Structured Content-Only Information Retrieval Using Term Proximity and Propagation of Title Terms

Michel Beigbeder

Structured Content-Only Information Retrieval Using Term Proximity and Propagation of Title Terms

Michel Beigbeder

École Nationale Supérieure des Mines de Saint-Étienne
michel.beigbeder@emse.fr

Abstract. Our experiments in the 2006 INEX ad'hoc track were based on the use of the proximity of the query terms in the documents to rank them. More precisely we define around each occurence of a query term an influence function. For an occurence appearing in the text itself, this influence function is linearly decreasing from 1 to 0 depending on the distance to the occurence. When a query term happens to appear in a title of a structured document its influence is uniformly 1 from the beginning to the end of the (sub-)section. We use boolean queries and these influence functions are combined according to the tree of a query using fuzzy logic. The score of any part of a document is the summation of the resulting influence function at the root of the query tree on the range of this part. We present and comment the results.

1 Introduction

The needs for information retrieval are now quite well established and the tools have a large acceptance from the users. Though quite every documents are created with some structure in mind, the methods and tools are mainly dedicated to flat documents as opposed to structured documents.

Moreover most of the methods used for information retrieval on flat texts don't even take into account the basic structure of text: its linearity. In fact they are based on frequencies of terms (both in the documents and in the collection) and on the document lengths. Though there were some attempts to use the position of word occurences in the text with either explicit proximity operators in the query language or ranking based on proximity of the query terms. These attempts are reviewed in section 2.

Concerning the logical structure which is the structure commonly refered to when speaking about structured documents, it is only quite recently that a sufficiently widespread representation for it is available so that large corpora of structured documents are available. So it is now possible to experiment in the large some of the ideas developped for structured information retrieval in the past and to design new methods.

We present in this paper an extension to structured documents retrieval of a proximity based method originally dedicated to flat texts. Our model can easily

compute a score for any segment of text, in particular for any section or the whole document. First in section 3, we present the document model this method deals with, and in section 4 the method itself. In section 5 we present the experiments made within the INEX 2006 campaign.

2 Proximity Use in Flat Document Retrieval

The idea of using the proximity of the query keywords for retrieving flat documents was first implemented in boolean systems with a NEAR operator. This operator itself was an extension of the ADJ operator. These two operators can be used between keywords in a boolean query and its truth value is related to the positions of the two connected keywords. The NEAR operator evaluates to true if the two terms appear within k words of each other (k is *one* for the ADJ operator).

The motivation for the ADJ (resp. NEAR) operator is to be able to describe in the query the needs for phrases (resp. loose phrases). These operators still are in use in tools used for searching in library catalogs. Though, from a technical point of view, they suffer from two handicaps that slowed down their use in plain text search engines. The first one is that they are closely linked to the boolean retrieval model which does not allow to rank the retrieved documents. The second one is that they do not fit well in the boolean query language model itself because they can only connect keywords and cannot be consistently extended to connect boolean sub-expressions.

More recent ideas for using keyword proximity were developped and they don't have these two limitations. Concerning the second one, the queries accepted by the query language model are either bags of terms or classic boolean expressions (only AND and OR operators). About the first one all the methods score the documents with respect to the positions of the keywords occurences, taking into account their proximity. We will now describe the basis of some of these methods

2.1 Interval Based Methods

For their participation to the TREC-4 campaign, both Clarke *and al.* [1] and Hawking *and al.* [2] developped similar methods to rank text documents according to the proximity of the query keywords. The ideas are to select some intervals of text that contain all the keywords; to attribute a score to these intervals (the shorter the interval, the greater the score) and to sum up all these scores to score the document.

The two methods differ in the selected intervals: for Clarke *and al.* intervals cannot be nested because only the shortest ones are selected. For Hawking *and al.*, for each occurence of any of the keywords, the shortest interval that contains all the keywords is selected. So if there are two successive occurences of the same keyword without any occurences of any other keyword in between, two nested intervals are selected.

The two methods also differ in the interval scoring, Clarke *and al.* chose a score that is roughly inversely proportional to the interval length and Hawking

and al. chose a score roughly inversely proportional to the square root of the interval length.

The idea of using intervals was then revisited by Rasolofo *and al.* [3]. They chose to base their method on *Okapi* and they add an additional score to the Okapi probability. This additional score is based on the intervals containing any query terms pair: Each of the intervals shorter than a specified constant (6 in their experiments) that contains occurences of two query terms contribute to this additional score.

2.2 Fuzzy Influence Function Model

Beigbeder *et al.* [4] developped a retrieval model based on *the fuzzy proximity of the keywords*. More precisely each occurence of a keyword has a fuzzy influence on its neighbouring. This influence reaches its maximum value *one* at the keyword occurence position and decreases with the distance to this position. The most simple function that have this behaviour is a triangle function. Moreover there is an easy way to define a control parameter in such a function: its width, the length of the triangle basis. We will call k half of this length, it controls the range of the influence of an occurence.

Given a term the influences of its occurences are combined with a maximum operator. If the influence function is symmetrical, it consists in considering that at a given position the influence is determined by the nearest occurence of the term.

Their query language model is that of the classical boolean model with AND, OR and NOT operators (neither NEAR nor ADJ). The influences of the query terms are combined in the query tree according to the fuzzy logic interpretation of the union and intersection operators

Let us consider an example with the document `X X X X A X X X B X X X X X` where there is an occurence of the term A (resp. B) at position 5 (resp. 9) and where X denotes any term different from the terms A and B. Figure 1 shows the proximities to the terms A and B in this sample document (with $k = 5$) and their combination with a minimum corresponding to the AND operator.

Finally the score of a document is the summation of the influence function over all the positions in the text. It consists in evaluating the area under the curve associated to the root of the query tree. With our example, this is the area under the triangle of the curve A AND B.

This is this model that we extended to some kind of structured documents.

3 Our Model of Structured Documents

Our work is pragmatic with respect to the structure of documents. We want to take into account the basic structure of many kinds of document models: nested sectionning and titles. This is the basis for scientific articles and technical documents but also for many more informal documents. We ignore any other structure, such as lists and emphasis for instance. As a particular case, we consider that a document is the highest level in the sectionning hierarchy.

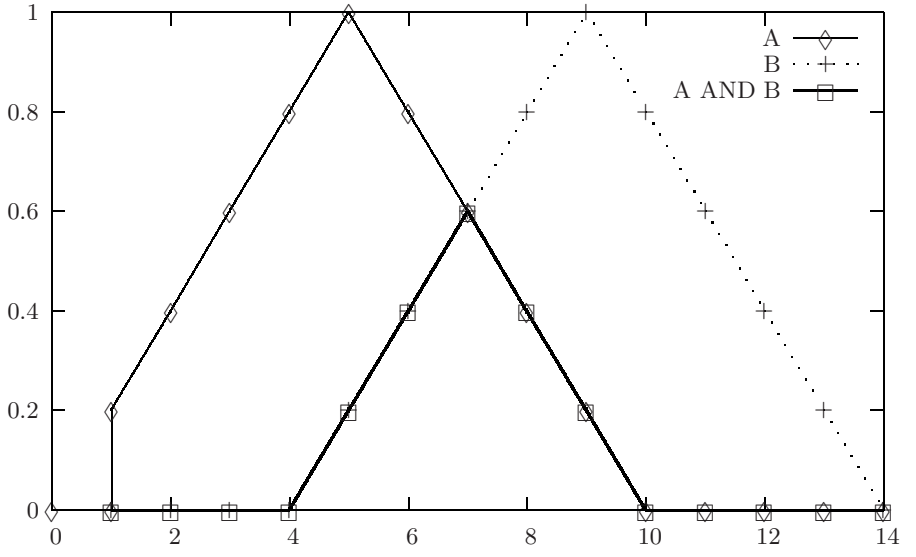

Fig. 1. Proximities to the terms A and B and their combination for the query A AND B: *x-axis:* position of words in the text, *y-axis:* fuzzy proximities

Another point is that sectionning and titles are tightly related so that in the LaTeX styles, only sectionning commands (`\section`, `\subsection`, ...) are available and the titles are given as parameters to these commands.

So the basis for our document model is the family of document which could be coded in the LaTeX styles with the sectionning commands only. Here is an example:

```
\title{title 1}           % highest level, level 0
                          %    the document level and its title
bla bla                   % level 0 text
  \section{title 2}       % level 1 and its title
  bla bla                 % level 1 text
    \subsection{title 3}  % level 2 and its title
    bla bla               % level 2 text
  \section{title 4}       % level 1 and its title
  bla bla                 % level 1 text
```

This example can be coded in XML with:

```
<section><title>title 1</title>
bla bla
  <section><title>title 2</title>
  bla bla
    <section><title>title 3</title>
    bla bla
    </section>
  </section>
```

```
  <section><title>title 4</title>
  bla bla
  </section>
</section>
```

Formally the grammar of our document model is:

```
document        = section
section         = '<section>''<title>' title_text '</title>'
                     section_content
                  '</section>'
section_content = (section_text | section)*
```

4 Influence of Keywords Occurences

In the model presented in section 2.2, the influence of a term was only modelized for linear text. With our model of structured document, we have to modelize the influence of an occurence of a query term depending on the structural part in which this occurence does appear. As our document model is very simple there are only two cases: Occurences can appear in *section_text* parts or *title_text* parts.

For a term occurence which appears in the *section_text* parts, the basis is the same as in linear text: A decreasing value of the distance to the occurence. But we add another constraint, the influence is limited to the *section_text* part in which the occurence does appear.

Let us consider a document with the same text than the sample document of section 2.2 but with some structural tags: `<section> <title> X X X X A </title> X X X B X X X X X </section>`. The occurence of the term B is in the *section_text* part of the section. Figure 2 shows the limitation of the triangle proximity to the term B in the document to the surrounding section.

For term occurences which appear in the *title_text* parts their influence is extended to the full content of the section and recursively the subsections contained in the corresponding *section_content* part.

Considering our sample structured document the occurence of the term A is in the *title_text* part of the section. Figure 2 shows the propagation of the influence of the occurence of the term A that appears in the title to the whole section.

Otherwise, like in the model presented in section 2.2, we use a boolean language query model and we combine the influence functions with `min` and `max` operators on the internal nodes of the boolean query tree. The basic score of a section is the summation of the influence function at the root of the query tree. We normalize this score by the maximum score reachable for this section. As the maximum value of the influence function is *one*. The maximum score simply is its length. Note that this maximum can actually be reached, for instance if all the query terms appear in the title of a section.

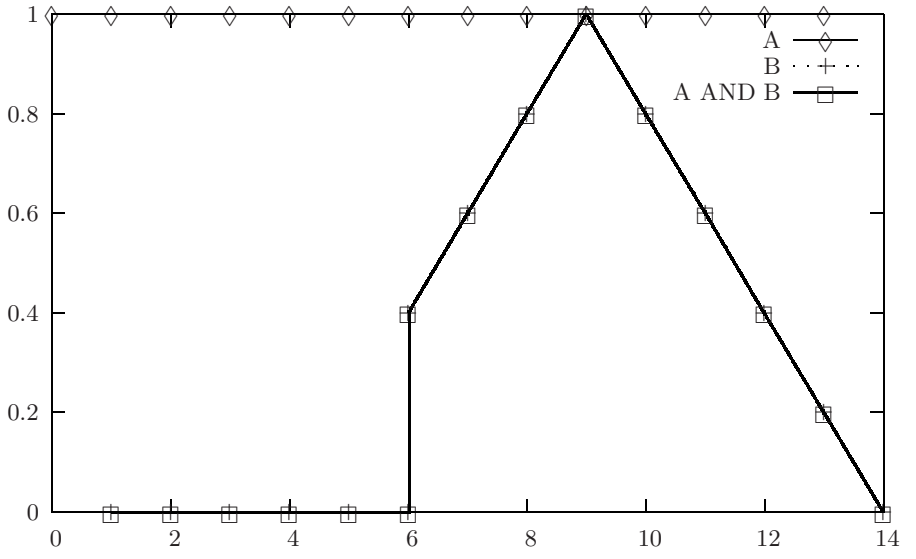

Fig. 2. Limitation of the influence of an occurence to the *section_text* part in which it appears (proximity to the term B); propagation of a title term (proximity to the term A): *x-axis:* position in the text, *y-axis:* fuzzy proximities

5 Experiments and Implementation

5.1 Converting the Documents to Our Document Model

The documents of the Wikipedia collection used for the 2006 INEX campaign are written in XML. But the structure of the documents is more complex than that of our document model of section 3, as 1056 different tags are used.

The main part of the conversion consists in keeping the text and the section and title tags with their corresponding closing tags. This can easily be done with an `xslt` processor, but some non obvious choices have to be made about the textual content, particularly concerning the spaces. Unfortunately, at the syntactic level no right choice can be made because of an inconsistent use of some tags. For instance, the document number 1341796 contains the following highligth (the attributes of the `collectionlink` tags are removed):

```
This is a
<emph3>List of
<collectionlink ...>poison</collectionlink>ings</emph3>in
alphabetical order of victim. It also includes confirmed attempted
and fictional poisonings. Many of the people listed here committed
or attempted to commit
<collectionlink ...>suicide</collectionlink>by
poison;
others were poisoned by others.
```

The question is to insert or not a space after the closing of the `collectionlink` tag. If a space is inserted, the following text is generated (mistake is emphasized):

> `This is a List of` *`poison ings`* `in alphabetical order of victim. It also includes confirmed attempted and fictional poisonings. Many of the people listed here committed or attempted to commit suicide by poison; others were poisoned by others.`

which is correct for the second instance of the tag, but not for the first one. If no space are inserted, the following text is generated:

> `This is a List of poisonings in alphabetical order of victim. It also includes confirmed attempted and fictional poisonings. Many of the people listed here committed or attempted to commit` *`suicideby`* `poison; others were poisoned by others.`

with the reverse correctness. Notice that a choice about spaces are to be made for each tag, and in the above examples, the `emph3` tags were replaced by spaces.

As no consistant choice could be made, we chose to insert space for each tag.

5.2 Indexation Tool and Index Structure

We used as a basis for the indexation the tool LUCY[1] in version 0.5.4. Though it is an outdated version that is now replaced by different versions of ZETTAIR[2], we had some experience with it as we extended it with an implementation of the model presented in section 2.2. It is a good basis because it keeps within its index the position of every occurence of every term, and its lexical analyzer can recognize the syntax for any XML tags. At the indexation phase, we added the code necessary to keep track of the position and the nesting of the `section` and `title` tags. Remember that all other tags were removed in the previous step when the documents were converted to our model.

5.3 Building the Queries

Queries could be automatically built with the conjunction of the terms that appear in the title field of the topics. As our method is highly selective, there would be very few results if any in the retrieved list of documents with such queries. So either the basic conjunctive queries or the retrieval procedure have to be relaxed in some way. Keeping these automatic conjunctive queries it is possible to enlarge the result set by using a lemmatization both in the indexation phase and in the query analysis. We didn't try this solution but we chose to build the queries manually.

With a basis of the conjunction of the terms found in the title field, sometimes, some terms were removed, but more often, these terms were expanded with disjunction of variations of the terms. These variations could simply be flexionnal

[1] `http://www.seg.rmit.edu.au/lucy/`

[2] `http://www.seg.rmit.edu.au/zettair/`

ones (plural vs. singular) or derivationnal ones (verb, noun, adjective) or even semantic ones (synonyms or related concepts).

For instance, the title field of the topic number 289 is

```
emperor "Napoleon I" Polish
```

With a simple conjonction, the query could be (the '&' symbol is used for the boolean AND operator):

```
emperor & Napoleon & I & Polish
```

But some relaxation of it can be derived, for instance:

```
emperor & Napoleon & Polish
Napoleon & Polish
Napoleon & (Polish | Poland)
```

By using the description, narrative and ontopic_keywords fields, other queries can be formulated, for instance:

```
Napoleon & (Polish | Poland | Laczynska | Malewski | Poniatowski)
```

We built two sets of queries, the short ones where we used only the title keywords with very minor expansion, and the enhanced queries for which we used terms from any field with expansions.

5.4 Runs

Given the queries and the value of the parameter k, our method is able to compute the fuzzy proximities to the query terms for each leaf of the query tree and to combine these influences up to the root of the tree. With our two query sets we used two values of k, 50 and 200. As only three runs could be submitted, we sent as official runs these combinations except the one with enhanced queries and $k = 50$.

Then a score can be computed for any span in the document. We computed scores for the whole documents and all its sections and subsections recursively. This was our participation to the THOROUGH task.

For the BEST IN CONTEXT task we searched for the maximum of the influence function at the root of the query tree and returned the section of highest level which contained the position at which this maximum was reached.

Finaly for the FOCUSED task, the different parts of the documents were sorted with two keys: first, the score of the document to which it belongs; and then its own score.

6 Results

Whatever the task, our best results were obtained with our set of enhanced queries. With the set of short queries, our two sets of results with $k = 50$ and $k = 200$ are very close one to each other and quite worse than with enhanced

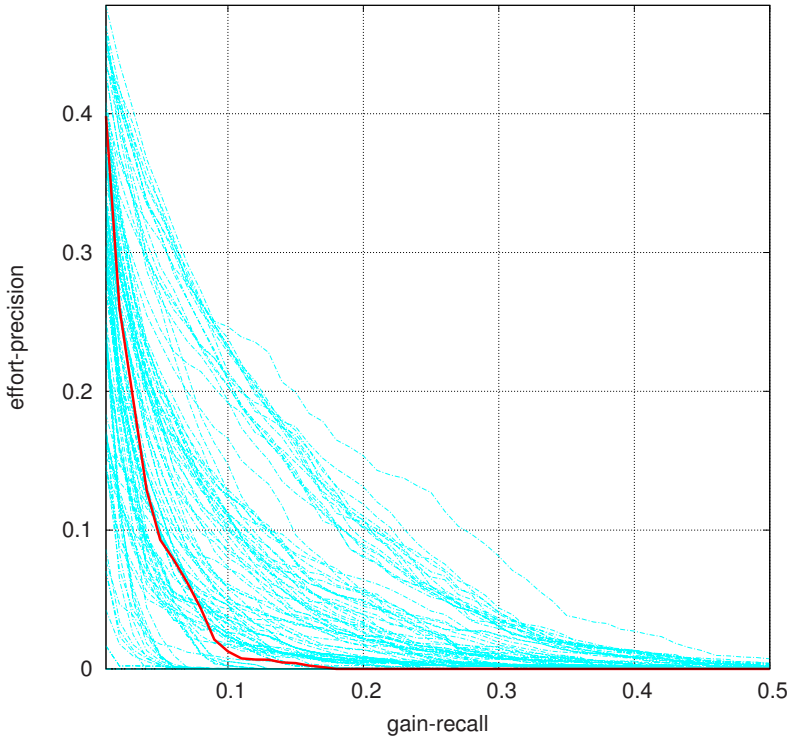

Fig. 3. INEX 2006: Results' Summary, Metric: ep-gr, Quantization: gen, Task: thorough, Run: title_Q_Prox200NT02

queries. This shows that the expansion mechanism is mandatory. More experiments should be done on the value of k to draw some conclusion. In the following all the results are related to the set of enhanced queries.

Figure 3 shows our results summarized with the ep-gr metric for the set of enhanced queries and $k = 200$. At the first level, the results are quite good but there is a rapid decrease in quality and it is very near from zero after 0.1. The same behaviour is seen in the results for the focused task either with overlap On (Fig. 4) or Off (Fig. 5). Moreover these figures display the result measure at 5 documents. When looking further in the list of results, our method compare less and less favourably to other methods. We develop some explanations about this remark in the following.

Our queries are conjunctions of the terms that a (part of a) document has to contain to have a non zero score. Moreover the occurences of the different terms must be close one to each other. As a result this method is highly selective. Table 1 displays the distribution of queries in regards to the length of the result list built by our method. This highlights the fact that our list of results are quite short. The length of the lists are longer as k increases — which is trivial — and longer also with the enhanced queries as most of them relax the constraints on

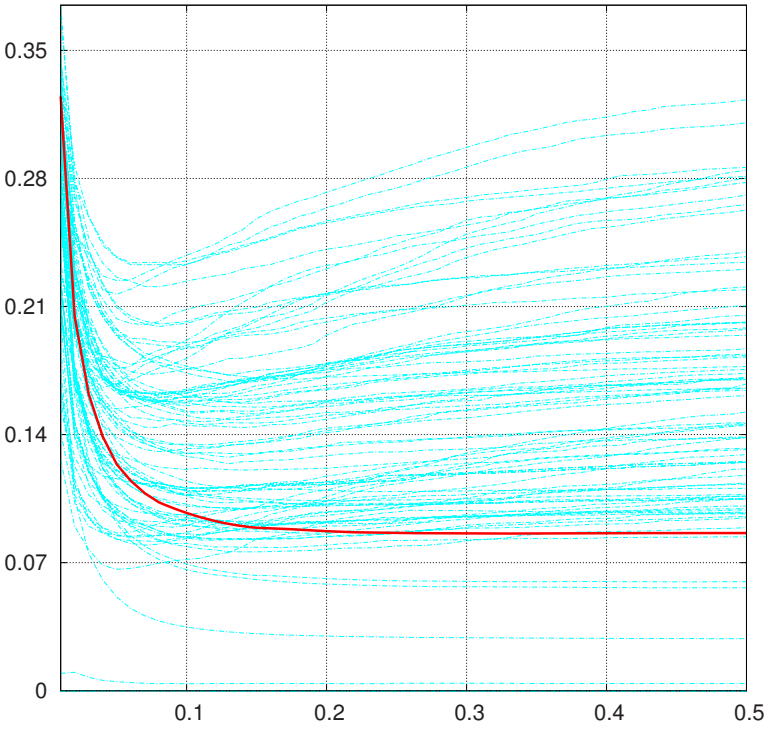

Fig. 4. INEX 2006: Results' Summary, Metric: nxCG[5], Quantization: gen, Task: focused, Overlap=On, Run: title_Q_Prox200NF02

the retrieved documents. Also it can be seen that the very large majority of our result lists are much shorter than the limit of 1500 imposed by INEX.

It seems though that we obtain a quite good precision. Our summarized results were also disavantaged because of our simplification of the structure. In the thorough task we only returned elements tagged as sections and did not add higher level elements when one section was retrieved.

Finally our propagation mechanism was disturbed by strange documents. For instance document number 192509 contains many text in lists between the `<title>` and `</title>` tags. This results with our structure simplification with a very long title whose terms are propagated to the whole title and its associated section. If it happens that all the query terms appear in this erroneous title the section reach a maximal score of 1. Besides this "document bug", it highlights a drawback of our method.

7 Conclusion

We presented in this paper the ideas used for our participation to the INEX 2006 campaign in the ad'hoc retrieval task. Our method is based on the proximity of

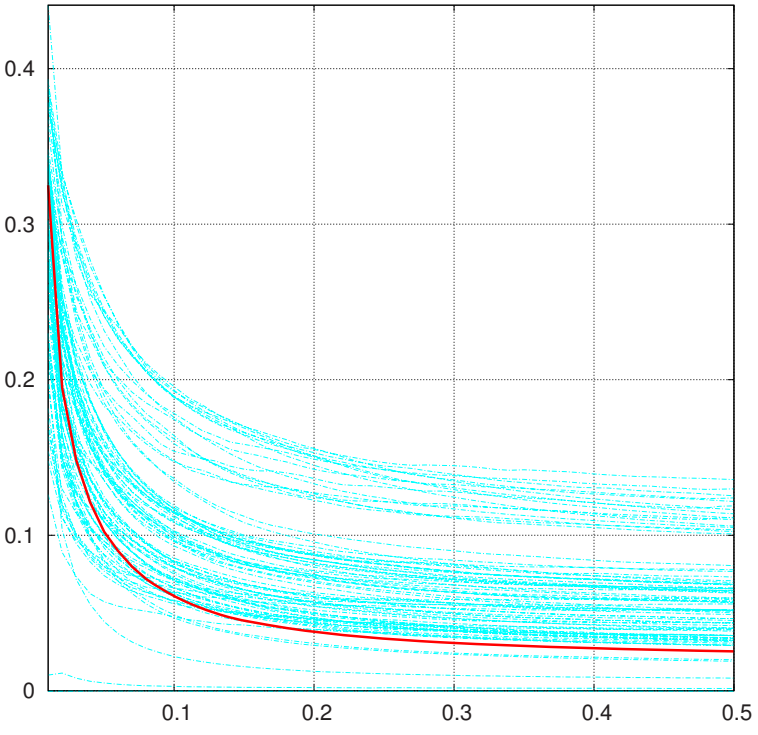

Fig. 5. INEX 2006: Results' Summary, Metric: nxCG[5], Quantization: gen, Task: focused, Overlap=Off, Run: title_Q_Prox200NF02

Table 1. Distribution of queries as a function of their result list length

	No result	1 to 10	10 to 100	100 to 1000	more than 1000
Thorough k=50 short	13	21	50	37	4
Thorough k=200 short	7	20	46	44	8
Thorough k=200 enhanced		7	53	55	10
Foc./Best k=50 short	13	40	48	22	2
Foc./Best k=200 short	7	30	60	26	2
Foc./Best k=200 enhanced		15	75	32	3

the keywords in the document and the propagation of the title words to the whole associated section. We obtained quite good precision results. The summarization done by the official measure was at our disadvantage because our result lists were very short and because we simplified the structure so we did not return every kind of elements. We also found that this method needs query expansion mechanism. Further works are to be done to study the best setting for the parameter k which controls the range of influence of a term in the text. Also some work has to be done to improve the propagation mechanism of title words which is very crude at the time.

References

1. Clarke, C.L.A., Cormack, G.V., Burkowski, F.J.: Shortest substring ranking (multitext experiments for TREC-4). [5]
2. Hawking, D., Thistlewaite, P.: Proximity operators - so near and yet so far. [5]
3. Rasolofo, Y., Savoy, J.: Term proximity scoring for keyword-based retrieval systems. In: Sebastiani, F. (ed.) ECIR 2003. LNCS, vol. 2633, pp. 207–218. Springer, Heidelberg (2003)
4. Beigbeder, M., Mercier, A.: An information retrieval model using the fuzzy proximity degree of term occurences. In: Haddad, H., Liebrock, L.M., Omicini, A., Wainwright, R.L. (eds.) SAC, pp. 1018–1022. ACM Press, New York (2005)
5. Harman, D.K. (ed.): In: Harman, D.K., (ed.) The Fourth Text REtrieval Conference (TREC-4), Department of Commerce, National Institute of Standards and Technology (1995)

Set of Enhanced Queries

```
289 Napoleon & (Polish | Poland)
290 genetic & (algorithm | algorithms) &
(history | algorithm | function | (data &
(structure | structures)) | implementation)
291 (Olympian | Olympie) & (god | goddess)
292 Italian & Flemish & painting & Renaissance
293 wifi & security & encryption
294 user & interface & (design | usability)
295 software & ((intellectual & property) |
(patent & license))
296 (Borussia & Dortmund)
297 cool & jazz & West & coast & (musician |
musicians)
298 ((George & Orwell) | (Eric & Arthur &
Blair))
299 software & development & process &
iterative
300 Airbus & A380 & (order | orders | ordered
| (air & (compagny | compagnies)))
301 (Vector & Space & Model) | (Latent &
Semantic & Indexing) | (salton & smart) |
(extended & Boolean & model)
302 (web & services & security)
303 fractal & (applications | application)
304 (allergy | allergies) & (treatment |
treatments)
305 revision & control & (system | systems)
306 (genre & (theory | classification))
| (structuralist & approach) | ((Plato |
Aristotle) & (form | forms))
307 (Islam | Islamic) & faith
308 wedding & (traditions | customs) & (day &
ceremony)
309 Ken & Doherty & (finals | final)
310 Novikov
311 global & warming
312 recessive & (gene | genes) & ((hereditary
& (disease | diseases)) | (genetic &
(disorder | disorders)))
313 kant & philosophy
314 ((food & additive) | (E & number)) &
(toxin | carcinogen)
315 (spider | spiders) & hunting & (insect |
insects)
316 gymnastics & sport & ((discipline |
disciplines) | (movement | movements))
317 (tourism | visit) & paris & ((museum |
museums) | (cathedral | cathedrals))
318 atlantic & ocean & islands & (slave |
slaves)
319 ((northern & lights) | (polar & lights) |
(aurora & borealis))
320 paris & (Gare & de & Lyon) & (Gare & du &
Nord)
321 Antoni & Gaudi & Barcelona
322 (castles | castle | kasteel) & netherlands
323 ikea & founder
324 composition & planet & rings
325 (Cirque & du & Soleil)
326 Scotland & tourism
327 (clone | clones | cloning) & ((United &
States & of & America) | USA)
328 NBA & European & basketball & player
329 (national & (clothing | dress)) &
(Scottish | Scotland)
330 (nobel & prize) & laureate & physics &
(dutch | Netherlands)
331 (tulips | tulip) & (figure | figures)
332 (NCAA & basketball & tournament) | (march
& madness)
333 (steve & wozniak) & (steve & jobs)
334 (Silk & Road) & China
335 acorn & (eat | eating)
336 (species & monotreme)
337 (security & (algorithms | algorithm))
338 high & blood & pressure & (effect |
effects)
339 (Toy & Story)
340 (Reinforcement & Learning) | (Q &
Learning)
341 microkernel & operating & (system |
systems)
342 (birthday & party) & (nick & cave)
343 Goodkind & (novel | novels)
344 XML & database
345 (Sex & Pistols) & Manchester
```

346 unrealscript
347 (state & machine) & (Moore | Mealy)
348 drinking & water & germany
349 protocol & wireless & security
350 animal & flight
351 (Chinese | China) & wedding & (custom | customs | tradition)
352 faster & than & light & travel
353 (in & place) & (sort | sorting) & (algorithm | algorithms)
354 (novel | novels) & (adaptation | adaptations) & (science & fiction & (film | films))
355 (best & actress) & (academy & (awards | award)) & winner
356 (natural & language & processing) & (information & retrieval)
357 (babylonia | babylonian | assyriology) & culture
358 (information & retrieval) & ((semantic & indexing) | (ontologies | ontology))
359 (shortest & path) & (problem | algorithm | algorithms)
360 solar & energy & (electricity | heating)
361 Europe & after & (second & world & war)
362 (effect | effects) & nuclear & power & plant & accident
363 (Bob & Dylan) & (Eric & Clapton)
364 mushroom & (poisonous | poisoning)
365 Peru & international & investment
366 Fourier & transform & (applications | application)
367 (true & story & films) & ((best & director) | (movie & award))
368 Hymenoptera & Apocrita
369 Pillars & Hercules
370 sport & (offside | (off & side)) & (rule | rules)
371 William & Buckley
372 voodoo & (rituals | ritual)
373 Australia & Echelon & spy & network
374 2004 & Tsunami & (aid | aids)
375 (states | countries) & (nuclear & (proliferation | nonproliferation) & treaty) | npt
376 ((diabetes & mellitus) | (type & 2 & diabetes)) & (symptoms | symptom)
377 (malvasia & grape) & (vinification | wine)
378 indoor & (sports | sport) & ball
379 embargo & Cuba
380 headache & fatigue & nausea & symptoms
381 ubiquitous & computing & (application | applications)
382 Aphrodite
383 Lyon
384 (albert & einstein) & (politics | political)
385 arnold & schwarzenegger & ((co & (starring | star)) | cast | casting)
386 fencing & (weapon | weapons)
387 bridge & types
388 rhinoplasty
389 (cryptography | encryption) & key & (algorithm | algorithms)
390 Insomnia & (cause | causes)
391 (rule | rules | play | playing) & cricket
392 Australian & aboriginals & stolen & generation
393 wireless & (devices | device) & (Health & Hazards)
394 global & warming & (effect | effects)
395 September & 11 & conspiracy & (theory | theories)
396 2004 & Tsunami & Indian & Ocean & Earthquake
397 SUSE & Linux
398 ringo & starr & (musicians | musician)
399 mobile & phone & UMTS & (country | countries)
400 (violent) & revolution & (country | countries)
401 (award | awards) & ((eddie & murphy) | (jim & carrey) | (robin & williams))
402 (countries | country) & (europe | european) & (capital | capitals)
403 color & television & analog & (standard | standards)
404 (french | france) & (singer | singers)
405 The & Old & Man & and & the & Sea
406 architecture & (book | books)
407 (Football & World & Cup) & (Miracle & of & Bern)
408 (electroconvulsive & therapy) & depression
409 (Hybrid & Vehicles) & ((fuel & (efficiency | sources)) | types)
410 (Routers | Router | Switches | Switch) & (computer & (network | networks))
411 (GSM & CDMA) & (standards | standard | coverage | roaming | price | prices)
412 (NT | linux | windows) & (stability | price | prices | security)
413 ((capital & cities) | (capitals)) & Europe & (coordinates | population | latitude | longitude)