# Context based 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 can be done either content based by exploiting word similarities or structure based by exploiting characteristics of the link graph. Our approach focuses on a content based strategy by detecting Wikipedia titles as link candidates and selecting the most relevant ones as links. The relevance calculation is based on the context, i.e. the surrounding text of a link candidate. Our goal was to evaluate the influence of the link-context on selecting relevant links and determining a links best-entry-point. Results show, that a whole Wikipedia page provides the best context for resolving link and that straight forward inverse document frequency based scoring of anchor texts achieves around 4% less Mean Average Precision on the provided data set.

**Key words:** INEX, Link-the-Wiki, Context Exploitation

## 1 Introduction

This paper outlines the efforts taken by the Know-Center Graz in the Link-the-Wiki Track of INEX 2008. The track focuses on automatically linking an orphan page to already existing Wikipedia pages (outgoing links; out-links) and from already existing Wikipedia pages to the orphan page (incoming links; in-links). In contrast to last years focus on identifying source and target pages of a link, this years track also includes the identification of anchor position and best-entry-points (BEP). Anchor positions mark the character position of a link in the source page; best-entry-points in the target page.

In last years Link-the-Wiki Track [6], matching page titles for identifying link candidates have been quite successful [4]. It was shown that without considering contextual information around the link, reasonable results could be achieved; a fact supported also from outside the INEX community [10]. Besides page titles, link structure provides valuable information. In [7] an algorithm using anchor texts and link structures achieved a very high accuracy. However, non of these

approaches took the context of a link, i.e. its surrounding text, into account, while [4] argued on the potential of such approaches.

In our approach we evaluate the potential of different context types to calculate the relevance of a possible link candidate. Link candidate identification itself utilizes word sequence matching based on a finite state machine gazetteers. Thereby, entries of the gazetteer contain not only the title of a Wikipedia page, but also anchor texts, similar to work reported in [7]. Link candidates via gazetteer matching are ranked subsequently using different context types, i.e. different ranges of words surrounding the anchor. This context based relevance should allow a more precise selection of correct hyperlinks hopefully removing high frequent, irrelevant links like for example "The" or "Are".

Our major contribution is an in depth analysis of different context types compared to straight forward, context free scoring mechanisms. Besides the official runs we also present a detailed parameter study using the "trec_val" evaluation tool and t-tests for estimating the influence of different parameter sets and syntactic matching properties like case sensitivity.

Experiments are evaluated on the Wikipedia XML Corpus consisting of 659,413 Wikipedia pages and split into two test sets by the track organizers. The *file-to-file test set*, has around 6.600 test documents with existing Wikipedia links as ground truth. The *anchor-to-bep test set* consists of around 50 manually annotated topics. The candidate page for automatic linking, a wiki page having all links removed from, is called an *orphan page*. 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.

In the following, section 2 outlines the corpus preparation and preprocessing and defines the anchor context types. This anchor context is used in section 3 to explain our link selection and scoring mechanism. Experiments, official results as well as internal parameter studies are shown in section 4, followed by the conclusion in section 5.

## 2 Preprocessing & Context Types

The Wikipedia corpus is indexed using the open source search engine Lucene [5] applying standard stop word removal and stemming. For each Wikipedia page the title and all anchors of links pointing to this page are extracted and stored as gazetteer list. For matching, this list transform into a finite state machine (FSM) consisting of three states. A start state serving as entry point, intermediate states retaining the structure of the FSM, and final states containing the URL of the Wikipedia page. Transitions between states of the FSM consist of words occurring in a gazetteer entry. Beginning at the start state, transitions are followed recursively if the transitions word occurs in the word sequence to match. If upon matching 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 a reasonable speed.

Orphan pages are preprocessed using the OpenNLP toolkit [1]. Preprocessing includes tokenization, sentence detection and part-of-speech tagging. Afterwards, the document is segmented into non-overlapping parts, defining the context for the following relevance calculations. In our experiments we distinguish between the following context types:

- *Document:* Most straightforward, the whole document is taken as context.
- *Section:* Sections are provided via the XML-Schema and correspond to the Wikipedia sections of articles.
- *Paragraph:* Similar to sections, paragraphs are also provided via the XML Schema.
- *Topics:* Topics are automatically annotated based on sentence clustering. Blocks of similar sentences are found and annotated as topic using the well known C99 segmentation algorithm[2].
- *Sentences:* Sentences are obtained from the sentence detector of the OpenNLP pipeline and serve as smallest possible context.

The context based linking strategies introduced in the next section exploit those context types in order to determine the relevance of a link.

## 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 (i.e. character based start and end position) 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, differentiating between out-link and in-link generation. While both follow the same conceptual approach, their implementation varies for reducing time complexity in the in-link generation step.

### 3.1 Out-link Generation

Out-link generation starts with preprocessing the orphan document $d_o$ as outlined in section 2. Matching the content of the document with the FSM- gazetteer returns a set of possible out-link candidates $O$, whereby for each link $l_i \in O$ we know its source $s_i$, its target $t_i$ and its span $sp_i$. For each link we determine the

*anchor context*, that is the context the link span is contained in. 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 file name of the page) as AND query part. Thus, for all links contained in the span of 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 link. 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 and its surrounding text we receive different scores hopefully disambiguating the tree data structure from the forest tree.

Besides context based scoring method an evaluation scheme solely based on the inverse document frequency of an anchor text is used for comparison reasons. The rational behind is that high frequent anchor texts like “The” or “Are” 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 $t$.

### 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 the orphan document to assign a score to a link. Given the nouns of this context as sequence $< w_1, \ldots, w_k >$ we are sending the following query to the backend:

$$\text{“}title\text{”}\ AND\ (w_1\ OR\ w_2 \ldots OR\ w_k)$$

where “*title*” indicates a phrase query for the title of the orphan 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. As for out-link generation, we calculate the relevance either as the highest relevance of a link, the average relevance of a link or simply by counting the occurrences of a link. 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.

### 3.3 Best-Entry-Point Detection

Both in-link and 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 the link context. 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 $t_{r,1} \ldots t_{r,k}$. Segments are either

sentences or topics and correspond to the context defined in section 2. 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:

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

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 $\overrightarrow{t}_{r,1} \ldots \overrightarrow{t}_{r,k}$ and rank them accordingly. Segments containing the title of the anchor page are favored by increasing the similarity as follows:

$$s(\overrightarrow{a}, \overrightarrow{t}_{r,i}) = \begin{cases} title \in t_{r,i}: & (1 + \frac{\overrightarrow{a} \cdot \overrightarrow{t}_{r,i}}{\|\overrightarrow{a}\| * \|\overrightarrow{t}_{r,i}\|})/2 \\ title \notin t_{r,i}: & \frac{\overrightarrow{a} \cdot \overrightarrow{t}_{r,i}}{\|\overrightarrow{a}\| * \|\overrightarrow{t}_{r,i}\|} \end{cases} \tag{3}$$

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 [3] 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.

The runs can be differentiated in file-to-file in-link/out-link generation, anchor detection and best-entry-point detection. File-to-file runs are evaluated on the 6.600 topics defined by the organizers. Anchor detection and best-entry-point detection are conducted on the 50 topics defined by the participants. After the development of our algorithms we did an in depth parameter analysis by taking the available ground truth of the 6.660 topics test set and evaluated file-to-file and anchor-to-file runs on it. This allowed an in depth evaluation of all runs but the manually assessed 50 topic based anchor-to-bep runs.

### 4.1 Parameter Analysis

Basically our experiments are focused on analysing the following parameters:

- Case sensitive ($CS$) matching distinguished between considering the case in gazetteer matching or not.
- Longest Common Sequence Matching ($LCS$) removed overlapping gazetteer annotations by taking those annotations with the longest common sequence of tokens.
- *Title only* matching only considers page titles in the gazetteer while otherwise anchors of links are also included in the gazetteer list.
- The *context* level determined the type of context to use for the anchor context scoring scheme. If no context was provided the anchor IDF scoring scheme was used.
- For the file-to-file runs 3 different merging strategies - maximum, average and count- for aggregating anchors on the file level have been considered

Permutation of the different parameters yielded 120 test runs for the file-to-file task and 40 test runs for the anchor-to-file task. Since in-link creation is conceptually similar, we restricted the parameter analysis task to out-link detection only. In order to cope with the large number of runs, statistical significance testing was used to determine the influence of the different parameters.

For determining the most influential parameters, we started determining statistically significant differences between runs using a one-sided paired t-test [9]. Statistically significant differences allow us to calculate a parameter value's "success rate", defined as how often a run with the particular parameter value is significantly better than all other runs. More formally, given $B(r_i)$ resp. $W(r_i)$ as the number of runs where run $i$ is significantly better resp. more worse and given $R_{p_a=v}$ as the set of runs where parameter $p_a$ has value $v$, the success rate $S_{p_a=v}$ of value $v$ for parameter $p_a$ is calculated as

$$S_{p_a=v} = \frac{\sum_{r_i \in R_{p_a=v}} B(r_i)}{\sum_{r_i \in R_{p_a=v}} B(r_i) + \sum_{r_i \in R_{p_a=v}} W(r_i)} \tag{4}$$

By ranking parameter values according to their success rate the most influential parameter value, i.e. those parameter values most often participating in successful run, can be estimated. In other words, by selecting a parameter value with a

high success rate it is very likely that this run will perform good, independent of the other parameter values.

By analysing file-to-file runs it turned out that context based evaluation strategies had an overall success rate of around 67% outperforming all other parameters. Also, only using page titles yields to a higher success rate of around 62% than using gazetteers based on anchor texts. Merging links using the maximum score also turned out to outperform the average score and count based merging strategy. Case sensitive vs. case insensitive matching as well as longest common sequence matching did not have a huge impact on the performance of a run. Analyzing the context parameter for file-to-file runs more closely showed that taking the whole document gives a success rate of 96%. Thus, nearly every time the whole document is used as anchor context the run outperforms all other runs. Topic detection also turned out to have a high success rate (82%), outperforming sentences, paragraph and sections as topic. However, for the later two it must be noted that a large number of queries did not have sections or paragraphs assigned, thereby biasing the results. Similar results are achieved by the anchor-to-file task. However, different to the file-to-file runs anchor idf based scoring turned out to be as good as context based scoring.

**Table 1.** Results for out-link Generation for the file-to-file run with 6600 orphan test pages and the anchor-to-file task with 50 orphan pages. Runs with no context used the anchor IDF scoring method. NA depicts measures not available due to missing ground truths.

| Task | Title Only | LCS | CS | Context | $MAP_{intern}$ | $MAP_{official}$ | $MAP_{reeval}$ |
|---|---|---|---|---|---|---|---|
| file-to-file (6.600) | true | false | false | document | 0.548 | 0.1129 | 0.516 |
| file-to-file (6.600) | true | true | false | none | 0.5038 | 0.1407 | 0.475 |
| file-to-file (6.600) | true | false | true | document | 0.471 | NA | NA |
| file-to-file (6.600) | true | true | true | none | 0.4508 | NA | NA |
| file-to-file (6.600) | false | true | true | none | 0.4392 | NA | NA |
| file-to-file (6.600) | true | false | true | topic | 0.4258 | NA | NA |
| file-to-file (6.600) | false | true | false | none | 0.4215 | NA | NA |
| file-to-file (6.600) | false | false | true | document | 0.3827 | NA | NA |
| anchor-to-file (50) | true | false | false | document | NA | 0.2131 | 0.2350 |
| anchor-to-file (50) | true | false | false | topic | NA | 0.2643 | 0.2908 |
| anchor-to-file (50) | true | false | false | sentence | NA | 0.2309 | 0.309 |
| anchor-to-file (50) | true | true | false | none | NA | 0.2873 | 0.3130 |

### 4.2 Official Results

We submitted 2 runs for the file-to-file task for comparing the best anchor context method with the context free anchor IDF approach. For anchor-to-bep we submitted a combination of 3 different out-link generation and 4 different in-link

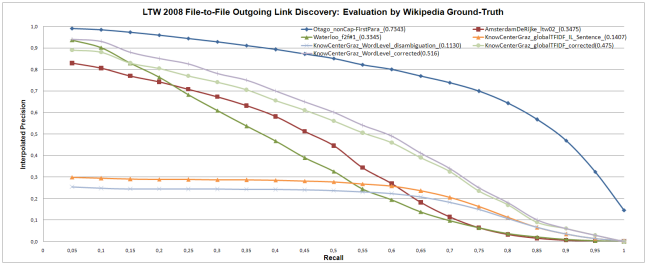


**Fig. 1.** Precision-Recall curve of our out-link file-to-file runs comparing the official runs with the corrected runs and with the official results of the best LTW 08 runs.

generation approaches again distinguishing between context based and context free approaches. For the remaining parameters we took the best choices obtained by the parameter analysis.

**Table 2.** Results for in-link generation file-to-file

| Parameters | | file-to-file | | | anchor-to-bep |
|---|---|---|---|---|---|
| Title as OR Query | context | $MAP_{intern}$ | $MAP_{official}$ | $MAP_{reevaluated}$ | $MAP_{official}$ |
| false | document | 0.6355 | 0.5300 | 0.625 | 0.2384 |
| false | sentence | 0.5938 | NA | NA | 0.1895 |
| false | topic | NA | NA | NA | 0.2619 |
| false | no context | 0.5938 | 0.5369 | 0.606 | 0.1968 |
| true | document | 0.5066 | NA | NA | NA |
| true | sentence | 0.4088 | NA | NA | NA |
| true | no context | 0.4088 | NA | NA | NA |

MAP of the official and the best internal runs for *out-link generation* are depicted in table 1. Due to an error in the submission format, our official runs scored much more worse than our internal benchmarks. We corrected the submission error on the submitted files and re-evaluated the results. Those re-evaluated mean average precisions are depicted as $MAP_{reeval}$ in the tables. Figure 1 shows the precision recall curve for the out-going links comparing official with the in-official results for the submitted runs and comparing the official runs with the best runs in the Link-The-Wiki track. By correcting the submission format our runs performed quite well and would be ranked $2^{nd}$. It can be observed that considering the context of a link improves mean average precision by around 4%. While the increase is significant, we would have expected a larger increase through the more complex anchor context scoring mechanisms. Also, if the anchor context is other than the whole document, the differences becomes smaller and is nearly negligible.

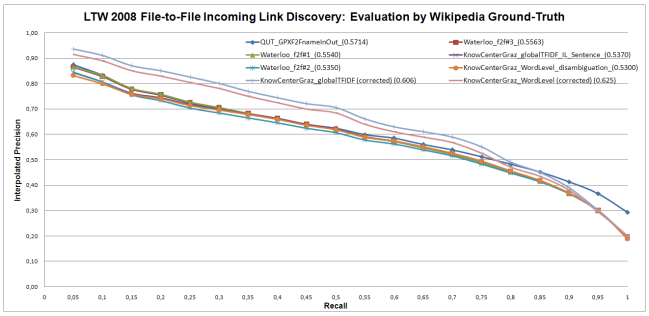


**Fig. 2.** Precision-Recall curve of our in-link file-to-file runs comparing the official runs with the corrected runs and with the official results of the best LTW 08 runs.

Similar performance figures can be observed for *in-link detection*, as shown in table 2. The difference between the best context scoring method - again a document based context - and the context free scoring method is smaller, with around 2% on the re-evaluated runs. Overall file-to-file in-link generation did quite well compared to the best runs of the track (see figure 2). The original runs have been ranked third, while the re-evaluated runs achieved the highest map. Also, precision-recall curves provide high precision values over large parts of the recall.

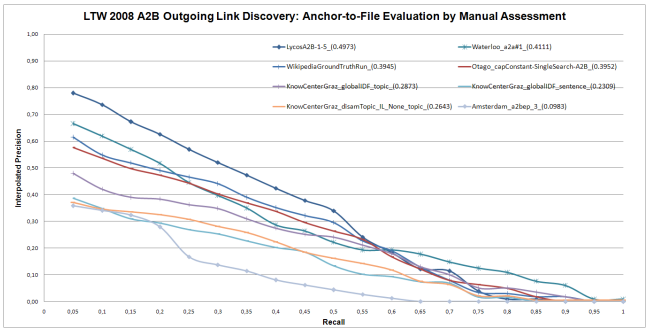


**Fig. 3.** Precision-Recall curve of our anchor-to-file runs compared to the best runs of other participants.

*Anchor-to-file results* of the manually assessed 50 topic task are provided in table 1 and precision-recall curves compared to the best other runs is shown in figure 3 . Overall, the performance of anchor detection was lower than what could be expected from file-to-file matching. Also, re-evaluation did not provide a huge increase due to the smaller size of links per topic. For the manually assessed anchors it seems that our context based scoring scheme does not score

well, especially since our top scoring run is based on no context at all.

*Anchor-to-bep evaluation*, depicted in table 2 and figure 4, shows very low mean average precision compared to the other participants. In contrast, file-to-bep evaluation performed well with an map of 12.219 compared to the best map of 20.79 from Lycos and being very close to the second best group of runs from Otago. Since the evaluation measure penalizes missing the exact position linearly with the number of characters, only those runs using sentences as context achieved a good BEP score. Topic based runs performed considerable worse. Overall, results on the manually assessed runs point toward the hypothesis that vector space based approaches using words surrounding a link are not discriminative enough for achieving reasonable accuracy values.

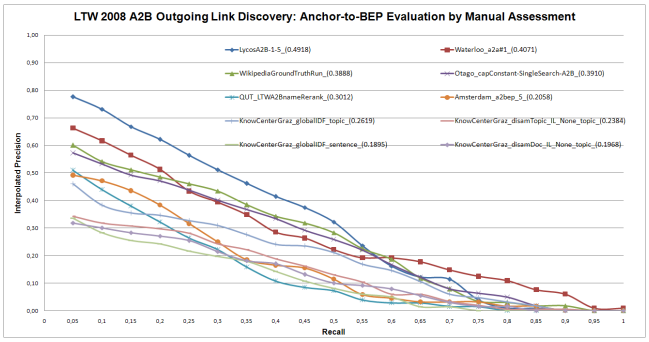


**Fig. 4.** Precision-Recall curve of our anchor-to-bep runs compared to the best runs of other participants.

## 5 Conclusion

In this paper we have outlined context based methods for automatically detecting links between Wikipedia pages. Experiments showed that considering the context of an link increases precision by around 4%. However, the choice of the type of context is critical. The whole document seems to be best suited as anchor context, followed by automatically detected topics. Predefined document structures like sections and paragraph are bad context choices, decreasing accuracy below the straightforward IDF approach. Constructing gazetteers from page titles only seem to be more appropriate than using anchor texts, from which follows that using context based scoring schemes hardly resolves noisy links introduced by anchor texts. Results obtained by the experiment point toward the hypothesis that vector space based approaches using words surrounding a link are not powerful enough, especially for anchor and BEP detection. Hence, sequence based approaches, language models or link based methods (c.f. [7]) may

be required for achieving reasonable accuracies.

In the future we plan to focus more on machine learning based approaches. As shown in recent work [8], machine learning can achieve rather high user judged accuracy while retaining parameter robustness. Another fruitful future challenge is the automatic labeling of link types. 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. Automatically identifying such relationship types may have both, a huge practical as well as a huge theoretical impact in the context of semantic wikis.[3]

## 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. 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.
3. Jeffrey Dean and Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters. *Commun. ACM*, 51(1):107–113, 2008.
4. Shlomo Geva. Gpx: Ad-hoc queries and automated link discovery in the wikipedia. pages 404–416, 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. Kelly Y. Itakura and Charles L. Clarke. University of waterloo at inex2007: Adhoc and link-the-wiki tracks. pages 417–425, 2008.
8. 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.
9. Mark D. Smucker, James Allan, and Ben Carterette. A comparison of statistical significance tests for information retrieval evaluation. In *CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 623–632, New York, NY, USA, 2007. ACM.
10. 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.

[3] *Acknowledgement:* The Know-Center is funded within the Austrian COMET Program - Competence Centers for Excellent Technologies - under the auspices of the Austrian Federal Ministry of Transport, Innovation and Technology, the Austrian Federal Ministry of Economy, Family and Youth and by the State of Styria. COMET is managed by the Austrian Research Promotion Agency FFG.