Stealing Anchors to Link the Wiki

Philipp Dopichaj, Andre Skusa, and Andreas Heß

Lycos Europe GmbH
Carl-Bertelsmann-Str. 29
P. O. Box 315
33311 Gütersloh
Germany
dopichaj@acm.org, andre.skusa@googlemail.com, mail@andreas-hess.info

Abstract. This paper describes the Link-the-Wiki submission of Lycos Europe. We try to learn suitable anchor texts by looking at the anchor texts the Wikipedia authors used. Disambiguation is done by using textual similarity and also by checking whether a set of link targets "makes sense" together.

1 Introduction

In this paper, we describe the Link-the-Wiki submission of Lycos Europe. Details about INEX and the Link-the-Wiki track are given elsewhere in these proceedings, so we do not repeat them here. In the following, we use *new text* to refer to the text which should be linked (conceptually, this is a text entered by a user of the platform without any links; the aim of the system is to support the user to find suitable links). We use *anchor text* or *anchor* to refer to the link label, that is, the clickable part of the text that links to a *target page*.

Our approach to the Link-the-Wiki task is based on that described by Itakura and Clarke [2]: All existing anchor texts from the training collection are indexed along with their link targets, and the new text is scanned for these anchor texts to find links.

The main difference is that we try to select the best-matching target dynamically whereas Itakura and Clarke use a static mapping from anchor text to target – the target is always the page most frequently referenced by the anchor. For example, in a text about computers, the anchor *Apple* is more likely to refer to the page *Apple Computers* than to the page *Apple Records*. We use heuristics based on text similarity and link structure to determine which of the potential targets is the most likely real target.

Finding outgoing links is done in the following steps:

1. The potential anchor texts are identified. The chosen anchor texts do not overlap, and each anchor text has one or more potential targets associated with it.
2. For each potential anchor text, a ranking of the potential targets *in the context of the new text* is performed. Furthermore, general statistical information obtained at indexing time – like absolute frequency – is used.

Our main focus is finding outgoing links, as opposed to finding incoming links from existing documents to the newly-added content. Outgoing links are determined in two main steps that will be described in the following sections:

1. Finding the parts of the texts that should serve as links to other documents (*anchor texts*).
2. Finding the correct target pages for the anchor texts in case of ambiguities.

The second point means that even if a given anchor text is known to refer to *some* other document, it is not necessarily known to *which* article it refers.

2 Preparations for Finding Outgoing Links

This section describes how potential anchor texts are found in the new text and also what index structures are needed to support this.

2.1 Finding Potential Anchor Texts

The first step toward identifying links in a new document is to find potential anchors; this is done by searching for occurrences of the training anchors in the new text. We give preference to longer anchor texts: For example, in the example text from figure 1a, we have the sequence *Mac OS X v10.2*. Potential anchors include *Mac*, *Mac OS*, and *Mac OS X v10.2*; here, the last one is the longest anchor text, so it is selected. In case of overlapping anchor texts, the anchor occurring earlier is selected.

Apple bundled a similar program, Sherlock 3 , with Mac OS X v10.2 .

(a) Input text.

[[Apple]] bundled a similar program, [[Sherlock 3]] , with [[Mac OS X v10.2]] .

(b) Selected anchors.

Apple:	Apple Computer, APPLE, Apple Records, Apple (album), Apple II family, Malus, Apple Store (retail), Apple (super mario), Yabluko, Apple I
Sherlock 3:	Sherlock 3
Mac OS X v10.2:	Mac OS X v10.2

(c) Possible targets of the selected anchors

[[Apple Computers|Apple]] bundled a similar program, [[Sherlock 3]] , with [[Mac OS X v10.2]] .

(d) Final linked text, with the *Apple* anchor directed to *Apple computers*.

Fig. 1: Processing of an input text from the Wikipedia article *Karelia Watson*.

Using word boundaries as implemented in the Java regex package for anchor detection does not work for two reasons:

- Due to the idiosyncrasies of the INEX Wikipedia collection, spurious spaces are inserted or removed around markup, even in the middle of words, so word boundaries cannot be trusted.
- Anchors may only partly cover a given word; this is bad style, but there are instances where *child* as part of *children* is linked to the corresponding article. For other languages like German, compound words can be formed without spaces, so this might happen more frequently.

Although the second case is rare – especially in the English version of Wikipedia used for INEX –, the first reason is sufficient to justify the decision not to analyze word boundaries.

The result of this stage is a collection of non-overlapping anchor texts that might be turned into links. Based on the training data set, we know for each anchor set the possible target pages as well as the absolute frequency of references to a certain target page under the given name. We now have to develop a ranking of the targets for every potential anchor.

2.2 Reducing the Size of the Anchor Index

Our approach requires statistics about the existing links in the training collection. We examine every link in the collection and store the anchor text along with the target page's ID. Then, we count the number of occurrences for each anchor text/target page pair to see how often a given anchor text is used to refer to the given page.

This information is sufficient input for our approach, but to both keep the index size small and remove spurious entries, we remove all anchor text/target page pairs with one of the following properties:

- The length of the anchor text is less than 5 or greater than 60. Very long anchors include anchors like *Best Writing, Story and Screenplay Based on Factual Material or Material Not Previously Published or Produced*; they mostly refer to very specific page titles that are unlikely to occur in normal text. Short anchors are removed because they are usually ambiguous and they can lead to false positives.
- The anchor text refers to ten or more different pages. This implies that the anchor text is very general like, for example, *her father*.
- The anchor text occurs less than five times in the collection.

The numbers used were chosen in a rather ad-hoc fashion; further research is required to determine whether these numbers are good (or even whether the filtering is needed at all). We will test this once the results and evaluation tools are available.

3 Link Target Disambiguation

In many cases, anchor texts refer to only one possible target, like *Sherlock 3* in the example in figure 1c. However, the anchor text *Apple* from the same example shows that there is not always a one-to-one mapping of anchor texts to target pages, so the link detector has to make a choice. Furthermore, it may be necessary to remove spurious anchors.

One obvious problem is that anchor texts are frequently only sensible in the context in which they occur; for example, the anchor text "her father" refers to different persons depending on who "her" refers to. Since low-level information about the document frequency of terms is not available in our setup, we could not use Itakura and Clarke's formula for selecting anchors to index, so we implemented the simple heuristics from section 2.2.

The remainder of this section is based on the following values that influence the choice of which targets to use for a given anchor:

1. The rank of this target for this anchor, based on the total number of references;
2. the rank of the target page when doing a full-text search for the new article's title; and
3. the rank of the target page when doing a full-text search for the new article's full text (optional).

We chose to use a linear combination of these factors to obtain the final rank of a target.

3.1 Analysis of Anchor/Link Frequency

In absence of any other information, the link finder can still look at the *prior probability* of a given anchor text referring to a given target. This information can be obtained by analyzing the frequencies of the different target pages for a certain anchor text. For example, in the INEX collection, the anchor *Apple* refers to *Apple Computer* 399 times, to *APPLE* 83 times and to *Apple Records* 65 times, so in absence of any further information, *Apple Computer* is most likely the correct target.

3.2 Analysis of the Target Text

Simply using the frequency of targets in the training collection, however, does not take into account the context provided by the new document: for example, the text of the document should already give a strong indication whether the article is about computers or music. Thus, a straightforward approach is to calculate the *textual similarity* of the new text and the possible targets; if the new text and a target have a high similarity, it is likely that they are about the same general topic (like computers or music).

In our implementation, we implement this by doing a single full-text search for the complete new text respectively its title on an index that comprises the

Table 1: Target distribution of the anchor *Apple* (case sensitive).

Rank	Count	Target page
1	399	Apple Computer
2	83	APPLE
3	65	Apple Records
4	7	Apple (album)
5	2	Apple II family
6	2	Malus
7	1	Apple Store (retail)
7	1	Apple (super mario)
7	1	Yabluko
7	1	Apple I

full texts of all articles in the test collection. This results in a single ranked list of articles that are somehow related to the new text; for every anchor text that is found in the new text, the highest-ranked article from this list is chosen.

3.3 Analysis of the Link Structure

According to our observation, it is likely that the documents that are linked from the same source document are connected. This is because these pages typically share a main topic, so if two topics are mentioned (or pages are referenced) on the same source page, these topics are more likely to be connected than two randomly chosen topics. We can exploit this to find the correct link target among a set of candidates; for every such set, we determine how many links to the target pages for the *other* anchor texts exist. The more links exist, the more likely the target is to be the correct link target for this anchor.

Figure 2 demonstrates that the pages linked from a single page tend to be heavily connected. We can see that *APPLE* is not connected to the pages that are actually referenced from the source page at all and that *Apple Records* only has one link, whereas *Apple Computer* has many links in this cluster of pages.

The link analysis will not work properly if there is a very low number of targets (or, more generally, if the potential targets are mostly unconnected). In this case, the link finder should select potential targets even if they are isolated. The exact mechanism and threshold for this are the subject of future research.

3.4 Combination of these Approaches

Of course, it is possible to not only use these approaches in combination, but also to combine the evidence to obtain better quality. Since each of the approaches can be used to find a ranked list of possible targets for a given anchor text, we chose to use a weighted combination of the different ranks as the basis for the final decision. Given the example rankings from table 2, and the weights $w_1 = 1$ (anchor/link frequency), $w_2 = 5$ (text similarity), and $w_3 = 2$ (link analysis)

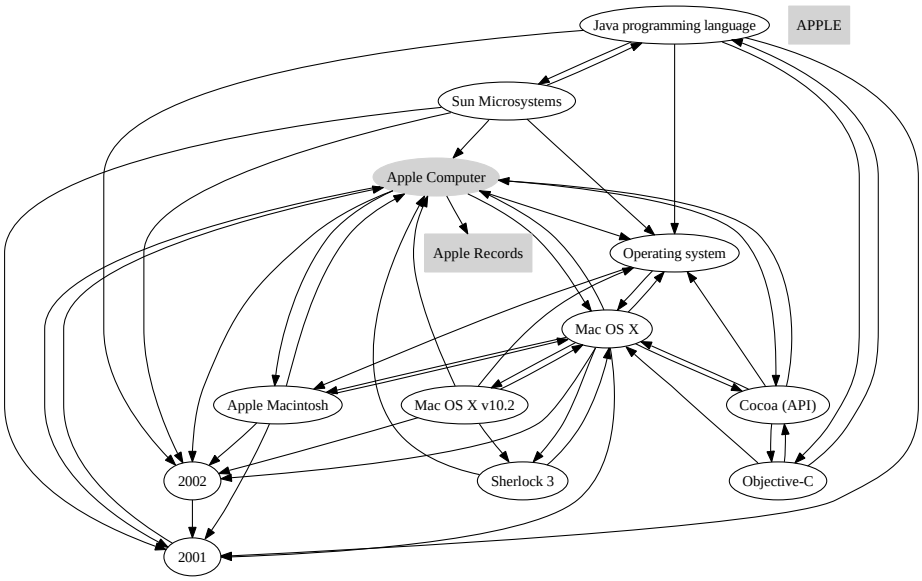

Fig. 2: The link network for pages linked from the page *Karelia Watson*. Our focus is on the shaded items, which are potential targets for the anchor text *apple*. The *Apple Records* and *APPLE* pages (in rectangles) are not linked from this page, but shown here for demonstration.

results in a final value of 12 for *Apple Computer*, of 13 for *Apple Records*, and of 24 for *Apple I*. Thus, in this case, the link target *Apple Computers* has the lowest combined rank and is selected as the final target. (Note that a higher weight value decreases the influence of the corresponding factor.)

Table 2: Example for combining the different aspects of target rankings.

Target	Anchor/link freq.	Text sim.	Link analysis
Apple Computer	1	2	1
Apple Records	2	1	3
Apple I	3	3	2

Since we did not finish the implementation of link-based target disambiguation in time, we only submitted runs using anchor/link frequency and text similarity. For text similarity, we search the full text of all articles for occurrences of the title of the new page to be linked. From a quality point of view, it would probably be better to search for the complete body text of the new article – otherwise we implicitly assume that the concept is already mentioned in the existing articles, although it does not have an article of its own. Unfortunately, the cost for doing this was prohibitive on our setup, so we had to settle for searching for the titles only. We used the different combinations of text similarity–anchor/link frequency weight, from equal weights for both (run LycosA2B-1-1), a weight of 5 for one and 1 for the other (runs LycosA2B-1-5 and LycosA2B-5-1). Furthermore, we submitted runs using only one of the two factors (runs LycosA2B-0-1 and LycosA2B-1-0).

Note that we do not actually calculate a best entry point in the target file – we always use the start of the document instead.

3.5 Limitations

One base limitation of our work is that we assume that the collection already contains a large number of related articles. As Huang et al. [1] note, this assumption does not hold for a batch upload of related articles where links between the articles are at least as important as links to or from the collection. Another potential problem is that the anchor texts that have been used by the authors might not be meaningful (for example, "click here").

We believe, however, that the approach can work well in the right circumstances. We plan to use it on a community platform about German history, with the anchors from the German Wikipedia as a training set. The results from preliminary tests are quite promising.

4 Finding Incoming Links

Our current implementation for finding incoming links is simplistic: we simply search for the new document's title in the full-text index to determine a ranked list of candidate sources. (Note that no phrase search if performed, so in effect the results may contain pages where the terms from the title occur out of order.) Next, the title of the new document is searched for in each candidate's text, and the first occurrence is added to the list of links, ordered by the search rank. Finally, all pages where the title is *not* found – this may happen if the title comprises several words – are added to the end of the list.

5 Results and Discussion

At the time of writing, the evaluation tools have not been made public yet, so our evaluation only includes the official results; this means that we cannot discuss the effect of the network analysis.

5.1 Anchor to File

Although the original task was to find the best entry points in the link targets, many participants (including Lycos) always used the start of the document as the best entry point. Because of this, the anchor-to-best-entry-point results were also evaluated as anchor-to-file results, ignoring the best entry points if available.

Figure 3 shows that it pays to use a combination of text similarity and anchor/link frequency; the runs using both features better the runs using only one of the features. Interestingly, the exact weights used do not affect the results significantly, an equal weight for both factors performs as well as a 1:5 weight ratio in favor of either factor. On the other hand, omitting text similarity leads to a much higher loss in precision than omitting anchor/link frequency weights.

As figure 4 shows, it betters both the best submitted runs by other organizations and even the Wikipedia ground truth for most of the precision-recall curve. (The Wikipedia ground truth does not get perfect results because apparently the assessors disagreed with the article authors about what constitutes a good link.) Minor deficits can be seen in the high-recall regions (starting around 0.6), where our method trails the maximum of the other submissions by a significant margin.

The results for the global measures mean average precision (MAP) and R-precision (see table 3) are inconclusive: whereas the MAP of our method is significantly higher than that of all other methods (arounf 0.49 compared to at most 0.42 for the others), including the Wikipedia ground truth, our R-precision (0.40448) is lower than that of the best run, Amsterdam_a2bep_5 (0.42146). The reason for this is unclear; these measures have generally been shown to be highly correlated in information retrieval.

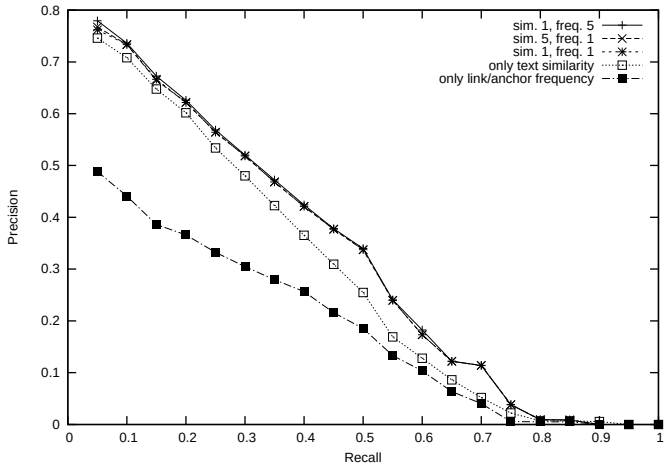

Fig. 3: Comparison of the Lycos runs for Anchor2BEP. "Best run" takes the maximum precision of all other runs for every recall (thus this is not an actual run). Clearly it pays to use both text similarity and anchor/link frequency.

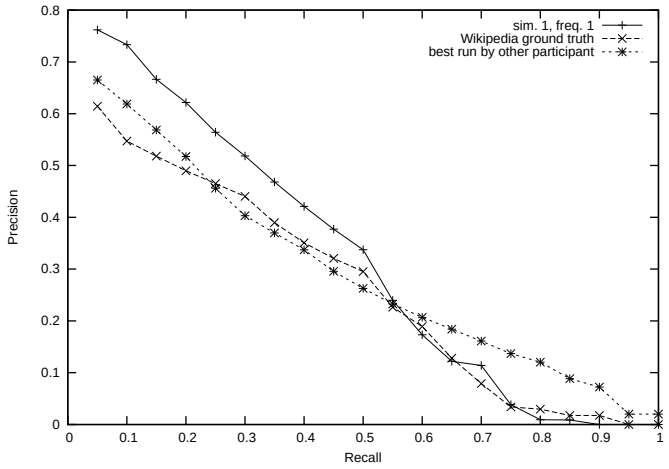

Fig. 4: Lycos versus Wikipedia ground truth and best of other submissions for Anchor2BEP.

Table 3: Official results for anchor-to-file evaluation. The highest numbers in each column are highlighted; for all given recall points as well as MAP, our run LycosA2B-1-5 has the best result. For R-precision, this run is surpassed by run Amsterdam_a2bep_5 and the Wikipedia ground truth.

Run ID	MAP	R-Prec	P5	P10	P20	P30	P50	P250
LycosA2B-1-5	*0.4973*	0.40498	*0.64400*	*0.61600*	*0.52100*	*0.44133*	*0.37840*	*0.07568*
LycosA2B-5-1	0.4931	0.40448	0.63600	0.61000	0.51900	0.44067	0.37800	0.07560
LycosA2B-1-1	0.4930	0.40448	0.63600	0.61000	0.51900	0.44133	0.37800	0.07560
LycosA2B-1-0	0.4708	0.37015	0.63200	0.59600	0.49900	0.41667	0.34280	0.06856
Waterloo_a2a#1	0.4111	0.33201	0.55600	0.50600	0.43100	0.36467	0.30560	0.06112
Otago_capConstant-SingleSearch-A2B	0.3952	0.35800	0.44400	0.45200	0.41400	0.37933	0.33240	0.06648
Otago_capConstant-TitleOnly-A2B	0.3952	0.35800	0.44400	0.45200	0.41400	0.37933	0.33240	0.06648
WikipediaGroundTruthRun	0.3945	0.40634	0.47600	0.46600	0.43500	0.39667	0.36520	0.07304
Otago_nCapConstant-WholeDocument-A2B	0.3896	0.35234	0.45600	0.46400	0.40200	0.36933	0.32040	0.06408
Waterloo_a2a#3	0.3874	0.34910	0.42800	0.44600	0.43200	0.39600	0.30560	0.06112
Waterloo_a2a#2	0.3355	0.39324	0.55600	0.50600	0.42600	0.35533	0.26680	0.05336
LycosA2B-0-1	0.3291	0.31201	0.35600	0.35800	0.32200	0.31933	0.31280	0.06256
QUT_LTWA2BnameRerank	0.3042	0.24854	0.39200	0.37200	0.32200	0.27733	0.22200	0.04440
QUT_GPXA2Bname	0.2912	0.22597	0.40000	0.37600	0.31100	0.26200	0.20480	0.04096
KnowCenterGraz_globalIDF_topic	0.2873	0.31495	0.25600	0.24600	0.26000	0.29533	0.35080	0.07016
KnowCenterGraz_disamTopic_IL_None_topic	0.2643	0.27409	0.24000	0.24400	0.25400	0.26533	0.30760	0.06152
KnowCenterGraz_globalIDF_sentence	0.2309	0.26539	0.22800	0.20800	0.20100	0.23400	0.28760	0.05752
KnowCenterGraz_disamDoc_IL_None_topic	0.2131	0.25125	0.14400	0.17000	0.20600	0.22267	0.28640	0.05728
Amsterdam_a2bep_5	0.2079	*0.42146*	0.37600	0.40800	0.35600	0.31400	0.22240	0.04448
KnowCenterGraz_disamTopic_IL_None_sentence	0.2076	0.22194	0.18000	0.19400	0.19700	0.21467	0.24480	0.04896
KnowCenterGraz_disamDoc_IL_None_sentence	0.1764	0.21270	0.11200	0.14000	0.16900	0.18800	0.24120	0.04824
CMIC_LTW_01	0.1760	0.18689	0.13200	0.18600	0.18700	0.17200	0.18280	0.03656
QUT_P9_GPXA2Btitle	0.1725	0.13739	0.21600	0.20600	0.18000	0.16467	0.12640	0.02528
CSIR_LTW_A2BEP_2	0.1307	0.10081	0.19600	0.17400	0.14000	0.11267	0.08480	0.01696
Amsterdam_a2bep_1	0.1271	0.25973	0.14000	0.14200	0.18600	0.18800	0.18520	0.03704
QUT_Anchor-BEP_1	0.1149	0.11075	0.11200	0.10400	0.11300	0.11333	0.11080	0.02216
Amsterdam_a2bep_2	0.1127	0.23963	0.12400	0.14800	0.19000	0.18400	0.16080	0.03216
Amsterdam_a2bep_3	0.0983	0.34507	0.13200	0.18800	0.23100	0.22000	0.16280	0.03256

5.2 Anchor to Best Entry Point

Surprisingly, the results for the anchor-to-best-entry-point evaluation do not differ much from the results for the anchor-to-file results (see figure 5). The main reason is presumably that most participants actually submitted anchor-to-file results to this task; furthermore, in many cases, the best entry point in a link target will in fact be at the very start of the document. Since the results are virtually identical, we will not discuss them here.

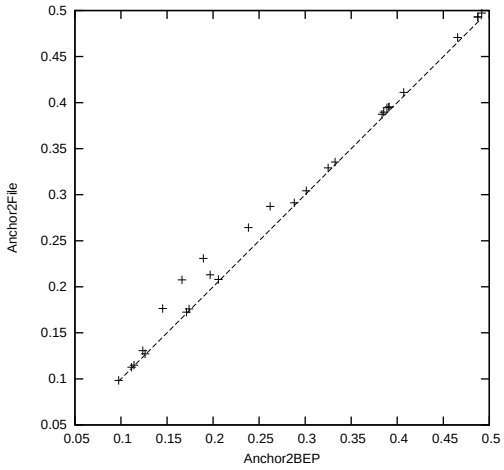

Fig. 5: Mean average precision for anchor-to-best-entry-point and anchor-to-file results. There is a strong correlation, only a few runs stand out.

5.3 File to File

Although our main focus was on the anchor-to-file runs, we also submitted runs to the file-to-file task. Our runs were created by simply omitting the anchor information, ordering by global score. Unsurprisingly, our runs did not perform well in this task, both for the outgoing and the incoming links.

6 Conclusions and Future Work

We have confirmed that the basic approach of Itakura and Clarke [2] works very well as a baseline for new methods. We have shown that this this method can be improved significantly by incorporating textual similarity to disambiguate anchor texts that could refer to several articles (the original method only used frequency statistics). Unfortunately, the run-time penalty for this can be rather high, since a similarity search on all articles is required for each file (but not for

every anchor!). State-of-the-art search engines are quite fast, so this should not be a major problem in all but the most time-critical settings.

Our submission should be regarded as a first attempt at the problem; in particular, we have not yet evaluated using network analysis for disambiguation. In preliminary experiments, we have found this to be quite successful, but we still need to perform more elaborate experiments on the INEX corpus. In future work, we plan to address this by taking more factors into account.

Acknowledgements

The research presented in this paper was partially funded by the German Federal Ministry of Economy and Technology (BMWi) under grant number 01MQ07008. The authors are solely responsible for the contents of this work. We thank our colleagues at Lycos Europe who gave valuable feedback.

References

1. Darren Wei Che Huang, Yue Xu, Andrew Trotman, and Shlomo Geva. Overview of INEX 2007 link the wiki track. In *Proc. INEX 2007*. Springer, 2008.
2. Kelly Y. Itakura and Charles L. A. Clarke. The University of Waterloo at INEX2007: Adhoc and Link-the-Wiki tracks. In *Proc. INEX 2007*. Springer, 2008.