Improving the Automatic Retrieval of Text Documents

Maristella Agosti, Michela Bacchin, Nicola Ferro, and Massimo Melucci

Department of Information Engineering
University of Padua, Via Gradenigo, 6/a – 35031 Padova, Italy
{maristella.agosti,michela.bacchin,nicola.ferro,massimo.melucci}@unipd.it

Abstract. This paper reports on a statistical stemming algorithm based on link analysis. Considering that a word is formed by a prefix (stem) and a suffix, the key idea is that the interlinked prefixes and suffixes form a community of sub-strings. Thus, discovering these communities means searching for the best word splits that give the best word stems. The algorithm has been used in our participation in the CLEF 2002 Italian monolingual task. The experimental results show that stemming improves text retrieval effectiveness. They also show that the effectiveness level of our algorithm is comparable to that of an algorithm based on a-priori linguistic knowledge.

Keywords: Italian Text Retrieval; Information Retrieval; Web Information Gathering; Stemming; Link-based Analysis

1 Introduction

The main objective of the research reported in this paper is to design, implement and evaluate a language-independent stemming algorithm based on statistical and link-analysis methods. To accomplish this objective, we designed our experiments to investigate whether stemming does not deteriorate or even enhances the effectiveness of retrieval of documents written in Italian, using both a linguistic and a statistical stemmer. We then investigated whether a statistical and language-independent stemming algorithm can perform as effectively as an algorithm developed on the basis of a-priori linguistic knowledge.

The paper is organized as follows: Section 2 illustrates our previous work in multilingual information retrieval. We report on a background activity for the building of test collections of documents written in Italian – the seed work that has led us to investigate and concentrate on stemming as a fundamental building block in the retrieval of textual documents written in Italian. Using our experience in the construction of a document collection for the retrieval of Italian documents as a starting point, we realized that a real weakness in text retrieval is dependence on the specific languages used in the documents of interest. Thus we concentrate on efforts for developing stemming methods and algorithms that can be independent of the specific languages of interest. The second part of the paper reports on results obtained in this area. In particular, Section 3 introduces

C.A. Peters (Ed.): CLEF 2002, LNCS 2785, pp. 279–290, 2003.

the stemming process, Section 4 reports on the methodological approach we followed to build a new statistical and language-independent stemming algorithm, Section 5 describes our runs and the results obtained, and Section 6 reports some conclusions and future work.

2 Background

One of the essential tools needed to conduct research in information retrieval for a language other than English is a test collection of documents written in the language of interest. The purpose of such a collection is to make research results repeatable and available for evaluation by different research groups, with different systems, and in different contexts. At the Department of Information Engineering of the University of Padua, research work on multilingual information retrieval dates back to 1998, when we started to design and implement a test collection for the retrieval of documents written in Italian in order to perform experiments using stemming algorithms [1]. Unfortunately, the test collection is not yet publicly available due to copyright constraints.

When we started the investigation, we decided to select a large set of full-text documents, representing one year's publication of an Italian newspaper. We chose a newspaper that publishes the complete collection of articles for each year on CD-ROM together with an information retrieval system that can be used to search it. The collection we chose has almost seventy thousand documents, so its size is comparable to the size of one of the TREC sub-collections, such as WSJ, distributed on TREC disk 2, which contains 74,520 newspaper documents taken from the "Wall Street Journal, 1990-1992".

Examining the content of the source collection of documents, some information needs were identified and corresponding topics were written in natural language in order to create a query set. Our objectives were twofold:

- to build up a set of "real" queries, i.e. reflecting queries a hypothetical "real" final user would submit to a system providing access to the collection, and
- to ensure that some of the queries constructed refer to real events or facts being reported in newspapers, other than general subjects.

Therefore, the test queries refer to specific facts, and include specific words, such as proper names or dates. The final requirement concerned the query size: queries were similar in size to real ones as formulated by real final users of the source collection. In order to be able to create sufficiently specific queries, referring to real facts or events, it was decided to use the classification scheme provided on the CD-ROM, as a way to "suggest" potential reasonable test queries. The class names were used as a starting point for the definition of the queries, and each query was compiled using one of a set of selected categories from the classification scheme.

The task of assessing the relevance of test documents with respect to test queries should in principle be exhaustive, i.e. should be made for every document-query pair in order to have the total number of relevant documents of the entire

collection for each query. However, such a exhaustive task is clearly impossible, and this is why sampling methods have been proposed [21].

We developed a new method that combines the experience of TREC with the exploitation of the classification system which was used to classify all the documents of the set. The main contributions were provided by (1) the different tools and evidence used to compile the relevance assessments, and (2) the assignment of relevance assessments to different document parts to enable the evaluation of tasks that are different from classical document retrieval, such as passage retrieval.

The methodology we set up to build document set samples was based on the combined use of a classification system and of a query tool. In fact, each year during the preparation of the CD-ROM of the newspaper articles, a group of experts manually classifies all the documents using a classification system that has the following characteristics: it has been built by experts in the domain of the source database; it is a specialized classification; it allows overlapping, so a document can be classified in two or more classes. The query tool available on the CD-ROM provides quite sophisticated Boolean searching capabilities, since the tool has been tuned to the available set of documents in order to provide effective retrieval.

At relevance assessment time, our human assessors were asked first to find relevant documents in some predefined categories which were likely to include relevant material. Then, they had to find additional relevant documents from the whole document collection using the query tool. While the classification system helped to increase retrieval precision, the query tool made it possible to increase recall by retrieving many other relevant documents outside the categories in which it was thought that the newly retrieved relevant documents should have been classified.

Moreover, as newspaper documents have different lengths and structures, it was quite common to have long documents in the retrieved set. Therefore, it was necessary to give relevance judgments on different document parts in order to capture different relevant portions of the documents. Specifically, assessors were asked to assess distinctly the relevance of the title, first paragraph, and whole text. In this way, there are three levels of assessment for each document. The resulting test collection is endowed with relevance judgments at a level of granularity which allows us to design document structuring-based applications. One of the problems that remained open after the design of this Italian test collection was the availability of an effective stemming algorithm, preferably an algorithm that adopted a methodology that could be reusable for languages other than Italian. Thus we decided to focus our attention on stemming, and we decided to participate in the CLEF 2002 campaign with the aim of evaluating a language-independent algorithm.

3 Stemming

Stemming is used to reduce variant word forms to a common root. The assumption is that if two words have the same root, then they represent the same concept. Hence stemming permits an IR system to match query and document terms which are related to the same meaning but which can appear in different morphological variants.

The effectiveness of stemming is a debated issue, and there are different results and conclusions. If effectiveness is measured by the traditional precision and recall measures, it seems that for a language with a relatively simple morphology, like English, stemming influences the overall performance little [7]. In contrast, stemming can significantly increase retrieval effectiveness [14] and can also increase precision for short queries [9], for languages with a more complex morphology, like the Romance languages. Finally, as system performance must reflect user's expectations it must be remembered that the use of a stemmer is apparently assumed by many users [7], who express a query to a system using a specific word without taking into account that only a variant of this word may appear in a relevant document. Hence, stemming can also be viewed as a feature that is related to the user-interaction interface of an IR service.

When designing a stemming algorithm, it is possible to follow a linguistic approach, using prior knowledge of the morphology of the specific language, or a statistical approach using methods based on statistical principles to infer the word formation rules in the language studied from a corpus of documents in that language. The former implies manual labor which has to be done by linguistic experts – in fact, it is necessary to formalize the word formation rules and this is hard work, especially for those languages with a complex morphology. Stemming algorithms based on statistical methods imply low costs to insert new languages in the system, and this is an advantage that can become crucial, especially for multilingual IR systems.

4 Methodology

We will consider a special case of stemming, which belongs to the category known as *affix removal stemming* [4]. In particular our approach follows a suffix stripping paradigm which is adopted by most stemmers currently in use by IR, such as those reported in [10, 13, 16]. This stemming process splits each word into two parts, prefix and suffix, and considers the stem as the sub-string corresponding to the prefix obtained.

Let us consider a finite collection of unique words $W = \{w_1, ..., w_N\}$ and a word $w \in W$ of length $|w|$, then w can be written as $w = xy$ where x is a prefix and y is a suffix. If we split each word w into all the $|w| - 1$ possible pairs of sub-strings, we build a collection of sub-strings, and each sub-string may be either a prefix, a suffix, or both, of at least an element $w \in W$. Let X be the set of the prefixes of the collection and $S \subseteq X$ be the set of the stems. We are interested in detecting the prefix x that is the most probable stem for the

observed word w. Hence, we have to determine the prefix x^* such that:

$$x^* = \arg\max_x Pr(x \in S \mid w \in W) \tag{1}$$

$$= \arg\max_x \frac{Pr(w \in W \mid x \in S) Pr(x \in S)}{Pr(w \in W)} \tag{2}$$

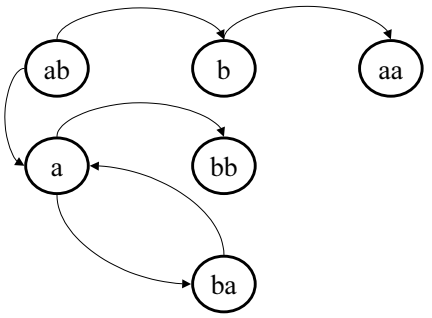
where (2) is obtained applying Bayes' theorem which lets us swap the order of dependence between events. We can ignore the denominator, which is the same for all splits of w. $Pr(w \in W \mid x \in S)$ is the probability of observing w given that the stem x has been observed. A reasonable estimation of that probability would be the reciprocal of the number of words beginning with that stem if the stems were known. However note that the stems are unknown – indeed stem detection is the target of this method – and the number of words beginning with a stem cannot be computed. Therefore we estimated that probability by the reciprocal of the number of words beginning by that prefix. With regard to $Pr(x \in S)$ we estimated this probability using an algorithm that discloses the mutual relationship between stems and derivations in forming the words of the collection.

The rationale of using mutual reinforcement is based on the idea that stems extracted from W are those sub-strings that:

- are very frequent, and
- form words together with very frequent suffixes.

This means that very frequent prefixes are candidate stems, but they are discarded if they are not followed by very frequent suffixes; for example, all initials are very frequent prefixes but they are unlikely stems because the corresponding suffixes are rather rare, if not unique – the same holds for suffixes corresponding to ending vowels or consonants. Thus, there are prefixes that are less frequent than initials, but followed by suffixes that are frequent but less frequent than ending characters: these suffixes and prefixes correspond to candidate correct word splits and we label them as "good". The key idea is that interlinked good prefixes and suffixes form a community of sub-strings whose links correspond to words, i.e. to splits. Discovering these communities is like searching for the best splits.

To compute the best split, we used the quite well-known algorithm called HITS (Hyperlink Induced Topic Search) reported in [8] and often discussed in research papers as a paradigmatic algorithm for Web page retrieval. It considers a mutually reinforcing relationship among good authorities and good hubs, where an authority is a web page pointed to by many hubs and a hub is a web page which points to many authorities. The parallel with our context will be clear when we associate the concept of a hub with a prefix and that of an authority with a suffix. The method belongs to the larger class of approaches based on frequencies of sub-strings to decide the goodness of prefixes and suffixes, often used in statistical morphological analysis [12, 5], and in pioneer work [6]. The contribution of this paper is the use of the mutual reinforcement notion applied



(a)

substring	prefix score	suffix score
a	0.250	0.333
aa	0.000	0.167
ab	0.375	0.000
b	0.125	0.167
ba	0.250	0.167
bb	0.000	0.167

(b)

Fig. 1. (a) The graph obtained from W. (b) The prefix and suffix scores from W

to prefix frequencies and suffix frequencies, to compute the best word splits which give the best word stems as explained in the following.

Using a graphical notation, the set of prefixes and suffixes can be written as a graph $g = (V, E)$ such that V is the set of sub-strings and $w = (x, y) \in E$ is an edge w that occurs between nodes x, y if $w = xy$ is a word in W. By definition of g, no vertex is isolated. As an example, let us consider the following toy set of words: W={`aba`, `abb`, `baa`}; splitting these into all the possible prefixes and suffixes produces the graph reported in Figure 1a.

If a directed edge exists between x and y, the mutual reinforcement notion can be stated as follows:

> good prefixes point to good suffixes, and good suffixes are pointed to by good prefixes.

Let us define $P(y) = \{x \in V : \exists w, w = xy\}$ and $S(x) = \{y \in V : \exists w, w = xy\}$ as, respectively, the set of all prefixes of a given suffix y and the set of all suffixes of a given prefix x. If p_x and s_x indicate, respectively, the prefix score and the suffix score, the criteria can be expressed as:

$$p_x = \sum_{y \in S(x)} s_y \qquad s_y = \sum_{x \in P(y)} p_x \tag{3}$$

under the assumption that scores are expressed as sums of scores and splits are equally weighed.

The mutual reinforcement method has been formalized through the HITS iterative algorithm. Here we map HITS in our study context, as follows:

Compute suffix scores and prefix scores from W
V: the set of substrings extracted from all the words in W
$P(y)$: the set of all prefixes of a given suffix y
$S(x)$: the set of all suffixes of a given prefix x
N: the number of all substrings in V
n: the number of iterations
$\mathbf{1}$: the vector $(1, ..., 1) \in \mathcal{R}^{|V|}$
$\mathbf{0}$: the vector $(0, ..., 0) \in \mathcal{R}^{|V|}$
$\mathbf{s}^{(k)}$: suffix score vector at step k
$\mathbf{p}^{(k)}$: prefix score vector at step k
$\mathbf{s}^{(0)} = \mathbf{1}$
$\mathbf{p}^{(0)} = \mathbf{1}$
for each iteration $k = 1, ..., n$
$\mathbf{s}^{(k)} = \mathbf{0}$
$\mathbf{p}^{(k)} = \mathbf{0}$
for each $y \in V$
$s_y^{(k)} = \sum_{x \in P(y)} p_x^{(k-1)}$;
for each $x \in V$
$p_x^{(k)} = \sum_{y \in S(x)} s_y^{(k)}$;
normalize $\mathbf{p}^{(k)}$ and $\mathbf{s}^{(k)}$ so that $1 = \sum_x p_x^{(k)} = \sum_y s_y^{(k)}$
end.

Using the matrix notation, graph g can be described with a $|V| \times |V|$ matrix $\mathbf{M}$ such that

$$m_{ij} = \begin{cases} 1 & \text{if prefix } i \text{ and suffix } j \text{ form a word} \\ 0 & \text{otherwise} \end{cases}$$

As explained in [8], the algorithm computes two matrices: $\mathbf{A} = \mathbf{M}^T\mathbf{M}$ and $\mathbf{B} = \mathbf{M}\mathbf{M}^T$, where the generic element a_{ij} of $\mathbf{A}$ is the number of vertices that are pointed by both i and j, whereas the generic element b_{ij} of $\mathbf{B}$ is the number of vertices that point to both i and j. The n-step iteration of the algorithm corresponds to computing $\mathbf{A}^n$ and $\mathbf{B}^n$. In the same paper, it was argued that $\mathbf{s} = [s_y]$ and $\mathbf{p} = [p_x]$ converge to the eigenvectors of $\mathbf{A}$ and $\mathbf{B}$, respectively. The scores computed for the toy set of words are reported in Table 1.

As explained previously, we argue that the probability that x is a stem can be estimated with the prefix score p_x just calculated. The underlying assumption is that the scores can be seen as probabilities, and, in effect, it has been shown in a recent paper that HITS scores can be considered as a stationary distribution of a random walk [2]. In particular, the authors proved the existence of a Markov chain, which has the stationary distribution equal to the hub vector after the n^{th} iteration of Kleinberg's algorithm, which is, in our context, the prefix score vector $\mathbf{p}^{(n)}$. The generic element $q_{ij}^{(n)}$ of the transition matrix referred to the chain is the probability that, starting from i, one reaches j after n "bouncing" to one

Table 1. The candidate splits from W={aba, baa, abb}

word	prefix	suffix	words beginning by prefix	words ending by suffix	probability	choice
baa	b	aa	1	1	0.1250	
baa	ba	a	1	2	0.2500	*
aba	a	ba	2	1	0.1250	
aba	ab	a	2	2	0.1875	*
abb	a	bb	2	1	0.1250	
abb	ab	b	2	1	0.1875	*

of the suffixes which is associated with both i and j. To interpret the result in a linguistic framework, p_i can be seen as the probability that i is judged as a stem by the same community of substrings (suffixes) resulting from the process of splitting the words of a language. In Table 1, all the possible splits for all the words are reported and measured using the estimated probability.

5 Experiments

The aim of our CLEF 2002 experiments has been to compare the retrieval effectiveness of the link analysis-based algorithm illustrated in the previous section with that of an algorithm based on a-priori linguistic knowledge; the hypothesis is that a language-independent algorithm, such as the one we propose, might effectively replace one developed on the basis of manually coded derivational rules. Before comparing the algorithms, we assessed the impact of both stemming algorithms by comparing their effectiveness with that reached without any stemmer. In fact, we wanted to test whether system performance is not significantly hurt by the application of stemming, as hypothesized in [7]. If, on the contrary, stemming improved effectiveness, and the effectiveness of the tested algorithms were comparable, the link-based algorithm could be extended to other languages inexpensively, which is of crucial importance in multilingual settings. To evaluate stemming, we decided to compare the performance of an IR system when different stemming algorithms were used for different runs, all other things being equal.

5.1 Experimental Prototype System

For indexing and retrieval, we used an experimental IR system, called IRON, which has been developed by our research group in order to have a robust tool for carrying out IR experiments. IRON is built on top of the Lucene 1.2 RC4 library, which is an open-source library for IR written in Java and publicly available at [11]. The system implements the vector space model [17], and a $(\text{tf} \cdot \text{idf})$–based weighting scheme [18]. The stop-list which was used consists of 409 Italian frequent words and is publicly available at [19].

In order to develop the statistical stemming algorithm, we built a suite of tools, called Stemming Program for Language Independent Tasks (SPLIT), which implements the link-based algorithm and chooses the best stem, according to the probabilistic criterion described in Section 4. From the vocabulary of the Italian CLEF sub-collection, SPLIT spawns a 2,277,297-node and 1,215,326-edge graph, which is processed to compute prefix and suffix scores – SPLIT took 2.5 hours for 100 iterations on a personal computer equipped with Linux, 800 MHz CPU and 256MB RAM.

5.2 Runs

We tested four different stemming algorithms:

1. `NoStem`: No stemming algorithm was applied.
2. `Porter-like`: We used the stemming algorithm for Italian, which is freely available on the Snowball Web Site [15] edited by M. Porter. Besides being publicly available for research purposes, we chose this algorithm because it uses a kind of a-priori knowledge of the Italian language. Thus, comparing our SPLIT algorithm with this particular "linguistic" algorithm could provide some information with respect to the feasibility of estimating linguistic knowledge from statistically inferred knowledge.
3. `SPLIT`: We implemented our first version of the stemming algorithm based on a link-analysis with 100 iterations.
4. `SPLIT-L3`: We included in our stemming algorithm a minimum of linguistic knowledge, inserting a heuristic rule which forces the length of the stem to be at least 3.

5.3 A Global Evaluation

We carried out a macro evaluation by averaging the results over all the queries of the test collection. Table 2 shows a summary of the figures related to the macro analysis of the stemming algorithm for 2001 topics, while Table 3 reports data on 2002 topics, which are our official runs submitted at CLEF.

Table 2. Macro comparison of runs for 2001 topics

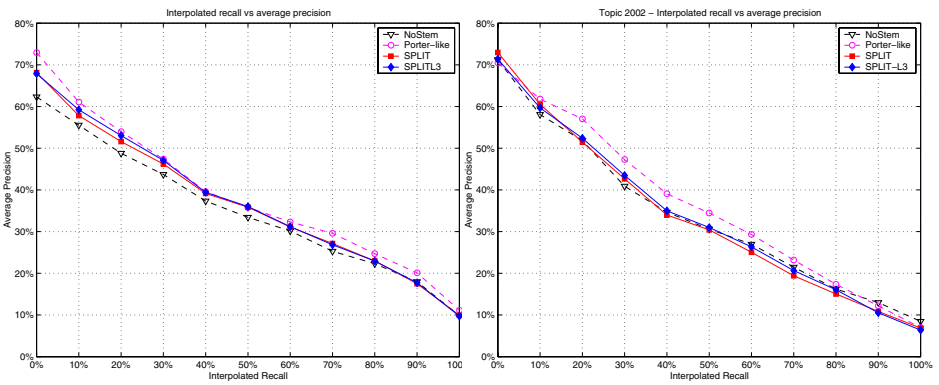
Algorithm	N. Relevant Retrieved	Av. Precision	R-Precision
`NoStem`	1093	0.3387	0.3437
`Porter-like`	1169	0.3753	0.3619
`SPLIT`	1143	0.3519	0.3594
`SPLIT-L3`	1149	0.3589	0.3668

Table 3. Macro comparison among runs for 2002 topics

Run ID	Algorithm	N. Relevant Retrieved	Av. Precision	R-Precision
PDDN	NoStem	887	0.3193	0.3367
PDDP	Porter-like	914	0.3419	0.3579
PDDS2PL	SPLIT	913	0.3173	0.3310
PDDS2PL3	SPLIT-L3	911	0.3200	0.3254

Note that both for 2001 and 2002 topics, all the stemming algorithms considered improve recall, since the number of retrieved relevant documents is larger than the number of retrieved relevant documents observed in the case of retrieval without any stemmer; this increase has been observed for all the stemming algorithms. With regard to precision, while for 2002 topics stemming does not hurt the overall performances of the system, for 2001 data stemming actually increases precision, and overall performance is higher thanks to the application of stemming.

Figure 2 shows the Averaged Recall-Precision curve at different levels of recall for the 2001 and 2002 topic sets. With respect to the use of link-based stemming algorithms, it is worth noting that SPLIT can attain levels of effectiveness that are comparable to algorithms based on linguistic knowledge. This is surprising if you know that SPLIT was built without any sophisticated extension to HITS and that neither heuristics nor linguistic knowledge was used to improve effectiveness, except for the slight constraint of SPLIT-L3. The result should also be considered good bearing in mind that it has been obtained for Italian, which is morphologically more complex than English.



CLEF 2001 Topic Set CLEF 2002 Topic Set

Fig. 2. Average Precision curves for four stemming algorithms

6 Conclusions and Future Work

The objective of this research was to investigate a stemming algorithm based on link analysis procedures. The idea is that prefixes and suffixes, which are stems and derivations, form communities once extracted from words. We tested this hypothesis by comparing the retrieval effectiveness of SPLIT, a link analysis based algorithm derived from HITS, with a linguistic knowledge based algorithm, on a relatively morphologically complex language such as Italian.

The results are encouraging because the effectiveness of SPLIT is comparable to the algorithm developed by Porter. The results should be considered even better since SPLIT does not incorporate any heuristics nor linguistic knowledge. Moreover, both stemming and then SPLIT have been shown to improve effectiveness with respect to not using any stemmer.

We are carrying out further analysis at a micro level to understand the conditions under which SPLIT performs better or worse than other algorithms. Further work is in progress to improve the probabilistic decision criterion and to insert linguistic knowledge directly in the link-based model in order to weight links among prefixes and suffixes with a probabilistic function that could capture available information on the language, such as, for example, the minimum length of a stem. Finally, further experimental work is in progress with other languages.

Acknowledgements

The authors wish to express their thanks to Carol Peters for the fruitful discussions on multilingual retrieval and for her support during their participation in the CLEF 2002 campaign. The authors have been partially supported by a grant of the Department of Information Engineering of the University of Padua. Massimo Melucci has been partially supported by the Young Researchers programme of the University of Padua.

References

[1] M. Agosti, M. Bacchin, and M. Melucci. Report on the Construction of an Italian Test Collection. Position paper at the *Workshop on Multi-lingual Information Retrieval* at the *ACM International Conference on Research and Development in Information Retrieval (SIGIR)*, Berkeley, CA, USA, 1999. 280

[2] A. Borodin, G. O. Roberts, J. S. Rosenthal, and P. Tsaparas. Finding authorities and hubs from link structures on the World Wide Web. In *Proceedings of the World Wide Web Conference*, pages 415–429, Hong Kong, 2001. ACM Press. 285

[3] C. Cleverdon. The Cranfield Tests on Index Language Devices. In K. Sparck Jones and P. Willett (Eds.). *Readings in Information Retrieval*, pages 47-59, Morgan Kaufmann, 1997.

[4] W. B. Frakes and R. Baeza-Yates. *Information Retrieval: data structures and algorithms*. Prentice Hall, 1992. 282

[5] J. Goldsmith. Unsupervised learning of the morphology of a natural language. *Computational Linguistics*, 27(2):154–198, 2001. 283
[6] M. Hafer and S. Weiss. Word segmentation by letter successor varieties. *Information Storage and Retrieval*, 10:371–385, 1994. 283
[7] D. Harman. How effective is suffixing? *Journal of the American Society for Information Science*, 42(1):7–15, 1991. 282, 286
[8] J. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, September 1999. 283, 285
[9] R. Krovetz. Viewing Morphology as an Inference Process,. In *Proceedings of the ACM International Conference on Research and Development in Information Retrieval (SIGIR)*, 1993. 282
[10] J. Lovins. Development of a stemming algorithm. *Mechanical Translation and Computational Linguistics*, 11:22–31, 1968. 282
[11] The Jakarta Project. Lucene. `http://jakarta.apache.org/lucene/docs/index.html`, 2002. 286
[12] C. D. Manning and H. Schütze. *Foundations of statistical natural language processing*. The MIT Press, 1999. 283
[13] C. D. Paice. Another Stemmer. In *ACM SIGIR Forum*, 24, 56–61, 1990. 282
[14] M. Popovic and P. Willett. The effectiveness of stemming for natural-language access to Slovene textual data. *Journal of the American Society for Information Science*, 43(5):383–390, 1992. 282
[15] M. Porter. Snowball: A language for stemming algorithms. `http://snowball.sourceforge.net`, 2001. 287
[16] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980. 282
[17] G. Salton and M. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York, NY, 1983. 286
[18] G. Salton and C. Buckley. Term weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523, 1988. 286
[19] Institut interfacultaire d'informatique. CLEF and Multilingual information retrieval. University of Neuchatel. `http://www.unine.ch/info/clef/`, 2002. 286
[20] C. Buckley. Trec_eval. `ftp://ftp.cs.cornell.edu/pub/smart/`, 2002.
[21] E. M. Voorhees. Special Issue on the Sixth Text Retrieval Conference (TREC-6). *Information Processing and Management*. Volume 36, Number 1, 2000. 281