Context Resolution Strategies for Automatic Wikipedia Linking

Michael Granitzer[1], Christin Seifert[2], and Mario Zechner[2]

[1] Knowledge Management Institute
Graz University of Technology
Inffeldgasse 21a, 8010 Graz
mgranitzer@tugraz.at,
http://kmi.tugraz.at/
[2] Know-Center Graz
Inffeldgasse 21a, 8010 Graz
{mzechner,cseifert}@know-center.at,
http://www.know-center.at/

Abstract. Automatically linking Wikipedia pages is done mostly by two strategies: (i) a content based strategy based on word similarities or (ii) a structural similarity exploiting link characteristics. In our approach we focus on a content based strategy by finding anchors using the title of candidate Wikipedia pages and resolving matching links by taking the context of the link anchor, i.e. its surrounding text, into account. Best-entry-points are estimated on a combination of title and content based similarity. Our goal was to evaluate syntactic title matching properties and the influence of the context around anchors for disambiguation and best-entry-point detection. Results show, that the whole Wikipedia page provides the best context for resolving links and that simple inverse document frequency based scoring of anchor texts is also capable of achieving high accuracy.

Key words: INEX, Link-the-Wiki, content based approach, similarity analysis

1 Introduction

This paper outlines the approach taken by the Know-Center Graz in the Link-the-Wiki Track of INEX 2008. While the task itself is outlined in detail in a previous chapter, we restrict ourselves to illustrate our approach and to discuss properties found to be relevant. Our approach has been evaluated on the 659,413 Wikipedia pages, wherefrom every link from or to a page in the test set as well as all pages of the test set itself have been excluded. In the following we refer to the corpus without the test set as the *Wikipedia corpus*. The two runs are distinguished as *file-to-file run*, having 6.600 test documents and *anchor-to-bep run* having 50 topics. A wiki page having all links removed from, is called an *orphan page*.

Basically we restricted ourselves to use standard retrieval techniques combined with efficient string matching technique as used in information extraction frameworks like GATE [2]. As shown in last years runs [6] and in related work from Wu and Weld [8] using such content based approaches yields reliable results. This fact and the applicability of standard IR and IE tools served as motivation for the presented work.

In our approach we identified the following steps for identifying links between Wikipedia pages:

1. *Source Document Identification*: In a first step the source of a link has to be determined. While the orphan wiki page determines the link source for out-links, in-link detection requires efficient search strategies for fast detection of source document candidates. Based on results from INEX 07, our approach utilizes searching for Wikipedia titles in order to determine source document candidates for in-links.
2. *Anchor Identification and Ranking*: The second step includes detection of the anchor position in the source document. We identify anchors by annotating source documents on the word level using a gazetteer list. The list is created from titles and/or anchor texts of links in the Wikipedia corpus. Each link is assigned a score based on the similarity of the surrounding text, called *anchor context* in the following, with the target page in the Wikipedia corpus. We use different document decomposition strategies for determining the *anchor context* based on the assumption that they influence the quality of the score.
3. *Target Document Identification*: The third step includes detection of the link target. Gazetteer matching already provides a list of target documents for each anchor. However, anchors may overlap. For example “United States of America” may have two overlapping links, one for “United States” and one for “United States of America”. Those overlapping links have to be resolved here. Furthermore, in the file-to-file linking runs links have to be merged since they have to be provided on the document instead of the anchor level.
4. *Best-Entry-Point Identification*: As a last step the best-entry-points in the target document have to be identified. Again we focused on using the anchor context and calculate the similarity between the anchor context and document parts in the target document. The target document parts have been again identified using different decomposition strategies. In addition, if a target document part contains the title or parts of the title of the source page, we increased the similarity significantly. Thus, document parts containing the title are preferred as best-entry-point.

Our work aims to provide answers on issues like syntactic matching (i.e. on the word level like case sensitivity, part-of-speech tags, title matching vs. title and anchors etc.), influence of document decomposition strategies (i.e. sentences, automatically generated topics and document) on determining the anchor context as well as scoring strategies for removing high frequent, noisy links like for example “The” or “Are”. Since we use standard information retrieval and information extraction technology our approach may be easily put into practice; a valuable aspect for industries. We hope that our approach provides contrast to other approaches and shed light into automatic linking strategies for Wikipedia as well as its use in practical settings.

2 Document Indexing

The Wikipedia corpus itself is indexed using the open source search engine Lucene [5], with standard stop word removal. For each Wikipedia page the title and all anchors of links pointing to this page are extracted and stored for usage as gazetteer list. A finite state machine (FSM) is filled from this gazetteer. Transitions between states of the FSM are words occurring in a gazetteer entry, while states are distinguished into final states or intermediate states. Final states contain the URL of the Wikipedia page and if upon matching such a final state is reached, an annotation pointing to the particular Wikipedia page is added. In this way gazetteer matching allows us to annotate word sequences with hyperlinks for a large number of possible link targets at reasonable speed.

Orphan pages are preprocessed using the OpenNLP toolkit [1], whereby preprocessing includes tokenization, sentence detection and part-of-speech tagging. In a last step, the FSM is applied to annotate possible links. For resolving the context of an anchor we are relying on either the complete document, the sentence an anchor occurs in as well as a automatically detected, topical coherent block of sentences around the anchor. This automatic topic segmentation of a page is done using the well known C99 segmentation algorithm [3] and sentences are obtained by the sentence detection algorithm.

3 Linking Strategies

For a given orphan page d_o, our system determines a set of n possible in-links $I = \{< l_1, s_1 > \ldots < l_n, s_n >\}$ and a set of m possible out-links $O = \{< l_1, s_1 > \ldots < l_m, s_m >\}$. Each out-link/in-link is assigned a score s_i determining the confidence of the system in generating such a link. One link is - as defined in the LTW result set specification - a quadruple $l_h =< s_h, t_h, sp_h, b_h >$ where for link l_h s_h denotes the source page, t_h the target page, sp_h the span determining start and end of the link in the source document and b_h the best-entry-point in the target document.

In the following we present how the different properties have been determined and thereby differentiate between out-link and in-link generation. While both follow the same principle approach there are slight implementation differences for keeping link generation computational feasible.

3.1 Out-link Generation

Out-link generation starts with preprocessing the orphan document d_o as outlined in section 2. Gazetteer matching returns the set of possible out-links O, whereby for each link l_i we know its source s_i, its target t_i and its span sp_i. For each link we determine the anchor context, defined as the text surrounding the link source. In our experiments we distinguish between sentence, automatically detected topics and the complete document as anchor context. All nouns of the anchor context are extracted and fed into the retrieval backend as Boolean OR query. To speed up this potentially large OR query we restrict the result set to pages pointed to by all links in the anchor context simply by adding all link target identifiers (i.e. the filename of the page) as AND query part. Thus, for all links having a span sp in the current anchor context we are receiving a score s. In particular the query is formulated as

$$(ID = t_1 \;\; OR \ldots OR \;\; ID = t_n) \; AND \; (w_1 \;\; OR \;\; w_2 \; \ldots \; OR \;\; w_k)$$

with $\{w_1 \ldots w_k\}$ as the nouns of the anchor context and t_k as unique identifier for the k^{th} link target and $''ID = ``$ specifying the search on the metadata field containing the unique identifiers of a Wikipedia page. Formally, the score (named *anchor context score* in the following) returned is obtained from standard Lucene ranking as

$$s_i = coord_{w,i} * norm(w) * \sum_{t \in w} \frac{\sqrt{tf_{t,i}} * idf_t^2}{norm(i)} \tag{1}$$

where

- $tf_{t,i}$ is the frequency of term t in document i
- $idf_t = 1 + \log \frac{\#D}{\#D_t + 1}$ is the inverse document frequency with $\#D$ as the number of documents in the corpus and $\#D_t$ the number of documents containing term t
- $norm(w)$ is the norm of the query calculated as $\sqrt{\sum_k idf_k^2}$
- $norm(i)$ is the length norm of document i, namely the number of terms contained in document i
- and $coord_{w,i}$ is a overlapping factor increasing the score the higher the number of overlapping terms between query and documents are.

The Lucene scoring equation has been proven as reliable heuristic for full text searching. It can be seen as an heuristic version of a cosine similarity between anchor context and target document with emphasize towards the number of overlapping words. This assumption is quite naturally for resolving the context of a key. For example “tree” in computer science will occur more frequently with terms describing data structures than the “tree” in nature. Thus, depending on the position of a link in the document we receive different scores. For linking on the word-level resp. for the best-entry-point task, links are ranked according to their score and the best $n = 50$ links are taken as candidates.

An alternative determination for the score is based on the observation that some pages have very common titles. Such pages are for example “The”, “Are”. For removing such high frequent anchors we simply took the inverse document frequency of the anchor text as scoring scheme. The rational is that noisy, high frequent links occur in nearly every document and therefore provide no additional information independent whether they are a true links or not. In particular the score, named *anchor IDF* in the following, is calculated as

$$s_i = \log \frac{\#D}{\#D_a} + 1$$

where $\#D$ is the number of wiki pages in the corpus and $\#D_a$ the number of wiki pages containing the anchor text of the link.

For the file-to-file task links pointing to the same target t but having different spans sp are merged. We distinguish three different merging strategies, namely the highest score of the link, the average score of the link or simply by counting the number of links to a target t.

Our approach raises several questions to analyze. For gazetteer matching we focused on a very high recall by allowing fuzzy matching strategies, assuming that we can disambiguate them efficiently in the following step. In particular the question is how much noise added by the gazetteer can be resolved afterwards. Therefore we considered the following properties upon gazetteer matching:

- Case sensitive vs. case insensitive matching may impact different link categories. Named entities like persons, technical abbreviations (e.g. *AJAX, R*) may be perfectly identified by case sensitive matching, while case insensitive matching prefers more general concepts like *web applications* etc. Considering this parameter may outline differences between the different link categories.
- Filtering gazetteer entries based on identified part-of-speech: Some titles contain numbers or particular syntactic elements like brackets. Filtering gazetteer entries based on their part-of-speech should allow making matching considering those cases by removing non nouns.
- Filling the gazetteer list using page titles only or titles and anchors of a link raises questions towards how much information is added by anchors of links in the Wikipedia corpus and how noisy it is. From a statistical point of view we obtain around 1.7 million gazetteer entries by taking titles and anchors into account, achieving a very high recall. Our question herein is whether the anchor text is capable to disambiguate those additional entries or not.
- Selecting longest common sequence matching links from overlapping anchors: Through gazetteer matching the span *sp* of links may overlap. Our hypothesis here is that the longer the sequence of words, the more specific a link is. Therefore, those more specific links should be chosen.

The core question in the following disambiguation step is how to determine the anchor context. We investigate three different levels, namely sentence, topic (as a sequence of sentences) or the whole document. We did not assume that the orphan document is structured in any way and thus topics are detected automatically using the above mentioned C99 algorithm. However, analyzing the impact of structures is up to following research. Regarding the scoring of an link our main interest lies in the difference between the anchor IDF scoring scheme and the anchor context scoring scheme. In particular the question is how much useful context information surrounds a link and whether it is worth to consider this context information.

3.2 In-link Generation

In-link generation is in principle similar to out-link generation with the difference that in a first step we have to determine the source document d_j of a particular link. Again we utilize title matching for doing so, but in contrast to out-link generation the title is used as search string instead of gazetteer matching. Similarly to out-link generation we are determining different contexts, in this case best-entry-point contexts, to assign a score to a link. Again sentences, topics or the whole document serves as context. Given the nouns of this context as sequence $< w_1, \ldots, w_k >$ we are sending the following query to the backend:

$$\text{``}\mathit{title}\text{''}\ AND\ (w_1\ OR\ w_2 \ldots OR\ w_k)$$

where "*title*" indicates a phrase query for the title of the Wikipedia page. Again the score is calculated as outlined in equation 1.
From the result set we obtain a ranked list of possible link source candidates. If the context is different than the whole document, merging strategies are required to merge the ranked lists of the different contexts. Similarly to out-link generation we utilized the highest scoring source candidate, the average score of a source candidate or a simple counting scheme. Taking the n best source candidates is either the input for determining the best-entry-points or gives us already the result for the file-to-file linking task.

Using more fine grained contexts than the document level follows the intuition that a link points to a Wikipedia page because the author wants to address a particular aspect of the relationship and not all aspects of the target page. By considering those fine grained similarities we tried to address this aspect.

3.3 Best-Entry-Point Detection

Either in-link or out-link generation provides a list of best matching links including target page, source page and the span of a link. In the final step, best-entry-points are determined again based on document decomposition. Our hypothesis is that the best-entry-point in the link target has to be similar to the anchor context. Furthermore, if the title of the source page is contained in the link target, those parts of the target document are preferred entry points. Since we obtain a score for each entry point, results are ranked and the best five entry points are taken as result.

In particular, similarity is calculated using a simple vector space model with local TFIDF weighting. Given the link target t, the textual content of the target is preprocessed and decomposed into segments $t_{r,1} \dots t_{r,k}$. Segments are either sentence or topics. After filtering out all non-noun words, each segment is converted into a term vector. The weight of a term is calculated according to the TFIDF scheme, but based on the extracted segments, as:

$$w_{r,l} = tf_{r,l} * \log(1 + \frac{\#R}{\#R_l}) \tag{2}$$

where $w_{r,l}$ is the weight of term l in segment r, $tf_{r,l}$ is the number of times a term l occurs in segment r divided by all terms in segment s, $\#R$ is the number of segments in the target document and $\#R_l$ is the number of segments containing term l.

Similarly to the target segments, the anchor context in the source document - denoted as a - is also converted into a term vector by filtering all non-nouns and applying equation 2.

The ranking of best-entry-points is obtained by calculating the cosine similarity between anchor context $\overrightarrow{a}$ and all target segments $\overrightarrow{t}_{r,1} \dots \overrightarrow{t}_{r,k}$ and rank them accordingly. Segments containing the title of the anchor page are favored by increasing the similarity as follows:

$$s(\overrightarrow{a}, \overrightarrow{t}_{r,i}) = \begin{cases} title \in t_{r,i} : & (1 + \frac{\overrightarrow{a} \cdot \overrightarrow{t}_{r,i}}{\|\overrightarrow{a}\| * \|\overrightarrow{t}_{r,i}\|})/2 \\ title \notin t_{r,i} : & \frac{\overrightarrow{a} \cdot \overrightarrow{t}_{r,i}}{\|\overrightarrow{a}\| * \|\overrightarrow{t}_{r,i}\|} \end{cases}$$

Best entry points are returned as starting point of the text segment since we assume that a reader does not want to start reading in the middle of a sentence or paragraph.

4 Implementation and Evaluation Details

As outlined above, Lucene [5] has been used as search backend and OpenNLP [1] for preprocessing. All algorithms are developed in Java, including the gazetteer component. Since our approach, at least for out-link detection, heavily relies on gazetteer matching the question is whether a gazetteer with low runtime and low memory resource consumption is feasible. In our FSM approach the gazetteer with titles and anchors consisted of around 1.7 million entries and used up around 800 MB main memory. Additionally, gazetteer entries may be distributed using distributed computing techniques like Map & Reduce [4] and thus scaling up is possible in our approach.

Runtime behavior also satisfies interactive requirements. On a dual core laptop with 4GB of main memory file-to-file runs took around 64 minutes using the more complex anchor context scoring - that is around 1.7 documents per second. After finding the link candidates, best-entry-point matching does not increase runtime complexity. Thus, the overall process can be seen as computational tractable and scalable.

During the development we did internal benchmarking for file-to-file and anchor detection using the TREC evaluation program trec_val. Those results differ strongly from the released preliminary results [3] so we assume that there has been an error in our submission format. However, both numbers are outlined in this section. Furthermore, results will be corrected in the post-proceedings after the release of the evaluation tool.

The runs can be differentiated in file-to-file in-link/out-link generation, out-link anchor detection and best-entry-point detection. File-to-file runs are evaluated on the 6.600 topics defined by the organizers. Out-link anchor detection and best-entry-point detection are run on the 50 topics defined by the participants. After the development of our algorithms we analyzed the different algorithmic properties by taking the available ground truth of the file-to-file and anchor detection runs. This allowed evaluation of all runs but the best-entry-point runs.

For out-link generation we distinguished between the title only vs. title and anchor matching, case sensitive vs. case insensitive matching (CS), longest common sequence (LCS) matching, different document segmentation level (DSL) as well as the two different ranking schemes, namely the anchor IDF and the anchor context score. Results of the internal runs including the official map of selected runs for file-to-file out-link generation are depicted in table 1, for anchor generation in the best-entry-point run in table 2. Figure 1 compares our out-link runs to the best runs of the other LTW track participants.

Table 1. Results for Out-link Generation File-to-File

Title Only	LCS	CS	Scoring	Segmentation	$\text{MAP}_{\text{intern}}$	$\text{MAP}_{\text{official}}$
true	false	false	anchor context	document	0.548	0.1129
true	true	false	anchor IDF	NA	0.5038	0.1407
true	false	true	anchor context	document	0.471	NA
true	true	true	anchor IDF	NA	0.4508	NA
false	true	true	anchor IDF	NA	0.4392	NA
true	false	true	anchor context	topic	0.4258	NA
false	true	false	anchor IDF	NA	0.4215	NA
false	false	true	anchor context	document	0.3827	NA
false	false	true	anchor context	topic	0.3809	NA
false	false	true	anchor IDF	NA	0.3478	NA
true	false	true	anchor context	sentence	0.3369	NA

Overall our findings for out-link generation can be summarized as follows:

- *Gazetteer Matching*: Using only nouns in the matching process did not have strong impact in our experiments and therefore detailed results have

[3] Preliminary results on file-to-file and anchor linking released on 28th of November

Table 2. Results for anchor out-link generation

Title Only	LCS	CS	Scoring	Segmentation	$\text{MAP}_{\text{official}}$
true	false	false	anchor context	document	0.2370
true	false	false	anchor context	topic	0.2039
true	true	false	anchor IDF	NA	0.2044

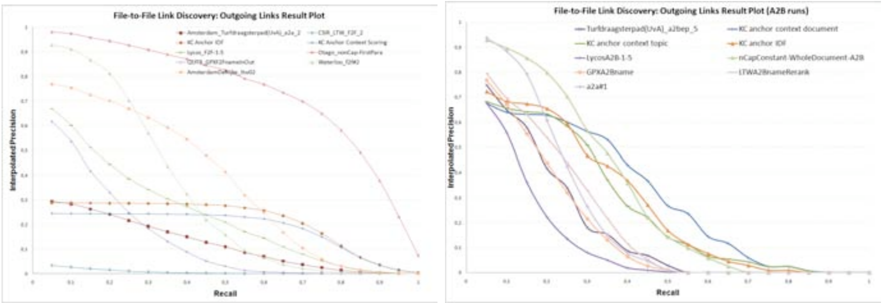

Fig. 1. Precision-Recall curve of our outlink file-to-file runs (left) and anchor-to-bep runs (right) compared to the best runs of other participants.

been omitted here. Taking only the longest common sequence of overlapping matches has an impact on the anchor TFIDF scheme but not on anchor context scoring. Our assumption is, that anchor context scoring is able to disambiguate overlapping matches while this is not the case for anchor TFIDF. Overall, case insensitive matching yields to better results than case insensitive matching.

- *Document Segmentation and Scoring Schemes*: Surprisingly using the whole wiki page as context worked out to be best followed by using automatically detected topics. Considering sentences as anchor context did not perform well. However, all except of the document based anchor context scoring stayed behind simple anchor IDF matching. Also, noise added by considering title & anchors during the matching step could not be resolved by either scoring scheme yielding to the conlcusion to exclude anchor texts in the matching step.
- *Merging Strategies*: Maximum and average merging seem to work well compared to just counting the score. However, more runs on different parameter variations have to be done in order to get a more concrete picture on the influence of merging strategies.

For in-link generation different document decomposition strategies have been analyzed. In addition we compared whether using the title as OR instead of a phrase query increases or decreases accuracy. Anchors are again identified by matching the title of the orphan page. Title matching has been done case insensitive. Results for file-to-file runs are outlined in table 3 and for anchor generation of in-links in table 4. Figure 2 compares our in-link runs to the best runs of the other track participants.

Table 3. Results for in-link generation file-to-file

Title as OR Query	Segmentation	$\mathrm{MAP}_{\mathrm{intern}}$	$\mathrm{MAP}_{\mathrm{official}}$
false	document	0.6355	0.5300
false	sentence	0.5938	0.5369
false	no context	0.5938	NA
true	document	0.5066	NA
true	sentence	0.4088	NA
true	no context	0.4088	NA

Table 4. Results for in-link anchor generation

Document Segmentation	$\mathrm{MAP}_{\mathrm{official}}$
Document	0.1685
Sentence	0.1663
Topic	0,1391
No context	0,1386

Results show that again using the whole document as context yields the best results, while sentences or topics as context perform similar to only searching for titles. Making an OR query out of the title did worsen results significantly.

After the release of the evaluation tool we plan to evaluate the differences between our internal and the official runs and evaluate best-entry-point results. Furthermore, some parameter combinations have not been evaluated till now and by using significance testing we plan to test the influence of different parameter combinations as well as take a more detailed look into the influence of particular parameters onto specific topics.

5 Conclusion

Based on the usage of standard IR and IE technology our results seem to be promising. However, numbers presented here have to be taken with care since internal benchmarks differ from official results and we assume that there are errors in our submission format. Besides this, the preliminary results point out that simple scoring strategies like our anchor IDF yield to reliable results while not being able to disambiguate the context of links as good as taking the whole document as anchor context into account. Regarding the context of a link our experiments showed that taking the whole page as context turns out to be best

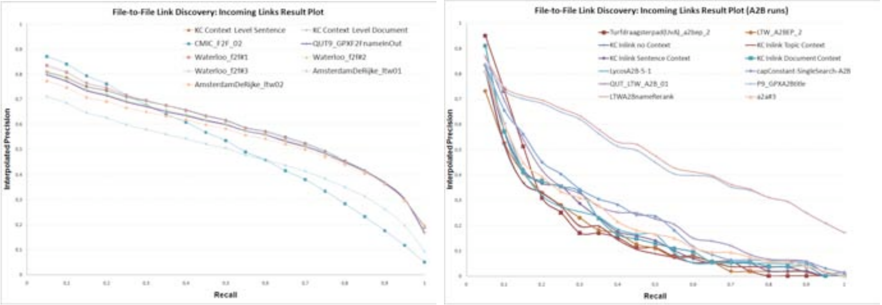

Fig. 2. Precision-Recall curve of our in-link file-to-file runs (left) and anchor-to-bep runs (right) compared to the best runs of other participants.

suited for outlink as well as for in-link detection. Although results look promising title matching and the subsequent scoring scheme seem to be too tightly coupled. For example without filtering out overlapping title matches by taking only the longest common sequence anchor TFIDF performs worse.

Such interwoven parameters are an indication for the need to decouple the different linking steps and the use of machine learning approaches in each step. Using machine learning seems to be very promising as shown in very recent work [7]. Also, we completely ignored the pre-given structure of Wikipedia pages like section and paragraphs. Maybe those structures allow a better definition of anchor contexts and best-entry-point contexts.

We think that two important aspects should be covered by the Link-the-Wiki track next year. One is the automatic labeling of the link types between Wikipedia pages. For example the page Berlin linking to Germany marks a part-of relationship while a link between Berlin and Capital marks a is-a relationship. Developing methods for automatically identifying such relationships for Wikipedia links may have a huge practical but also theoretical impact in boosting new technology like semantic wikis. The second possible extension regards linking to documents outside the Wikipedia, i.e. determining external links. Again we think there is a practical impact and yielding to new search paradigms with Wikipedia in its core.

Acknowledgement

The Know-Center is funded within the Austrian COMET Program - Competence Centers for Excellent Technologies - under the auspices of the Austrian Ministry of Transport, Innovation and Technology, the Austrian Ministry of Economics and Labor and by the State of Styria. COMET is managed by the Austrian Research Promotion Agency FFG. We also want to thank the organizer of the Link-the-Wiki track for their effort in making the track possible.

References

1. T. & Bierner G. Baldridge, J.; Morton. Opennlp: The maximum entropy framework. Web Site http://maxent.sourceforge.net/about.html, 2001. , last visited June 2008.
2. Kalina Bontcheva, Hamish Cunningham, Diana Maynard, Valentin Tablan, and Horacio Saggion. Developing reusable and robust language processing components for information systems using gate. In *DEXA '02: Proceedings of the 13th International Workshop on Database and Expert Systems Applications*, pages 223–227, Washington, DC, USA, 2002. IEEE Computer Society.
3. Freddy Y. Y. Choi. Advances in domain independent linear text segmentation. In *Proceedings of the first conference on North American chapter of the Association for Computational Linguistics*, pages 26–33, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.
4. Jeffrey Dean and Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters. *Commun. ACM*, 51(1):107–113, 2008.
5. Erik Hatcher and Otis Gospodnetic. *Lucene in Action (In Action series)*. Manning Publications, December 2004.
6. Darren Wei Che Huang, Yue Xu, Andrew Trotman, and Shlomo Geva. Overview of inex 2007 link the wiki track. *Focused Access to XML Documents*, LNCS 4862:373–387, 2007.
7. David Milne and Ian H. Witten. Learning to link with wikipedia. In *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge mining*, pages 509–518, New York, NY, USA, 2008. ACM.
8. Fei Wu and Daniel S. Weld. Autonomously semantifying wikipedia. In *CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 41–50, New York, NY, USA, 2007. ACM.