FuzzyPR

an Effective Passage Retrieval System for QAS

Christensen, Hans Ulrich; Ortiz-Arroyo, Daniel

Published in:
Rough Sets, Fuzzy Sets, Data Mining and Granular Computing

DOI (link to publication from Publisher):
10.1007/978-3-540-72530-5_23

Publication date:
2007

Document Version
Early version, also known as pre-print

*F*uzzyPR: An Effective Passage Retrieval System for QAS

Hans Ulrich Christensen and Daniel Ortiz-Arroyo

Computer Science Department
Aalborg University Esbjerg
Niels Bohrs Vej 8, 6700 Denmark
huc1405@student.aaue.dk, do@cs.aaue.dk

Abstract. In this paper we present *FuzzyPR*, a novel fuzzy logic based passage retrieval system for *Question Answering Systems (QAS)*. *FuzzyPR* employs a fuzzy logic based similarity measure that includes the best performing models to implement the question reformulation intuition. Our experiments show that *FuzzyPR* achieves consistently better performance in terms of coverage than JIRS on the TREC corpora and slightly better on the CLEF corpora.

Keywords: Information Retrieval, Question Answering Systems, Passage Retrieval, Fuzzy Logic.

1 Introduction

A *Question Answering System (QAS)* is one type of information retrieval (IR) system that attempts to find exact answers to user's questions expressed in natural language. In an *Open-Domain Question Answering System (ODQAS)*, questions are not restricted to certain domains and answers have to be found within an unstructured document collection. The *Passage Retrieval (PR) system*, one component of a QAS, extracts text segments from a group of retrieved documents and ranks these passages in decreasing order of computed likelihood for containing the correct answer to a question. Typically, such text segments are referred to as *candidate passages*.

A QAS is bound by the performance of its PR component. A PR system that fails to retrieve any answering passages to a question or returns many, large candidate passages will have a negative impact on the effectiveness of a QAS [1].

Previous research has proposed to use the *question reformulation* intuition: "*frequently, an answer to a (factoid) question can be found as a reformulation of the same question*" to build QAS. An example of the application of the *reformulation intuition* is the question "How much is the international space station expected to cost?" of QA@TREC 11 (QID: 1645)[1]. The answering passage contains the snippet: "(...)United States and Russia, are working together to build

[1] TREC's Question Answering collections are available from:
http://trec.nist.gov/data/qa.html

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 199–207, 2007.

the SPACE STATION, which is EXPECTED TO COST between $40 billion and $60 billion.(...)".

This paper presents *FuzzyPR*, a language-independent PR system for ODQAS. *FuzzyPR* includes a fuzzy logic based implementation of the reformulation intuition. The paper is organized as follows. Section 2 briefly describes related work on passage retrieval systems. Section 3 describes and analyzes the main component mechanisms of a PR system. Section 4 describes *FuzzyPR* and presents its performance results. Finally, Section 5 presents some conclusions and future work.

2 Related Work

JIRS [2] is a PR system that employs a n-gram model. JIRS supports two extensions to the basic n-gram matching mechanism (called *Simple Model*): term weights (called *Term Weight*) and both term weights and a distance measure (called *Distance Model*). JIRS basically ranks higher passages containing larger sequences of the terms contained in the questions. Brill et al.'s Web QAS [3] builds queries constructed as permutations of the terms employed in the question. Kong et al. [4] use fuzzy aggregation operators in a passage-based retrieval system for documents, where the relevance of a document is re-calculated taking into account the retrieved passages. Other research [**?**] [4] has also explored the application of fuzzy logic in a QAS.

Although the application of the *reformulation intuition* has been previously explored to build QAS [2] [3] to our knowledge we are the first to propose a fuzzy logic question-passage similarity measure to model such intuition.

3 Analysis of Main Component Mechanisms in a Passage Retrieval System

The *reformulation intuition* can be modeled using two characteristics of a candidate passage: "*most (important) question terms*" and "*close proximity*". The feature *"most (important) question terms"* is modeled by the fuzzy subset: *The degree to which candidate passages contain all question terms.* The degree of membership varies from 1 when all important question terms occur within a candidate passage to 0 if no question terms occur within the passage. *"Close proximity"* is modeled by the fuzzy subset: *The degree to which the question terms contained in a candidate passage are juxtaposed.* If all question terms of the passage are juxtaposed, then the passage's membership degree in this fuzzy subset is 1. Otherwise, the more distributed the terms are, the lower the degree of proximity approaching 0.

The third vague concept that can be used in the reformulation intuition is *term matching.* In ODQAS, questions and documents commonly suffer from grammatical inflections and typos that have a negative impact on performance. The fuzzy logic interpretation of binary term similarity is the fuzzy subset: *The* *degree to which two terms are identical* yielding 1 if the two terms are identical, a value in $]0,1[$ if they have some letters in common, and 0 if they are very different. In the following subsections we briefly analyze fuzzy models to implement: *proximity of question terms occurring in a passage* and *automatic detection of term variations*. Further details can be found in [5].

3.1 Proximity of Question Terms Occurring in a Passage

Fuzzy proximity measures calculate the degree of proximity within a document of two or more question terms, based on the following two intuitions: 1) if all matching document terms are juxtaposed then the measure yields 1, and 2) the farther away the matching document terms occur, the lower the degree of proximity.

We evaluated three different fuzzy proximity measures as to their ability in finding answering passages for the first 50 questions of TREC11's question set using the AQUAINT corpus. We used the standard QAS evaluation metrics *Mean Reciprocal Rank (MRR)* and *coverage*. *MRR* is defined as the average of the reciprocal rank r_i of the first hit to each question within the top 5 candidate passages:

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} RR_i \,. \tag{1}$$

where $RR_i = \frac{1}{r_i}$ if $r_i \leq 5$ or 0 otherwise and Q is the set of questions. As is done in the JIRS system [2], we measured coverage on the first top 20 passages. *Coverage* is defined as the proportion of questions for which an answer can be found within the n top-ranked passages:

$$cov(Q, D, n) \equiv \frac{|\{q \in Q | R_{D,q,n} \cap A_{D,q} \neq \emptyset\}|}{|Q|}. \tag{2}$$

where Q is the set of questions, D is the passage collection, $A_{D,q}$ the subset of D containing correct answers for $q \in Q$ and $R_{D,q,n}$ the n top ranked passages.

Fig. 1 shows that Mercier and Beigbeder's *Fuzzy Proximity Measure* [6] achieves the same level of coverage at ranks 1-20 as the Extended Distance Factor [5], but performs 7.2% better in terms of MRR.

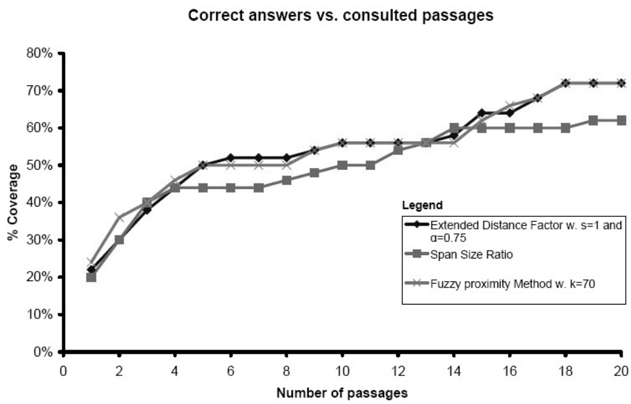
3.2 Automatic Detection of Term Variations

Term variations are lexical differences—in terms of meaning and spelling—between a word of the question typed by a user and an equivalent word contained in a document in the corpus. Reasons for term variations include grammatical inflection and spelling mistakes. Two main features are needed in a mechanism to handle term variations effectively: 1) *language-independence* and 2) *effectiveness*, measured as tolerance toward common misspellings and grammatical inflections, which are interpreted as a type of misspelling.

Fuzzy term similarity algorithms determine the degree of similarity between two strings. Reflexivity and symmetry are desired properties of these algorithms.

Proximity Measure	MRR
Span Size Ratio	0.2933
Fuzzy Proximity Measure	0.3363
Extended Distance Factor	0.3137

(a)



(b)

Fig. 1. The MRRs (a) and coverages (b) of the 3 fuzzy proximity measures

We performed a comparative evaluation on the effectiveness of six different algorithms when set to calculate the similarity between 300 English homophone[2] pairs. The average of the similarity computations yields the score of the fuzzy term matching algorithm.

Table 1. Average similarity scores of 8 Fuzzy similarity algorithms (sorted in decreasing order)

Algorithm	Average similarity score
Normalized longest common subsequence	0.5984
Inverse normalized DD	0.5569
Inverse normalized LD	0.5513
Szczepaniak and Gil	0.4395
Reciprocal DD	0.3751
Reciprocal LD	0.3720
Improved trigram algorithm	0.2477
Trigram algorithm	0.1691

Table 1 shows that the normalized longest common subsequence (nLCS) [5] performed best, giving an average homophone pair similarity rate of 0.5984.

[2] A *homophone pair* is two terms pronounced the same but differing in meaning and spelling, thus reflecting misspellings and typos. Examples include "advice vs. advise" and "cite vs. site".

4 FuzzyPR System and Performance Results

FuzzyPR consists of two components: 1) a question–passage similarity measure module and 2) a passage identification and extraction mechanism adapted to the special needs of QAS. The following subsections describe these components.

4.1 Similarity Measure

The similarity measure we propose is the fuzzy logic-based interpretation of the *reformulation intuition*: "*a passage p is relevant to the user's question q if many question terms or variations of these question terms occur in close proximity*" described by Equation 3.

$$\mu_{rel}(p,q) = wMin\left((v_1, \mu_f(p,q)), (v_2, \mu_p(p,q))\right). \tag{3}$$

This similarity measure combines lexical and statistical data extracted at *term-level* into the two fuzzy measures: $\mu_f(p,q)$ the weighted fraction of question terms q occurring in the passage p and $\mu_p(p,q)$ the proximity of question terms q within the passage. Using the results of the performance analysis described in Section 3, $\mu_f(p,q)$ and $\mu_p(p,q)$ are defined in equations 4 and 5.

$$\mu_f(p,q) = h_{\alpha_f}\left((v_1^f, sat(t_{q_1}, p)) \ldots (v_n^f, sat(t_{q_n}, p))\right). \tag{4}$$

where h is the AIWA importance weighted averaging operator [7] with an ANDness of $\alpha_f = 0.65$, t_{q_i} is a question term, $v_i^f = NIDF(t_{q_i}) = 1 - \frac{log(n_i)}{1+log(N)}$ [3], n=frequency of t_{q_i} in Ω the set of documents, $N = |\Omega|$. $sat(p, t_{q_i})$ measures the degree to which p contains t_{q_i} using the normalized longest common subsequence (nLCS), i.e. $sat(p, t_{q_i}) = \max\limits_{\forall t_p \in p}\left(\mu_{sim}^{nLCS}(t_p, t_{q_i})\right)$, where $\mu_{sim}^{nLCS}(t_p, t_{q_i}) = \frac{|LCS(t_p, t_{q_i})|}{\max(|t_p|, |t_{q_i}|)}$, LCS being the longest common subsequence. Finally,

$$\mu_p(p,q) = \frac{s(p,q)}{\max\limits_{\forall p_i \in \Omega} s(p_i, q)}. \tag{5}$$

where $\mu_p(p,q)$ is a max-normalization of Mercier and Beigbeder's *fuzzy proximity* method [6] described by $s(p,q) = \int_1^n \mu_t^p(x)dx$, $t \in q$ with the term influence function $\mu_t^p(x) = \max\limits_{i \in Occ(t,p)}\left(\max\left(\frac{k - |x-i|}{k}, 0\right)\right)$, where the parameter adjusting the support $k = 70$. The values of v_1, v_2, α_f and k are determined experimentally. Aggregating these two fuzzy measures using the weighted minimum gives the overall relevance score $wMin$, which is defined as:

$$wMin(v_1, v_2, \mu_f, \mu_p) = \min\left(\max(1 - v_1, \mu_f(p,q)), \max(1 - v_2, \mu_p(p,q))\right). \tag{6}$$

with the importance weights $v_1 = 1$, $v_2 = 1$ and both the passage p and the question q represented as sets of terms: $\{t_{p_1}, t_{p_2}, ..., t_{p_n}\}$ and $\{t_{q_1}, t_{q_2}, ..., t_{q_m}\}$,

[3] NIDF is an abbreviation of normalized inverse document frequency.

respectively. $wMin$ aggregates $\mu_f(p,q)$ and $\mu_p(p,q)$ into a single fuzzy value $\mu_{rel}(p,q)$ as described by Equation 3. $\mu_{rel}(p,q)$ is the fuzzy subset of passages providing a correct answer to the question q, where p is a specific passage. $\mu_{rel}(p,q)$ has the advantage of being *language-independent*.

4.2 Mechanism for Passage Identification and Extraction

A fuzzified variation of the concept *arbitrary passages*[4] is employed in *FuzzyPR*. An arbitrary passage is modeled as its membership function in the ideal set of passage sizes as stated in equation 7.

$$\mu_{Ideal\ passage\ size}(x) = \begin{cases} 1 & \text{if } 0 \leq x \leq d \\ \frac{x-b}{d-b} & \text{if } d < x < b \\ 0 & \text{if } x \geq b \end{cases} . \tag{7}$$

x is a term's location in the passages and d and b adjust the crisp support and the fuzzy support respectively. Due to efficiency concerns, the membership function of the ideal passage size set is transformed into an equivalent symmetric membership function, where the center term of a passage is required to have a question term similarity greater than α and a normalized IDF greater than β. This restriction is justified by the intuition that a passage containing none or very few of the question's terms is unlikely to provide an answer to the question.

4.3 Performance Results

We measured the effectiveness of *FuzzyPR* by comparing its ability to find correct answers to questions in a document corpora with both an adapted PR system that we have integrated within Lucene—a popular vector space search engine—and the JIRS PR system [2]. We decided to evaluate the simple model and the distance model of JIRS, because we found that the term weighted model and the simple model perform almost identically.

Both JIRS and the PR system implement an index of 3 sentence passages with 1 sentence overlapping. Llopis et al. in [8] report that this approach achieves good results. The PR system allows Lucene to be used as a PR module in a QAS by employing a simple query expansion method. In this method the question term with the lowest IDF is removed until ≥ 20 passages are retrieved from the index of 3 sentence passages.

Because *FuzzyPR* defines a passage as a number of consecutive terms, we computed and used the arithmetic mean of the average passage sizes of the top 100 passages retrieved by both Lucene, JIRS Distance Model and JIRS Simple Model. In table 2 the numbers in parenthesis are the actual passage sizes used by *FuzzyPR*.

As test data we used TREC12's set of 495 questions and the corpus called AQUAINT consisting of $1,033,461$ documents of English news text and

[4] Arbitrary passages are defined as: *"any sequence of words of any length starting at any word in the document"*.

Table 2. The average passage sizes of the PR systems used for comparison

PR system Test data	TREC12	CLEF04
Lucene	55.91	74.74
JIRS Distance Model	132.23	105.87
JIRS Simple Model	166.96	111.48
Arithmetic mean	118.37 (119)	97.36 (98)

CLEF04's 180 question and the AgenciaEFE corpus of 454,045 Spanish newswire documents. To answer questions automatically for TREC12 we used Ken Litkowsky's regular expression patterns of correct answers[5] and for CLEF4 we used the patterns supplied with JIRS[6]

The TREC12 question set was reduced to 380, since 115 questions do not have a recognizable pattern. As evaluation metrics we used *Mean Reciprocal Rank (MRR)* and *coverage* defined in Section 3. *%impr.* is the improvement (or worsening) *FuzzyPR* achieves compared to a PR system expressed as an percentage.

Table 3. MRRs obtained with TREC12's and CLEF04's QA test data

PR system / QA test data	TREC12	%impr.	CLEF04	%impr.
FuzzyPR	0.3394	-	0.3726	-
JIRS Distance Model	0.3180	6.73%	0.3721	0.13%
JIRS Simple Model	0.2724	24.60%	0.3771	−1.19%
Lucene	0.2910	16.63%	0.3399	9.62%

Tables 3 and 4 show that *FuzzyPR* consistently performs better than Lucene's vector space PR system independently of the number of top-ranked passages consulted tested with both TREC12 and CLEF04 QA test data. MRR is improved at least 9.62% and coverage@20 at least 14.47%.

Comparing the performance of *FuzzyPR* and the two variations of JIRS shows that for TREC12 QA test data in terms of both MRR and coverage *FuzzyPR* performs consistently better. Compared to the second best PR system: JIRS Distance Model, MRR is improved by 6.73% and coverage@20 by 4.15%. As Table 4(b) shows, *FuzzyPR* tested with CLEF04 QA test data in general (18 out of 20 cases) achieves slightly better coverage than JIRS. Table 4 reveals that although *FuzzyPR* fails to boost coverage at the ranks 1 to 3, at ranks 4 to 20 it achieves a 0%-7.87% higher coverage than number two: JIRS Distance Model.

[5] Ken Litkowsky's patterns are available from the TREC website: `http://trec.nist.gov`.

[6] Patterns of correct answers to CLEF QA test data are available from JIRS' web site: `http://jirs.dsic.upv.es/`.

Table 4. The PR systems' coverages tested with (a) TREC12 and (b) CLEF04 data

(a)

Rank	FuzzyPR	Lucene	%impr.	JIRS_SM	%impr.	JIRS_DM	%impr.
1	0.2500	0.2237	11.76%	0.2222	12.51%	0.2434	2.71%
2	0.3579	0.3053	17.23%	0.2698	32.65%	0.3201	11.81%
3	0.4184	0.3500	19.54%	0.2989	39.98%	0.3836	9.07%
4	0.4500	0.3711	21.26%	0.3466	29.83%	0.4206	6.99%
5	0.4868	0.4026	20.91%	0.3704	31.43%	0.4497	8.25%
6	0.5184	0.4237	22.35%	0.4048	28.06%	0.4788	8.27%
7	0.5421	0.4342	24.85%	0.4312	25.72%	0.4921	10.16%
8	0.5684	0.4526	25.59%	0.4471	27.13%	0.5079	11.91%
9	0.5816	0.4789	21.44%	0.4788	21.47%	0.5317	9.38%
10	0.5947	0.4947	20.21%	0.4894	21.52%	0.5476	8.60%
11	0.6105	0.5053	20.82%	0.4947	23.41%	0.5582	9.37%
12	0.6158	0.5237	17.59%	0.5053	21.87%	0.5688	8.26%
13	0.6211	0.5289	17.43%	0.5212	19.17%	0.5794	7.20%
14	0.6237	0.5368	16.19%	0.5265	18.46%	0.5899	5.73%
15	0.6237	0.5474	13.94%	0.5291	17.88%	0.5952	4.79%
16	0.6263	0.5500	13.87%	0.5317	17.79%	0.6032	3.83%
17	0.6316	0.5579	13.21%	0.5476	15.34%	0.6085	3.80%
18	0.6368	0.5605	13.61%	0.5556	14.61%	0.6111	4.21%
19	0.6368	0.5605	13.61%	0.5635	13.01%	0.6164	3.31%
20	0.6447	0.5632	14.47%	0.5714	12.83%	0.6190	4.15%

(b)

Rank	FuzzyPR	Lucene	%impr.	JIRS_SM	%impr.	JIRS_DM	%impr.
1	0.2833	0.2722	4.08%	0.3222	-12.07%	0.3000	-5.57%
2	0.3778	0.3722	1.50%	0.3889	-2.85%	0.3722	1.50%
3	0.4389	0.3944	11.28%	0.4111	6.76%	0.4444	-1.24%
4	0.4944	0.4222	17.10%	0.4500	9.87%	0.4833	2.30%
5	0.5333	0.4389	21.51%	0.4722	12.94%	0.4944	7.87%
6	0.5556	0.4556	21.95%	0.4944	12.38%	0.5278	5.27%
7	0.5611	0.4722	18.83%	0.5222	7.45%	0.5444	3.07%
8	0.5722	0.4722	21.18%	0.5278	8.41%	0.5667	0.97%
9	0.5722	0.4833	18.39%	0.5333	7.29%	0.5722	0.00%
10	0.5944	0.4889	21.58%	0.5611	5.93%	0.5833	1.90%
11	0.6000	0.4889	22.72%	0.5611	6.93%	0.5833	2.86%
12	0.6167	0.4889	26.14%	0.5667	8.82%	0.5944	3.75%
13	0.6222	0.4889	27.27%	0.5667	9.79%	0.6000	3.70%
14	0.6278	0.5000	25.56%	0.5778	8.65%	0.6056	3.67%
15	0.6278	0.5056	24.17%	0.5778	8.65%	0.6167	1.80%
16	0.6389	0.5056	26.36%	0.5778	10.57%	0.6167	3.60%
17	0.6389	0.5056	26.36%	0.5778	10.57%	0.6167	3.60%
18	0.6389	0.5167	23.65%	0.5778	10.57%	0.6222	2.68%
19	0.6444	0.5222	23.40%	0.5833	10.47%	0.6278	2.64%
20	0.6500	0.5333	21.88%	0.5833	11.43%	0.6333	2.64%

However, in terms of MRR, JIRS Simple Model achieves a MRR of 0.3771, which is 1.2% better than *FuzzyPR*. This indicates that sometimes answering passages in this collection do not conform to the *reformulation intuition*. However, this only seems to affect the ability to boost answering passages to higher ranks because JIRS Simple Model falls behind JIRS Distance Model and *FuzzyPR* for coverage@4–20.

FuzzyPR has been optimized using TREC11 QA test data, which might bias the TREC12 results. However, table 4(b) shows that *FuzzyPR* achieves the highest coverage at ranks 4 to 20 for CLEF04 QA test data, too. Because Gómez-Soriano et al. [2] evaluated JIRS with CLEF's Spanish, Italian, and French QA test data it is reasonable to assume that JIRS' system parameters have been optimized for these languages. *FuzzyPR* performs better than JIRS due to the incorporation of two additional fuzzy concept besides those included in the JIRS Distance Model: 1) terms are importance-weighted using inverse document frequencies and 2) instead of n-grams the similarity method uses subsequences of n question terms together with a proximity method yielding the highest similarity when the terms are juxtaposed. Furthermore, compared to JIRS's Distance Model *FuzzyPR* also fuzzifies 3) the definition of passage size and 4) question terms' occurrences in a passage. A last difference is that *FuzzyPR* computes the proximity of the question terms occurring in a passage rather than relying on n-gram or subsequence matching.

5 Conclusions and Future Work

In this paper we presented *FuzzyPR*. *FuzzyPR* implements a fuzzy logic based interpretation of the *reformulation intuition*. *FuzzyPR* has three main advantages: 1) its *passage identification and extraction methods* that enables it to retrieve candidate passages from documents at retrieval time thus avoiding the time-consuming indexing process[7] 2) its *language-independence* property, and 3) its ability to handling spelling errors and grammatical inflections.

Our experiments show that *FuzzyPR* achieves a consistently higher MRR and coverage than Lucene's PR system and JIRS on TREC corpora. Furthermore it performs better in terms of coverage than JIRS on the CLEF corpora at ranks 4 to 20. In future work we plan to evaluate *FuzzyPR* with CLEF's French and Italian corpora to test its performance when compared to JIRS.

References

1. Gaizauskas R., Greenwood M., Hepple M., and Roberts I.: The university of sheffields trec 2003 q&a experiments. *Proceedings of the 12th Text REtrieval Conference*, 2003.
2. Gómez-Soriano J., Montes y Gómez M., Arnal E., and Rosso E.: A passage retrieval system for multilingual question answering. *Proceedings of 8th International Conference of Text, Speech and Dialogue 2005 (TSD'05)*, Lecture Notes in Computer Science, Springer-Verlag, 2005. Karlovy Vary, Czech Republic.
3. Brill E., Lin J., Banko M., Dumais S. , and Ng A.: Data-intensive question answering. *In Proceedings of the Tenth Text REtrieval Conference (TREC 2001), Gaithersburg, Maryland*, pages 443–462, November 2001.
4. Kong K., Luk R., Ho K., and Chung F.: Passage-based retrieval using parameterized fuzzy set operators. *ACM SIGIR Workshop on Mathematical/Formal. Methods for Information Retrieval*, 2004.
5. Christensen H. U.: Exploring the use of fuzzy logic and data fusion techniques in passage retrieval for QA. Master's thesis, Aalborg University Esbjerg, December 2006.
6. Beigbeder M. and Mercier A.: An information retrieval model using the fuzzy proximity degree of term occurrences. *Proceedings of the 2005 ACM symposium on Applied computing*, March 2005.
7. Larsen H. L: Efficient andness-directed importance weighted averaging operators. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, pages 67–82, 2003.
8. Llopis F., Ferrández A., and Luis Vicedo J.: Text segmentation for efficient information retrieval. *In Third International Conference on Intelligent Text Processing and Computational Linguistics (CICLing 2002), Mexico City, Mexico*, Lecture Notes in Computer Science, Springer-Verlag:373–380, 2002.

[7] An unoptimized method in Java for segmenting and indexing the AQUAINT corpus took 4 hours on an AMD64 3400+ w. 2 GB RAM and RAID 0.