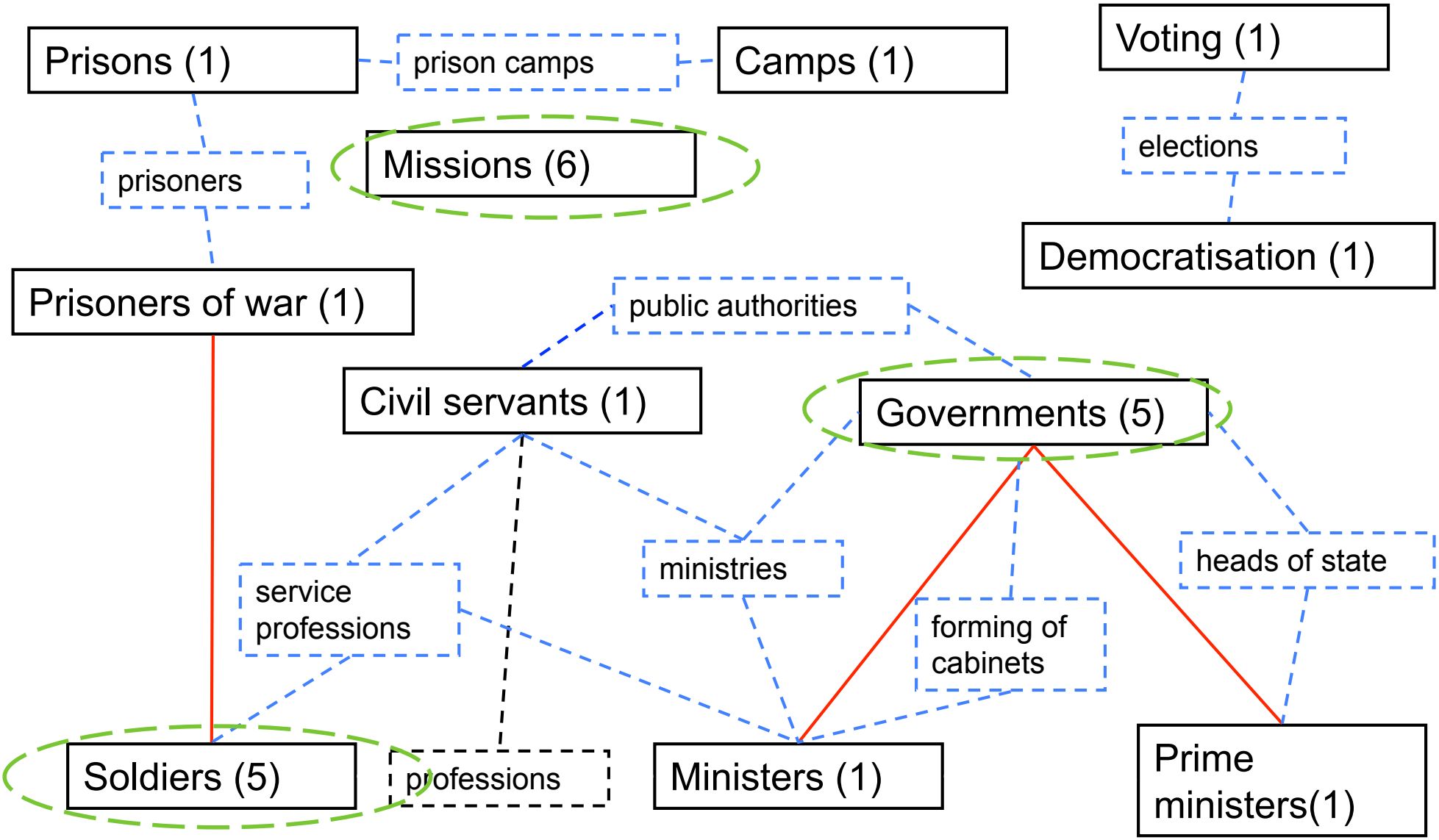
Centrality: realised relations in doc (order = 1)



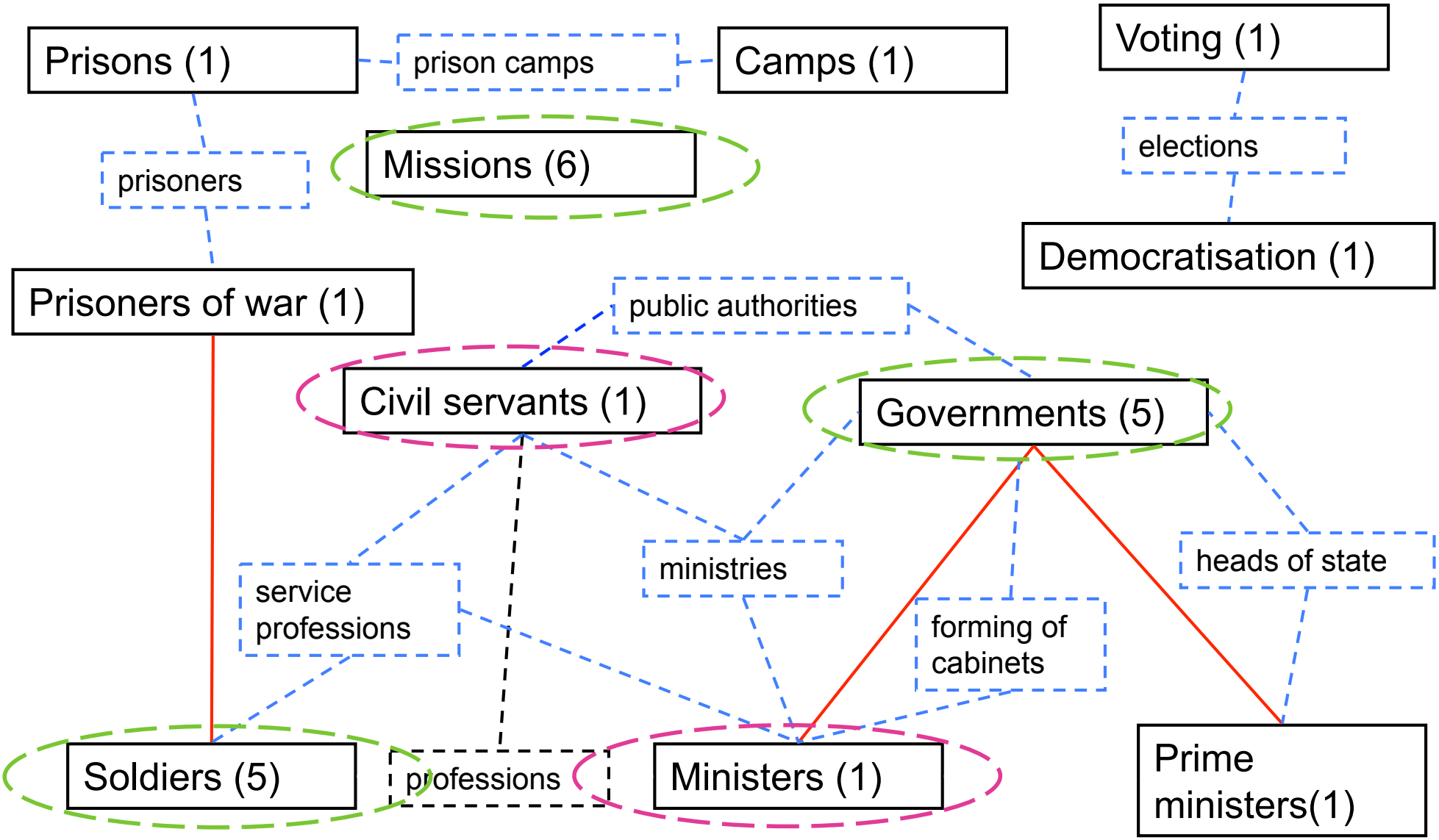
Centrality: realised relations in doc (order ≤ 2)



Ranking: frequency of terms (????)



Ranking: connectedness of terms !



TF.RR: Term frequency, Realised relations

- Select words in text that are concepts in the thesaurus
- Determine weight of (key)words by
  - Frequency
  - Number of thesaurus relations to other words in the text: central words in the text become higher weights

$$tf.rr(t,d) = tf(t,d)rr(t,d)$$

$$tf(t,d) = 1 + \log(n(t,d))$$

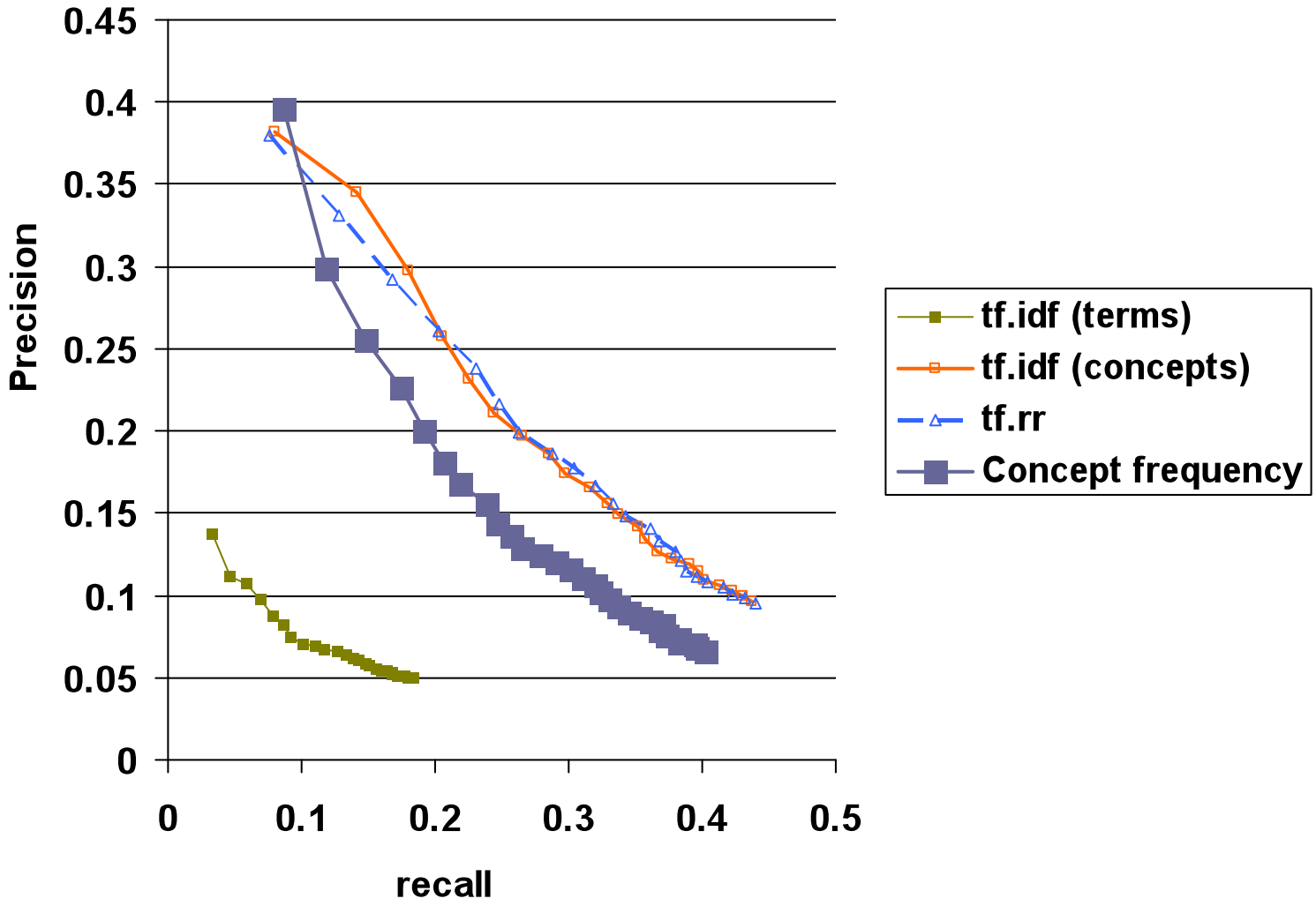
$$rr(t,d) = 1 + \mu r_1(t,d) + \mu^2 r_2(t,d)$$

  - with *n(t, d)* the number of occurrences of *t* in *d*,
  - *μ* = *α/avlinks* where *avlinks* is the average out degree of the thesaurus
    - average number of relations a term has in the thesaurus
  - We set *α* = ½

Evaluation

- Generate and rank keyword suggestions for TV-programs from contextual resources
  - 258 TVbroadcasts
  - 362 context documents,
    - Length = 25 -- 7000 words, av = 1000
  - Thesaurus, so called GTAA (Common Thesaurus Audiovisual Archives)
    - #keywords = 3860, #relations = 20 591
  - Manual annotation by Dutch Sound and Vision Institute.
    - #keywords = 1 -- 15, av = 5.7
  - Manual keywords ground truth for evaluation
    - Inter-annotator consistency 13% --77%, av = 44%

Results


0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
Precision
0
0.1
0.2
0.3
0.4
0.5
recall
tf.idf (terms)
tf.idf (concepts)
tf.rr
Concept frequency

Conclusion

- Using thesaurus relations improves on just counting concepts
- Results comparable to using a corpus. i.e.

A *good* thesaurus is a reasonable alternative to having access to representative corpus

MyMedia

Dynamic Personalization of Multimedia

Keyword Extraction using Word Co-occurrence

Christian Wartena (Novay)
Rogier Brussee (Univ. of Applied Sciences Utrecht)
Wout Slakhorst (Novay)

Problem description

- Keywords used for organising and retrieval 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 a (hopefully) a proxy for 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 occurences in *d* matching *t*
- **Document distribution** of a term
  - $Q_z(d)$ is the document distribution of *z*
  - "The fraction of term occurences matching z found in *d*"
- **Background distribution** of the corpus
  - $q(t)$ is the fraction of term occurences matching *t*

Co-occurrence distribution

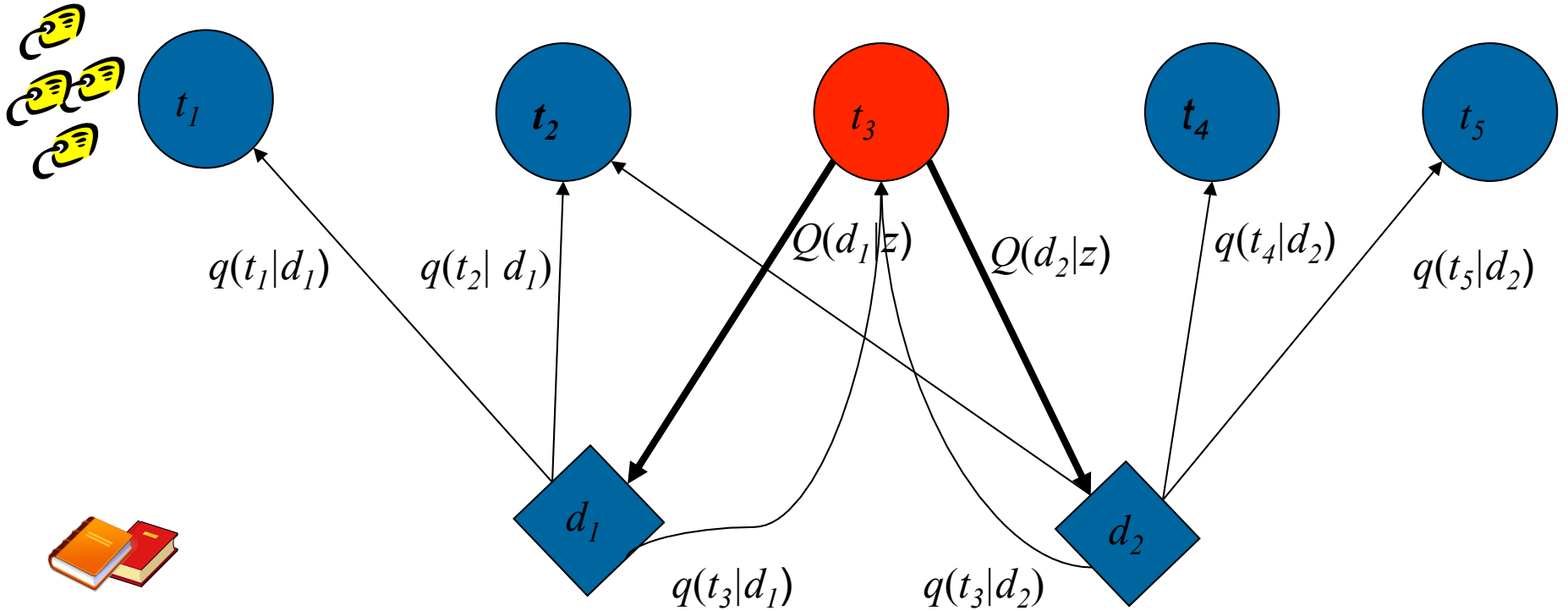
- Co-occurrence distribution of a term

$$\overline{p_z}(t) = \sum_d Q_z(d) q_d(t)$$

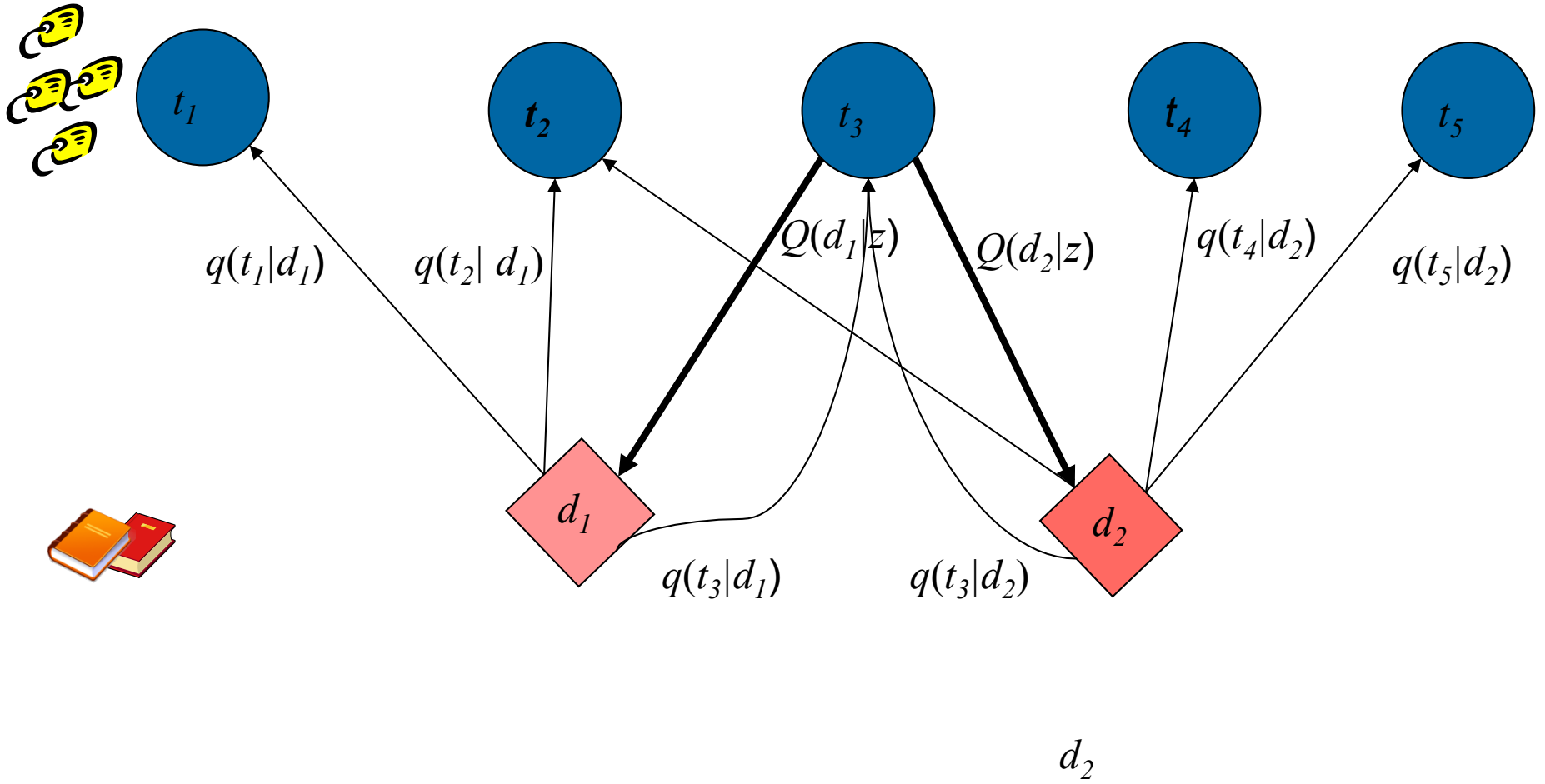
.

- Average distribution of terms co-occuring with t .

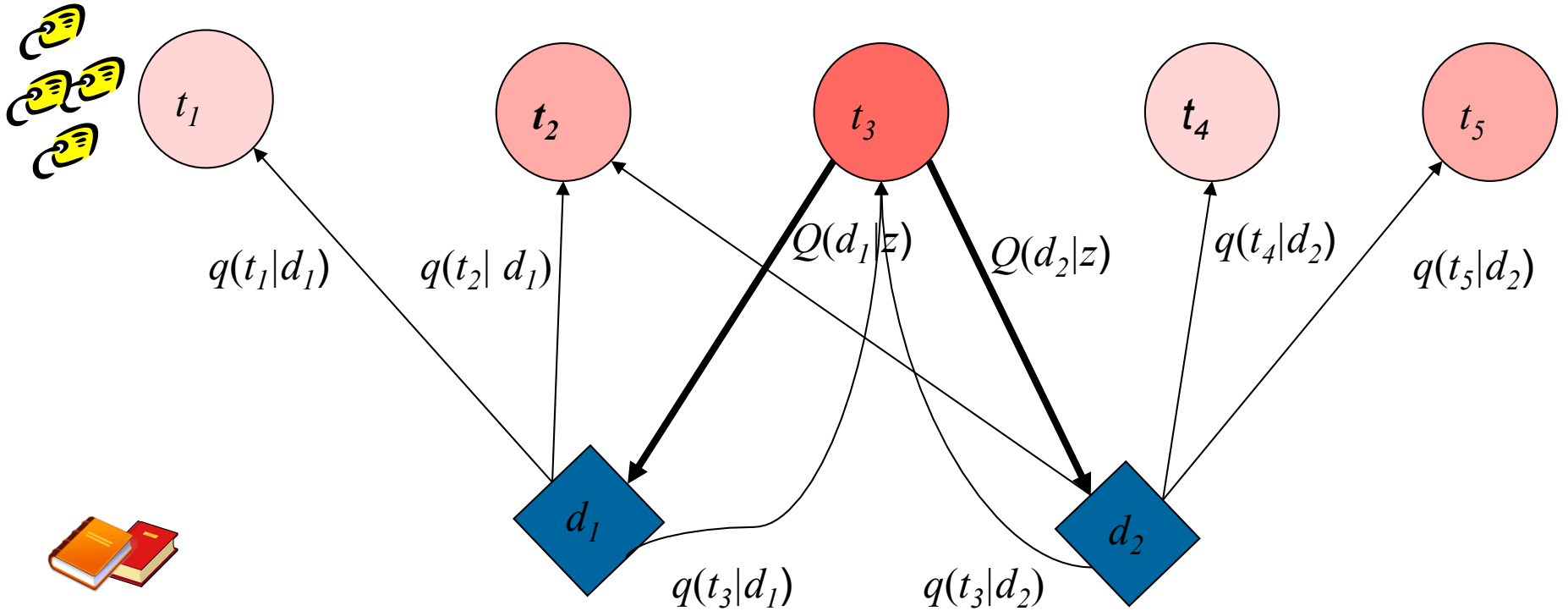
Co-occurrence of tags
“average tag cloud” *(Novay)*



Co-occurrence of tags
“average tag cloud” *(Novay)*



Co-occurrence of tags
“average tag cloud” *(Novay)*



Relevance measure for terms:

- Relevance measure for term *z*
- importance: $\overline{p_z}$
  - Closeness of p_z to document distribution q_d
- Specifity $\overline{p_z}$
  - Awayness of p_z from background *q*

- → 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 not abs continuous wrt q!
- Jensen Shannon divergence JSD(p,q)
  - #bits per term saved by compression using streams distributed like p and q seperately instead of mixture
- Naive correlation coefficient r(p,p';q)
  - Cosine similarity of (p-q) and (p'-q)

Kullback Leibler and Jensen Shannon Divergence

- Kullback Leibler divergence

$$D(p\|q) = \sum_t p(t)\log\left(\frac{p(t)}{q(t)}\right)$$

$$\geq 0 \text{ with equality iff } p = q$$

- Jensen Shannon divergence

$$JSD(p,q) = D(p \,||\, 1/2\,p + 1/2q) + D(q \,||\, 1/2\,p + 1/2q)$$

- Correlation coefficient

$$\frac{\sum_t (\overline{p_z}(t) - q(t))(\overline{q_d}(t) - q(t))}{\sqrt{\sum_t (\overline{p_z}(t) - q(t))^2}\sqrt{\sum_t (\overline{q_d}(t) - q(t))^2}}$$

Relevance measures for terms

- Only weigh closeness of term to document distribution

$$jsd(z,d) = JSD(\overline{p}_z, q_d)$$

- Weigh closeness of term to document and awayness to corpus

$$\Delta(z,d) = D(\overline{p}_z \,||\, \overline{q}_d) - D(\overline{p}_z \,||\, q) = \sum_t \overline{p}_z(t) \log(\frac{\overline{q}_d(t)}{q(t)})$$

- Correlate differences

$$r(z,d) = r(\overline{p}_z, q_d; q)$$

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 based pipeline with some GATE components

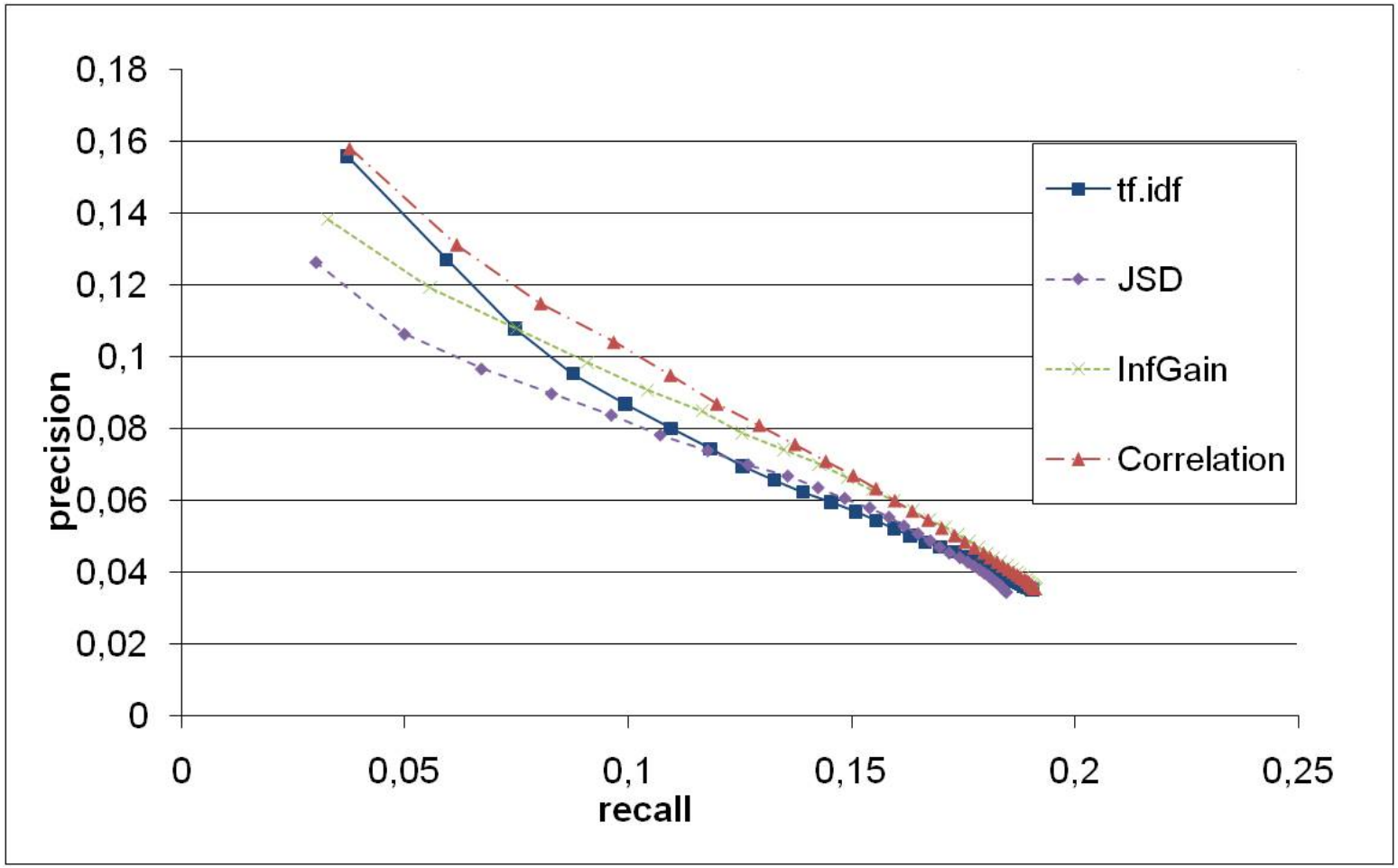
Multiword detection

- 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

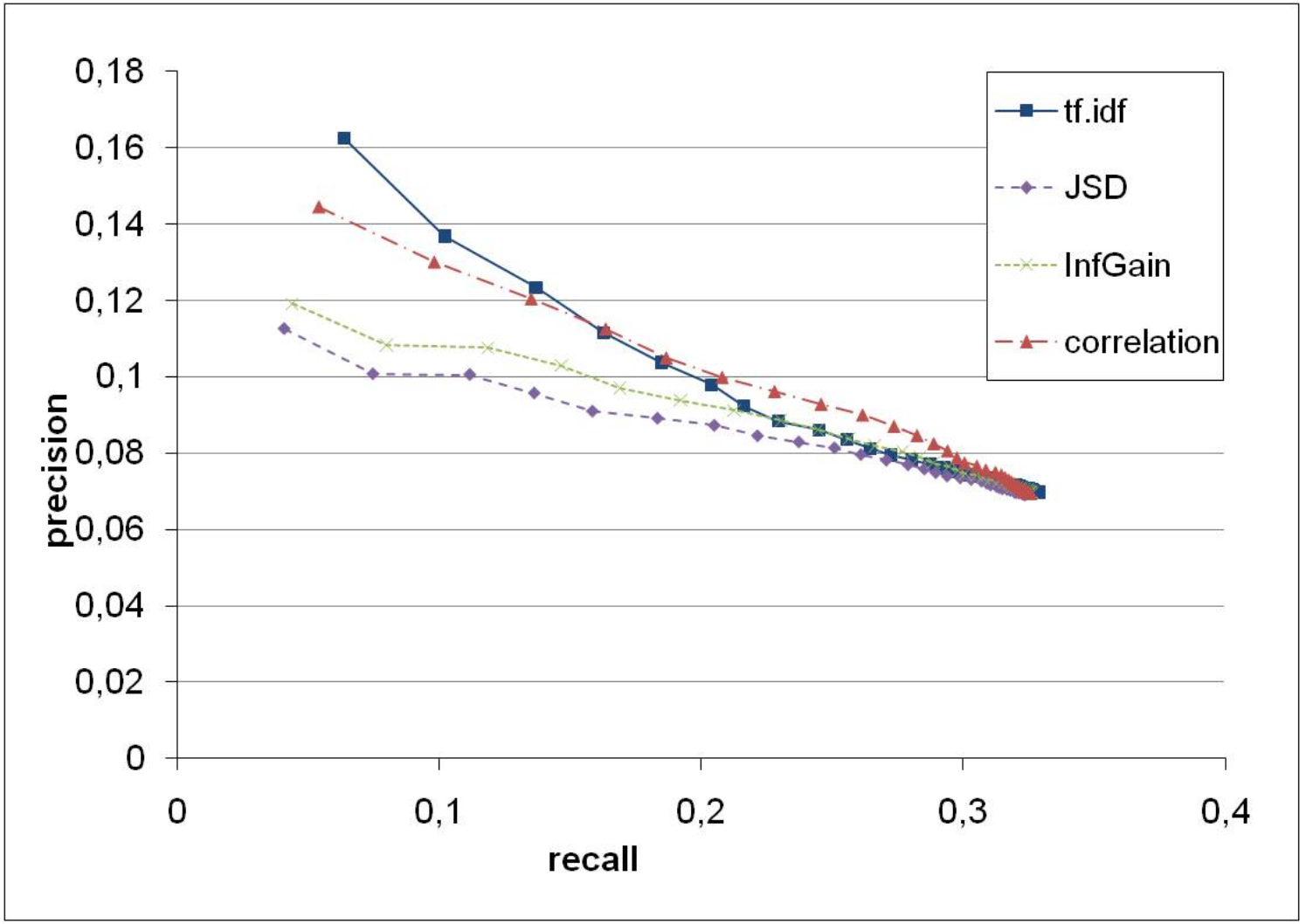
Evaluation BBC dataset

- 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-occurence data improves on tf-idf
- Slightly naive correlation coefficient works best.
- There is room for improvement
  - Christian Wartena has recently gotten good results for recommendation by using some clustering
  - Good multiword detection is really important.