C-value method for multi-word term extraction

Ismail Fahmi
i.fahmi@rug.nl
Alfa-informatica, RuG

Seminar in Statistics and Methodology
May 23, 2005

Agenda

- Example and Objective
- Terms vs Words
- Methods of Term Extraction
- C-value Method
- Experiment:
  - Corpus
  - Pre-processing
  - Filtering
  - Calculate C-value
- Evaluation
- Conclusion

Example

Given a sentence from the Elsevier's Medical Encyclopedia:

> Bij chronische myeloide leukemie bestaan de klachten meestal uit vermoeidheid (als gevolg van bloedarmoede) en een opgezette milt .
>
> In case of chronic myeloide leukaemia the complaints generally consist of fatigue (as a result of anaemia) and a swollen *milt* .

We want to extract terms from the sentence:

> *chronische myeloide leukemie,*
> *bloedarmoede*

Terms vs Words

- Terms:
  - ... refer to a defined concept ... (ISO704).
  - represent a limited number of part-of-speeches: nouns, verbs, adjectives, and adverbs.
- Words:
  - have functions in general reference... (Sager, 1990).
  - represent all part-of-speeches.

Characteristics of Terms

Kageura et al. 1996:

- Unithood: the degree of strength or stability of syntagmatic combinations or collocation.
- Termhood: the degree to which a linguistic unit is related to domain-specific concepts.

Methods of Term Extraction (1)

- Linguistic approaches, using linguistic filters, e.g.:
  - *Noun + Noun*
  - *Adj + Noun*
  - *Noun + suffix (-ase, -in)*
  - *((Adj | N)+ | (((Adj | N)* (N Prep)?) (Adj | N)*)) N*

Methods of Term Extraction (2)

Statistical approaches:

- Unithood:
  - Association ratio (close to the mutual information)
  - Loglikelihood
  - extracting word bigrams
- Termhood:
  - compound-noun-based measure, Imp (Nakagawa, 1998)
  - *C-value* (Frantzi et al., 1996)

C-value Method (1)

$$\textit{C-value(a)} = \begin{cases} \log_2|a| \,.\, f(a) & \text{if } a \text{ is not nested,} \\ \log_2|a| \left(f(a) - \frac{1}{P(Ta)} \sum_{b \in Ta} f(b) \right) & \text{otherwise.} \end{cases}$$

a	= candidate term (e.g., *myeloide leukemie*)
b	= longer candidate terms (e.g., *chronische myeloide leukemie*)
$\|a\|$	= length of the candidate term (number of words)
$f(a)$	= frequency of occurrence of a in the corpus
Ta	= set of extracted candidate terms that contain a
$P(Ta)$	= number of candidate terms in Ta
$f(b)$	= frequency of occurrence of longer candidate term b in the corpus.

C-value method (2)

C-value method:

- assigns a **termhood** to a candidate string
- using statistical characteristics of the candidate string:
  - frequency of occurrence in the corpus
  - frequency as part of other longer candidate terms
  - number of these longer candidate terms
  - the length of the candidate string (words)

Furthermore, C-value method:

- enhances the common statistical measure (frequency of occurrence)
  - frequency of occurrence only best for high frequency terms
- makes the measure sentitive to a particular type of multi-word terms, the nested terms
  - *C-value* gives more weight to the nested terms

Experiment: Calculating *C-value*

Step 1: Tag the corpus

Step 2: Extract strings using a linguistic filter

Step 3: Calculate the *C-value*

Step 1: Tag the corpus (1)

- Linguistic filter needs a tagged (POS) corpus
- Use Brill Tagger or Alpino Parser
- Corpus:

  Elsevier's Medical Encyclopedia

  21000 sentences

  367239 words

Step 1: Tag the corpus (2)

Example of a record in the corpus:

```
<med>
  <tw>leukemie</tw>
  <r>genhe</r>
  <bt>
    <media>
      <nua>ME256</nua> <kolbr>1</kolbr>
      ...
    </media>
    ...
    <ka>Symptomen</ka>
    <a>De verschijnselen van de acute vormen lijken veel op
    elkaar. Moeheid, bleekheid (als gevolg van bloedarmoede), koorts,
    keelontsteking , spontane blauwe plekken, lymfeklierzwellingen en
    vergroting van lever en milt zijn de voornaamste verschijnselen.
    Bij chronische leukemie is het begin van de ziekte vaak sluipend en
    duurt het vaak lang voordat de pati&euml;nt klachten krijgt. Bij
    chronische myeloide leukemie bestaan de klachten meestal uit
    vermoeidheid (als gevolg van <vw><st></st>bloedarmoede</vw>) en een
    opgezette milt. Bij chronische lymfatische leukemie kunnen tevens
    de lymfeklieren opgezet zijn en  treden er vaak infecties op.</a>
    ...
  <bt>
</med>
```

The record is tagged in an XML structure.

Some terms are already annotated with some defined tags, e.g., **vw** and **st**.

a = paragraph (to be extracted for further steps)

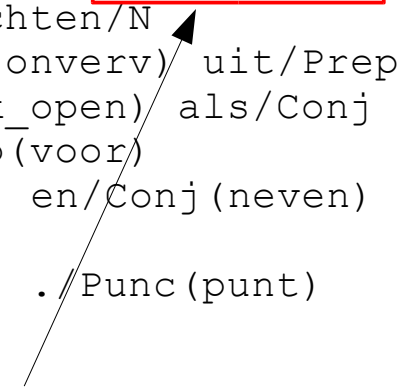
Step 1: Tag the corpus (3)

Clean up the sentences and tokenize, for example:

MEDENC-12227|**Bij chronische myeloide leukemie bestaan de klachten meestal uit vermoeidheid (als gevolg van bloedarmoede) en een opgezette milt** .

Tag the sentence using Brill Tagger (Drenth, 1997):

MEDENC-12227|Bij/Prep(voor) **chronische/Adj**(attr,stell,verv_neut) **myeloide/N**(soort,ev,neut) **leukemie/N**(soort,ev,neut) bestaan/N (soort,ev,neut) de/Art(bep,zijd_of_mv,neut) klachten/N (soort,mv,neut) meestal/Adv(gew,geen_func,stell,onverv) uit/Prep (voor) vermoeidheid/N(soort,ev,neut) (/Punc(haak_open) als/Conj (onder,met_fin) gevolg/N(soort,ev,neut) van/Prep(voor) **bloedarmoede/N**(soort,ev,neut))/Punc(haak_sluit) en/Conj(neven) een/Art(onbep,zijd_of_onzijd,neut) opgezette/V (trans,verl_dw,verv_neut) milt/V(trans,ott,3,ev) ./Punc(punt)



Step 2: Extract candidate strings (1)

- extract strings of maximum length,
- after we do not find any string of this length, decrease the length by 1 and make a new attempt,
- remove the word tags,
- create a list of strings of each length with their frequency of occurrence,
- concatenate the lists; the longest string appear at the top.

Step 2: Extract candidate strings (2)

Filter used in this experiment (Justeson, Katz, 1995):

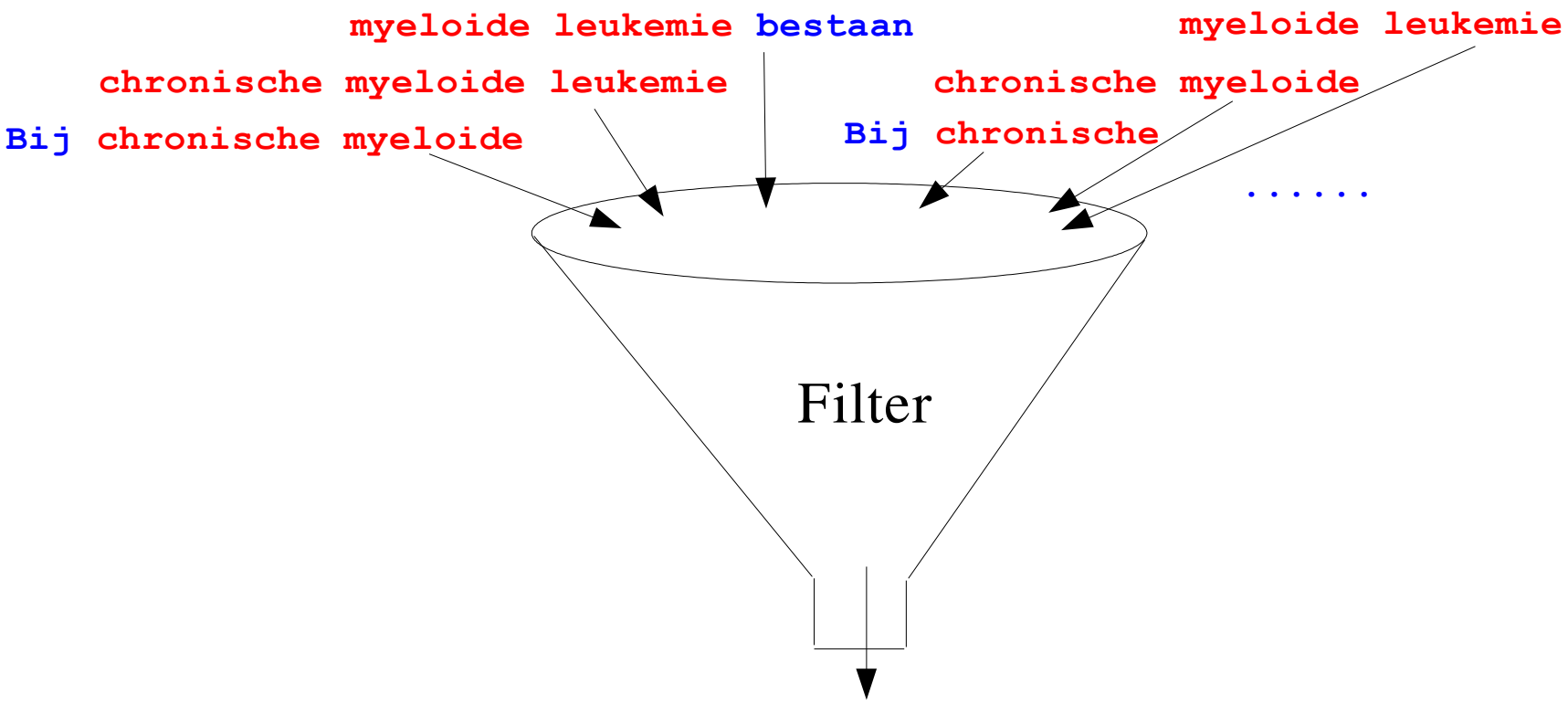
((Adj | N)+ | (((Adj | N) (N Prep)?) (Adj | N)*)) N*

Implementation of the filter in a Perl extended regular expression:

```
m{
   ((?:(?:(?:(?:(?:
        (?:\w+ \/ Adj) | (?:\w+ \/ N)
       )\s
     )+
     |
     (?:(?:(?:(?:
          (?:\w+ \/ Adj) | (?:\w+ \/ N)
        )\s
      )*
      (?: (?:\w+ \/ N) \s (?:\w+ \/ Prep) \s )?
     )
     ( (?:\w+ \/ Adj) | (?:\w+ \/ N) )*
    ))))
    \w+ \/ N
  )
 }xg
```

Step 2: Extract candidate strings (3)

"Bij chronische myeloide leukemie bestaan de klachten..."



Step 2: Extract candidate strings (4)

Candidate strings extracted from the example:

```
MEDENC-12227|chronische/Adj myeloide/N leukemie/N bestaan/N
MEDENC-12227|chronische/Adj myeloide/N leukemie/N
MEDENC-12227|myeloide/N leukemie/N bestaan/N
MEDENC-12227|gevolg/N van/Prep bloedarmoede/N
MEDENC-12227|chronische/Adj myeloide/N
MEDENC-12227|myeloide/N leukemie/N
MEDENC-12227|leukemie/N bestaan/N
MEDENC-12227|myeloide/N
MEDENC-12227|leukemie/N
MEDENC-12227|bestaan/N
MEDENC-12227|klachten/N
MEDENC-12227|vermoeidheid/N
MEDENC-12227|gevolg/N
MEDENC-12227|bloedarmoede/N
```

Step 2: Extract candidate strings (5)

Remove the word tags and create a list of the candidate strings with their frequency of occurrence, sort by their length (descending).

A list of candidate strings containing the word myeloide

Freq	Candidate string
1	CHRONISCHE MYELOIDE LEUKEMIE BESTAAN
4	CHRONISCHE MYELOIDE LEUKEMIE
2	ACUTE MYELOIDE LEUKEMIE
1	MYELOIDE LEUKEMIE BESTAAN
8	MYELOIDE LEUKEMIE
4	CHRONISCHE MYELOIDE
2	ACUTE MYELOIDE
8	MYELOIDE

Freq	# candidate strings
>= 3	1089
>= 2	2618
>= 1	17383

Step 3: Calculate the *C-value* (1)

Algorithm (Frantzi et al., 2000):

```
for all strings a of maximum length
    calculate C-value(a) = log2|a| . f(a);
    if C-value(a)>= Threshold
        add a to output list;
        for all substrings b
            revise t(b);
            revise c(b);

for all smaller strings a in descending order
    if a apperas for the first time
        C-value(a) = log2|a| . f(a)
    else
        C-value(a) = log2|a| . (f(a) - 1/c(a) . t(a))
    if C-value(a) >= Threshold
        add a to output list;
        for all substrings b
            revise t(b);
            revise c(b);
```

Step 3: Calculate the *C-value* (2)

Start with the longest string, e.g.:

```
   a = CHRONISCHE MYELOIDE LEUKEMIE BESTAAN
 |a| = 4
f(a) = 1
```

C-value(CHRONISCHE MYELOIDE LEUKEMIE BESTAAN) = $\log_2|a|.f(a)$

$= \log_2 4 \,.\, 1$

$= \mathbf{2}$

Step 3: Calculate the *C-value* (3)

For a nested string, e.g.:

```
   a = CHRONISCHE MYELOIDE LEUKEMIE
 |a| = 3
f(a) = 4
```

Appear in a longer string:

```
   b = CHRONISCHE MYELOIDE LEUKEMIE BESTAAN
f(b) = 1
P(Ta)= 1 (number of different longer strings)
```

C-value(CHRONISCHE MYELOIDE LEUKEMIE) = $\log_2|a|(f(a)-\frac{1}{P(Ta)}\sum_{b\epsilon Ta} f(b))$

= $\log_2 3$. (4 - 1/1)

= **4.75**

Step 3: Calculate the *C-value* (4)

How to determine that myeloide leukemie is a term, while acute myeloide is not?

```
   a = MYELOIDE LEUKEMIE
 |a| = 2
f(a) = 8
```

is a substring of some longer strings:

```
f(b):
 1        CHRONISCHE MYELOIDE LEUKEMIE BESTAAN
 4        CHRONISCHE MYELOIDE LEUKEMIE
 2        ACUTE MYELOIDE LEUKEMIE
 1        MYELOIDE LEUKEMIE BESTAAN
--------------------------------
Σf(b) = 8            P(Ta) = 4
```

C-value(MYELOIDE LEUKEMIE) = $\log_2|a|(f(a) - \frac{1}{P(Ta)} \sum_{b \in Ta} f(b))$

= $\log_2 2$. (8 - 8/4)

= **6**

Step 3: Calculate the *C-value* (5)

Now, we calculate the C-value for acute myeloide

```
   a = ACUTE MYELOIDE
 |a| = 2
f(a) = 2
```

is a substring of a longer string:

```
f(b):
 2        ACUTE MYELOIDE LEUKEMIE

Σf(b) = 2 (total frequency of the longer strings)
P(Ta) = 1 (number of different longer strings)
```

C-value(ACUTE MYELOIDE) = $\log_2|a|(f(a)-\frac{1}{P(Ta)}\sum_{b\epsilon Ta} f(b))$

= $\log_2 2$. (2 - 2/1)

= **0**

Step 3: Calculate the *C-value* (6)

Acute myeloide only occurs as a substring of one longer term. Its high degreee of dependence to a term gives negative effect to its termhood.

Myeloide leukemie occurs as a substring of many longer terms. This condition shows its high degree of independence, which then decreases the negative effect of being a substring.

Intuition:

- The more often a candidate term occurs alone and longer its size, the higher its termhood.
- The more often a candidate term occurs as a substring, the lower its termhood.
- But, the more the number of longer terms in which the candidate term occurs, the higher its termhood.

Step 3: Calculate the *C-value* (7)

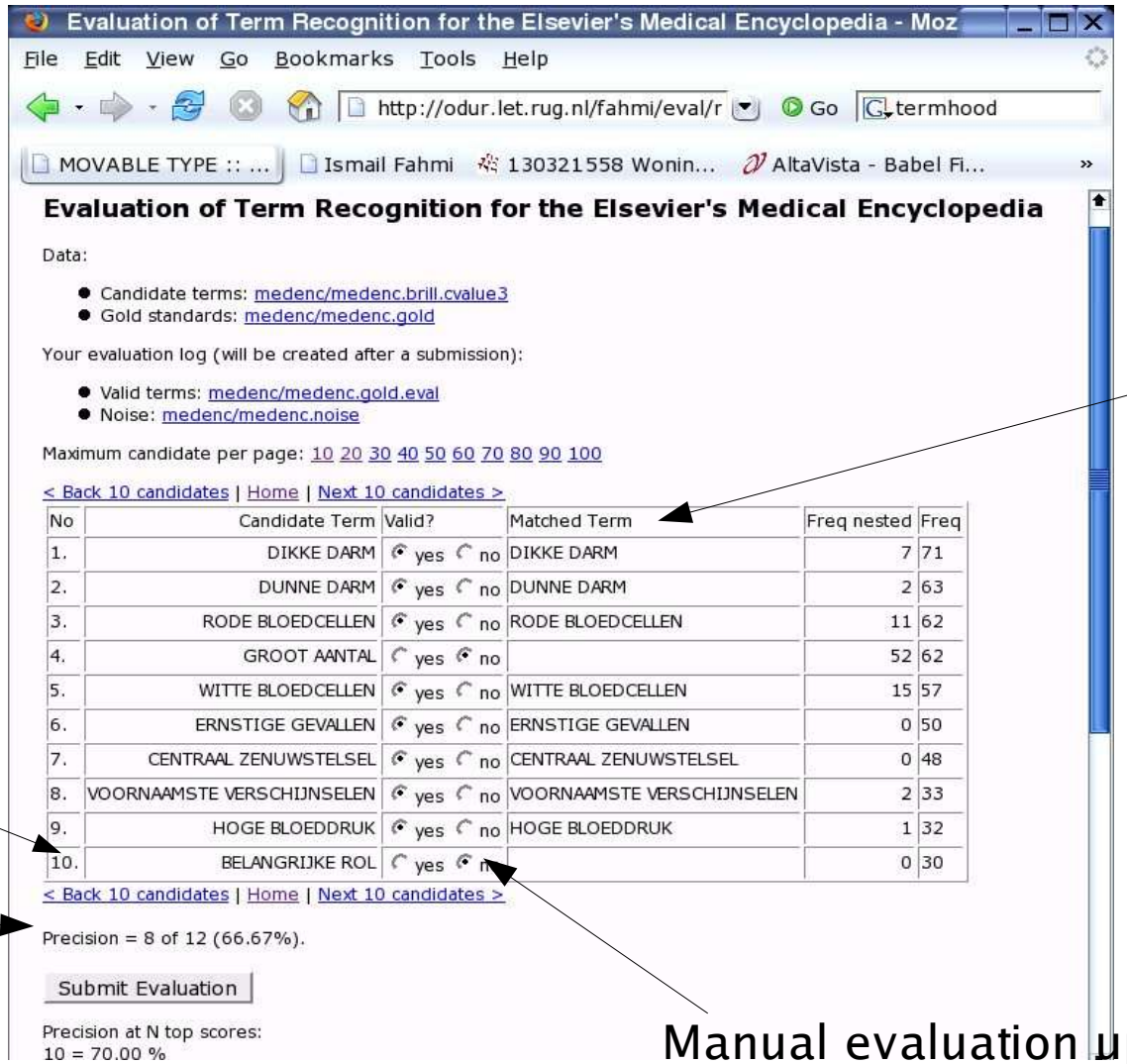
The C-value of terms containing the word myeloide

Rank	C-val	P(Ta)	f(b)	f(a)	Candidate Term
267.	6.00	4	8	8	MYELOIDE LEUKEMIE
392.	4.75	1	1	4	CHRONISCHE MYELOIDE LEUKEMIE
607.	3.17	0	0	2	ACUTE MYELOIDE LEUKEMIE
2397.	2.00	0	0	1	CHRONISCHE MYELOIDE LEUKEMIE BESTAAN
4485.	1.50	2	5	4	CHRONISCHE MYELOIDE
15940.	0.00	1	2	2	ACUTE MYELOIDE
16882.	0.00	1	1	1	MYELOIDE LEUKEMIE BESTAAN

The C-value of the top 10 candidate terms in the rank

Rank	C-val	P(Ta)	f(b)	f(a)	Candidate Term
1.	69.83	6	7	71	DIKKE DARM
2.	62.00	2	2	63	DUNNE DARM
3.	60.78	9	11	62	RODE BLOEDCELLEN
4.	60.70	40	52	62	GROOT AANTAL
5.	55.50	10	15	57	WITTE BLOEDCELLEN
6.	50.00	0	0	50	ERNSTIGE GEVALLEN
7.	48.00	0	0	48	CENTRAAL ZENUWSTELSEL
8.	31.00	1	2	33	VOORNAAMSTE VERSCHIJNSELEN
9.	31.00	1	1	32	HOGE BLOEDDRUK
10.	30.00	0	0	30	BELANGRIJKE ROL

Method of Evaluation



Gold standard
4950 terms

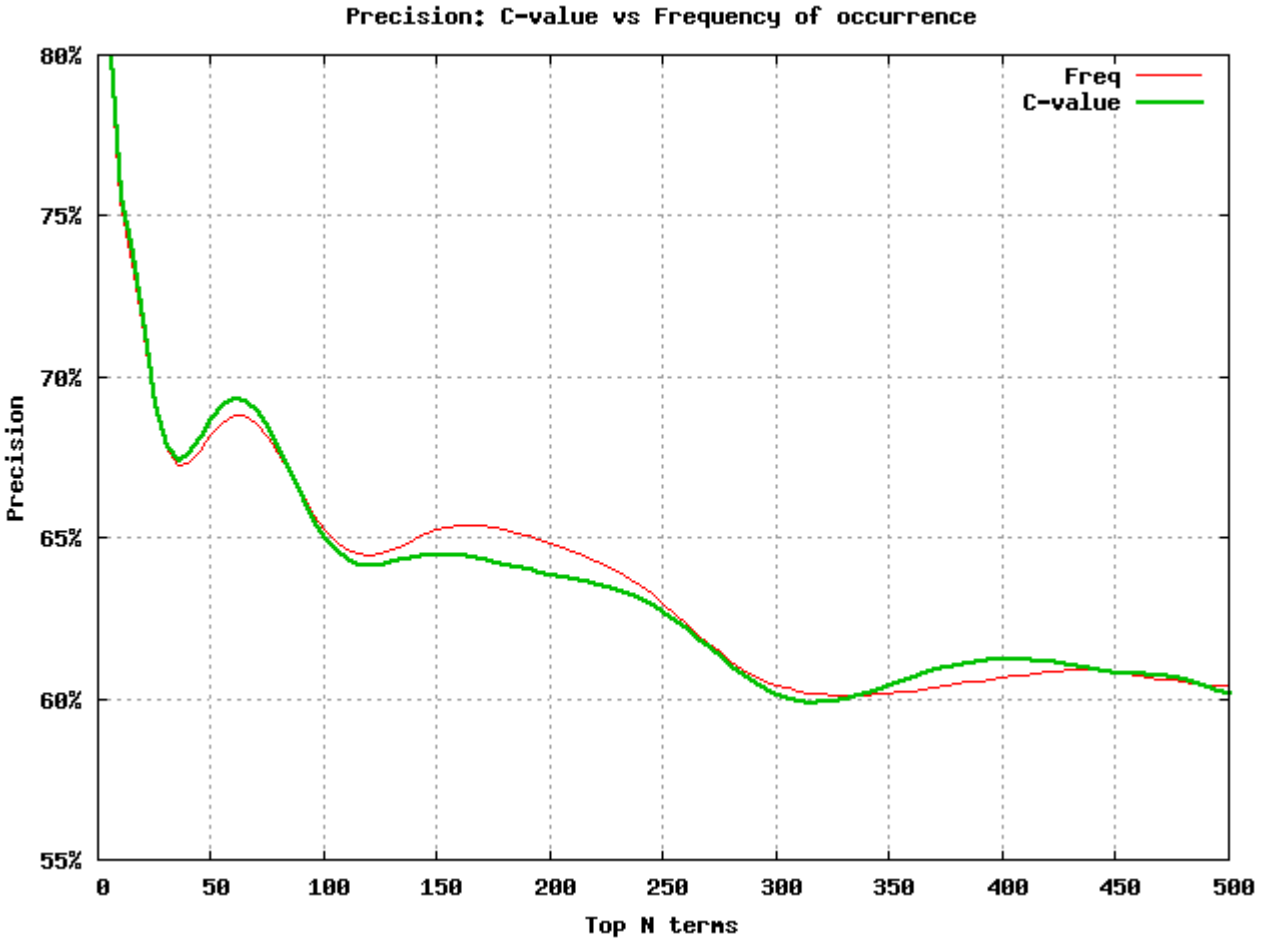
2638
candidate
terms

Precision

Manual evaluation up to
top 500 candidate terms
(by Lonneke)

Evaluation

Precision for 500 candidate terms by C-value and baseline (frequency)

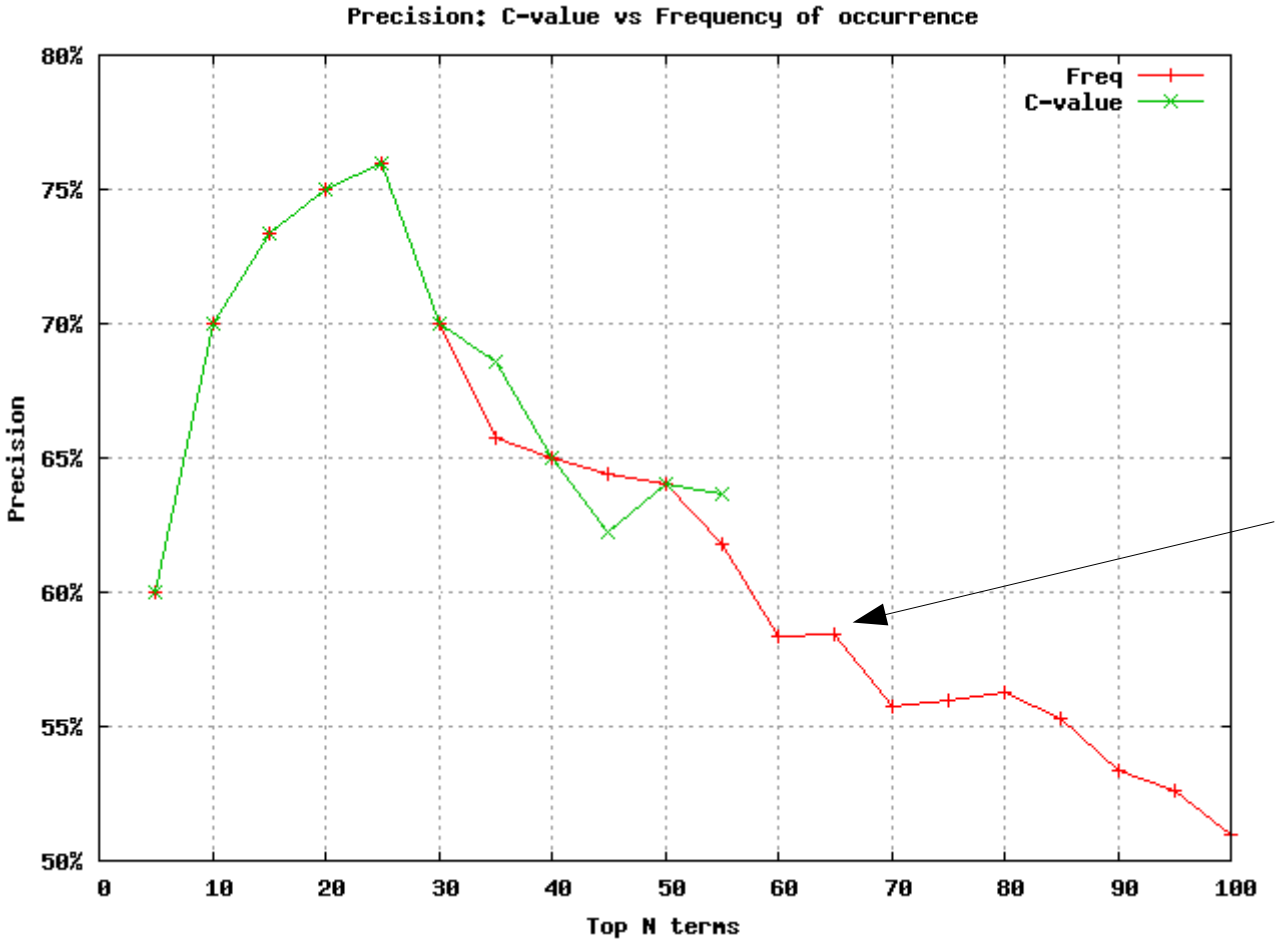


Majority of the candidate terms have never been seen as nested.

Therefore, C-value treats them in a similar way to frequency.

Evaluation (2)

Precision for 50 candidate terms that appeared as nested (substring)



Number of nested candidate terms: C-value, 58; frequency, 117.

The differences were assigned to 0 by the C-value.

Evaluation (3)

Problem with the C-value in this experiment:

C-val	P(Ta)	f(b)	f(a)	Candidate term
6.00	0	0	3	**FAMILIAIRE ADENOMATEUZE POLYPOSIS COLI**
1.00	2	6	4	POLYPOSIS COLI
0.00	1	3	3	FAMILIAIRE ADENOMATEUZE POLYPOSIS
0.00	3	9	3	ADENOMATEUZE POLYPOSIS
0.00	1	3	3	**ADENOMATEUZE POLYPOSIS COLI**

A string which occurs in only one longer string will be assigned to 0 by the C-value, although it is a term.

Solution (future work): Incorporate context information, by using part of speech elements of modifiers which appear before or after the candidate strings.

Conclusion

- The linguistic filter is very important because it contributes in the very early selection of candidate terms.
- Most of the candidate terms from the corpus are never seen as nested. Only 117 of 2638 are nested.
- Therefore C-value treated them in a similar way to frequency.
- C-value kicks out nested substrings. These candidate strings decrease the percision of the frequency.
- The corpus (encyclopedia) is well structured. Results on other corpora can not be claimed to be better or worse before we conduct experiments.