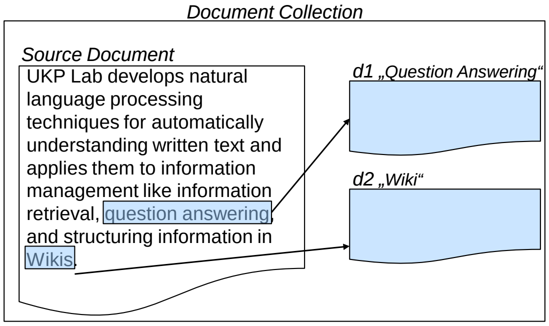
# Link Discovery: A Comprehensive Analysis

Nicolai Erbs, Torsten Zesch, Iryna Gurevych
Ubiquitous Knowledge Processing Lab
Technische Universität Darmstadt
Hochschulstrasse 10
D-64289 Darmstadt, Germany
http://www.ukp.tu-darmstadt.de

***Abstract*—We present a comprehensive analysis of approaches for discovering links in document collections. We classify link discovery approaches with respect to the type of knowledge being used: the text of a document, its title, and already existing links. Using an evaluation dataset derived from Wikipedia, we find that link-based approaches outperform all other approaches if they can draw knowledge from a very large amount of already existing links. Simulating other document collections with fewer links, we show that text-based approaches yield better results. Furthermore, we argue that knowledge from Wikipedia cannot necessarily be applied to other domains, e.g. in corporate intranets. Thus, we conclude that text-based approaches are the best choice for reliable link discovery in arbitrary document collections.**



UKP Lab develops natural language processing techniques for automatically understanding written text and applies them to information management like information retrieval, question answering and structuring information in Wikis.

Fig. 1. Linking from text in a source document to target documents.

## I. Introduction

The World Wide Web only works because the web pages are connected by links. Without links, it would be impossible to quickly navigate from one page to another. Additionally, search algorithms like HITS [1] or PageRank [2] utilize links to determine the relevance of pages.

Some regions of the Web already contain a large number of links. For example, links in Wikipedia are created by a large community of highly motivated contributors [3]. However, in other situations (e.g. in corporate intranets or wikis) it is more difficult to motivate people to contribute [4]. Users find it especially difficult to add links, as they need to decide what other pages constitute a valid link target. Even in smaller document collections like a corporate intranet this is a very difficult task, especially if the pages are subject to frequent changes. In such situation, link discovery approaches provide support for users trying to add new links to a certain document collection. Thereby, a link discovery algorithm usually first selects promising link anchors in a document, and then retrieves possible target documents for that anchor (cf. Figure 1). The user only has to decide whether the suggested link should be added to the document.

Current state-of-the-art link discovery approaches can be categorized according to the type of prior knowledge they utilize, e.g. already existing links, meaningful page titles, or the document text. Consider the link in Figure 1 that connects the anchor “question answering” with the target document $d1$. If this link was already present in the document collection, it provides knowledge that $d1$ is probably a good target whenever the anchor “question answering” is observed. If the title of this target page happens to be “question answering”, the link discovery algorithm can use this knowledge to predict the link even more reliably. Finally, the textual content of source and target document could be used to compute the document similarity which can also be used to suggest possible target documents.

In this paper, we argue that previous evaluations of link discovery approaches have always used document collections like Wikipedia in which a lot of prior knowledge in form of links and meaningful page titles is available. This obviously entails a bias towards link-based and title-based approaches which consequently outperformed text-based methods by a wide margin. However, in many realistic settings (e.g. in corporate intranets) one cannot rely on already existing links or page titles [5]. We show that the common evaluation setting which uses Wikipedia is more like a special case. In more realistic settings, links can be discovered with higher precision using text-based approaches.

We now take a deeper look at the process of *link discovery*, and the types of knowledge used for different approaches.

### A. Link Discovery

As shown in Figure 2 (horizontal grouping), the process of link discovery consists of two steps: anchor discovery and target discovery. *Anchor discovery* identifies which parts of a document should be used as link anchors, and *target discovery* identifies the best matching target for each anchor. Both steps consist of a selection process, in which possible candidates are found, and a ranking process, in which the list of candidates is sorted according to their relevance. For each step, various approaches can be applied that differ in

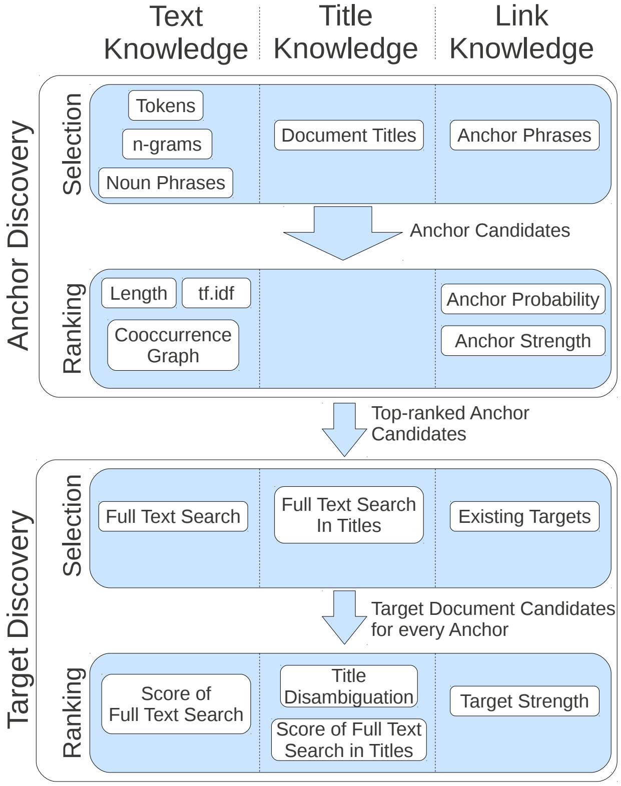


Fig. 2. Link discovery approaches split up into a step-by-step representation and classified by the type of knowledge used.

the type of collection-specific knowledge that is used (see vertical grouping in Figure 2). In the default case (which is also the most difficult one), approaches can only make use of raw text in a document (*text knowledge*) and no other prior knowledge. In some document collections (including intranets and wikis), each document has a title that can be used to facilitate the anchor selection process (*title knowledge*). If the collection contains links between documents, this is another important source of information (*link knowledge*). We now give a detailed overview about the approaches for anchor and target discovery, and categorize the approaches according to the type of prior knowledge used.

## II. Anchor Discovery

Anchor discovery extracts text spans that can be used as a link anchor by first selecting a set of candidates and then ranking them. The methods used for this purpose are very similar to *Keyphrase Extraction* [6], [7]. A notable difference is that keyphrases should describe the topics in the document, while anchor phrases must provide a good starting point for a link to another document. For example in a document about *"Baseball"*, good keyphrases are *"Pitcher"* and *"Home run"*, while *"famous players"* is not a good keyphrase. It is however a good anchor for linking to a document with a list of famous baseball players.

### A. Anchor Selection

In this step, a list of candidate anchors (i.e. phrases from the document) is selected. We categorize the approaches according to the amount of prior knowledge used.

*Text Knowledge:* A widely used approach to anchor selection is to select all possible candidate anchors that consist of a certain number of tokens (called *n-grams*) [8]. As this approach creates a lot of invalid anchors (e.g. *"is the yellow"* is a valid n-gram with very low probability of being a valid anchor), it heavily relies on the subsequent *anchor ranking* step to filter such cases. Linguistic preprocessing components like noun phrase chunkers [9] or named entity taggers [10] can be used to restrict the anchor candidates to a subset that is more likely to contain valid anchors. For example, if we restrict anchors to noun phrases, *"is the yellow"* would have not been selected as an anchor candidate, because it is not a valid noun phrase, while e.g. *"yellow submarine"* is valid and has a higher probability of being a useful anchor. A special case of noun phrases are named entities that correspond to predefined categories like persons (*"George Washington"*), locations (*"New York"*), or organizations (*"United Nations"*). Using only named entities further filters the list of selected anchors, as it also rejects common noun phrases like *"the beginning"*. A major disadvantage of linguistically motivated anchor selection approaches is that noun phrase chunkers or named entity taggers are not available for all languages and need to be trained for the specific document collection.

*Title Knowledge:* If the documents in a collection contain titles, they can be used to constrain the list of selected anchor candidates. This has two advantages: First, titles are usually well formed phrases. Thus, the list of anchor candidates can be pruned without the need to apply linguistic preprocessing tools that might not be available for all languages. Second, in the subsequent *target discovery* step, there will always be a document whose title exactly matches the anchor, and that is thus a very likely target document for this anchor. The downside is that titles are not available for all document collections, which limits the applicability of this approach. Also, it does not cover cases in which anchors are highly related to page titles, e.g. synonyms.

*Link Knowledge:* If a document collection already contains links, the corresponding link anchors constitute a good source of anchor candidates. A phrase that has already been selected by a human as an anchor in one document is probably still a good anchor in another document. This also solves the problem that good anchors are sometimes unusual phrases. However, it also means that, in order to reliably add links, the document collection already needs to contain links which turns the task into a 'chicken or the egg' dilemma.

### B. Anchor Ranking

The output of the anchor selection step is a (possibly noisy) list of candidates that needs to be ranked in order to select the best candidates. Taking the full list of anchor candidates might result in an over-linked article. However, the optimal number of links per document depends on the user preferences[1] and the domain[2]. Thus, we need to rank the full list of anchor candidates in order to return only a certain number of top-ranked anchors that are necessary in that context. Like the anchor selection approaches, we categorize anchor ranking approaches according to the amount of prior knowledge that is used:

*Text Knowledge:* There are three common text-based approaches for anchor ranking: the length of the anchor phrase, the tf.idf score [13] of the anchor phrase, and using a coccurrence graph [14].

**Length:** The length of an anchor candidate can be used as a baseline ranking method. Longer candidates correspond to longer page titles which are assumed to be more specific than others (and thus better anchors) [15]. For example, the longer candidate *"Queen of England"* is more specific than *"England"*, and should be ranked better. The usefulness of this approach strongly depends on the anchor selection strategy. Obviously, it will not work well, if anchor candidates are all of equal length (e.g. if only single tokens are selected), as all candidates will have the same rank.

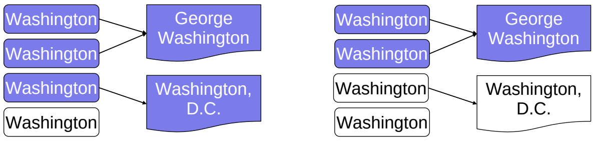
**tf.idf:** The underlying hypothesis of this approach is that more frequently appearing candidates are more likely to be good anchors. However, in order to avoid ranking common words to high, frequency should be combined with the inverse document frequency (idf), which is high if a candidate only appears in few documents. For example, if the terms *"the"* and *"United Nations"* both appear five times in a document, both would be ranked the same using only *tf*. However, *"the"* probably occurs in almost every document. Hence, its *idf* score will be very low, while *"United Nations"* only appears in a couple of documents resulting in a higher *idf* score. Overall, *"United Nations"* will be ranked much higher than *"the"* due to the higher combined *tf.idf* score.

**Cooccurrence Graph:** This method creates a graph representation of a document. In the graph, anchor candidates are used as the nodes, and an edge is added if two candidates cooccur in a certain context window (e.g. 3 words left or right of the anchor candidate, or in the same sentence as the anchor candidate) in the document. The weight of the edge is defined as the number of cooccurrences. A graph centrality measure like PageRank [2] is then used to rank the anchor nodes.

*Title Knowledge:* So far, no special methods relying on page title knowledge have been proposed for *anchor ranking*.

*Link Knowledge:* The methods in this section make use of the links that are already present in the document collection. We define a link as an anchor pointing from a source document to a target document. Formally, we define $p$ as a phrase (text span) and $D$ the set of all documents in the collection. We then define $l(p, d)$ as the number of links where $p$ is an anchor phrase in a source document and $d \in D$ is a target document

**Anchor probability** estimates the probability of a phrase p in a document to be selected as an anchor based on how often it has been previously used as an anchor for any target ($\Sigma_d l(p, d)$) divided by the total number of documents in which the phrase appeared ($|D_p|$). $D_p$ is defined as the subset of all documents containing the phrase $p$.



(a) Anchor probability (b) Anchor strength

Fig. 3. Example of anchor phrases partially used as link anchors for different target documents

$$anchor\ probability(p) = \frac{\Sigma_d\, l(p, d)}{|D_p|} \qquad (1)$$

Considering the example shown in Figure 3(a), the phrase *"Washington"* is used three times as an anchor and occurs four times in the text. Thus, its anchor probability is $\frac{3}{4}$. Anchor probability is also called *Keyphraseness* in the domain of keyphrase extraction [12]. Milne and Witte [16] use it as an important feature for their machine learning approach to *anchor ranking*.

**Anchor strength [17]** A disadvantage of *anchor probability* is that it does not consider whether an anchor is ambiguous. For example, if a phrase is used as a link anchor in each document, its *anchor probability* will be 1.0 even if it points to a different target every time. A common example is the anchor *"here"* in sentences like *"The documents can be found here."* where the link might point to almost any target. Thus, *anchor strength* also incorporates the ambiguity of an anchor by only counting the number of times an anchor points to its most frequent target.

$$anchor\ strength(p) = max_d\, \frac{l(p, d)}{|D_p|} \qquad (2)$$

For example in Figure 3(b), the anchor *"Washington"* points twice to the target document $d_1$ *"George Washington"* and only once to the target document $d_2$ *"Washington, D.C."*. As the anchor is still used four times in the collection, the resulting *anchor strength* is $\frac{2}{4}$ or $\frac{1}{2}$.

## III. Target Discovery

*Target discovery* identifies the best matching document for an anchor. Similar to *anchor discovery*, the task consists of a selection step and a ranking step. The step is similar to Information Retrieval, where relevant documents are retrieved given a query (in this case the anchor phrase).

### A. Target Selection

*Text Knowledge:* Following Information Retrieval approaches, the target documents are indexed using common search engine libraries like Terrier or Lucene [18]. The resulting index is then queried using the anchor candidate. However, with this approach only exact matches of the anchor candidate in the target documents can be found. More relevant

[1] For keyphrases, which are very similar to link anchors, it has been shown that that the favored density of keywords depends on the user [11].

[2] In Wikipedia, on average 6% of all words are anchors [12]

target documents could be retrieved using semantic search [19], where highly related terms like synonyms are taken into account. Another approach is to use query expansion, e.g. by using more context (like e.g. the first sentence from the source document) [20].

*Title Knowledge:* Instead of searching the full document text, we can constrain the search space to document titles only. Especially for huge collections this results in a faster response time which is of great importance in an online system for link discovery. However, only searching in the space of document titles also means that there needs to be an overlap between the anchor phrase and the document title.

*Link Knowledge:* If the document collection already contains links, we can check if there are already links with a given anchor phrase. We can then limit the list of target candidates to those that have already been linked to this anchor. For example, if the anchor phrase *"this approach"* occurs in the collection pointing to the targets *"Dijkstra Algorithm"* and *"Breadth-First-Search"*, only these two targets will be considered as target candidates.

### B. Target Ranking

*Text and Title Knowledge:* If a search engine is used for target selection, the resulting list is usually the full document collection with assigned relevance scores. Depending on whether the search engine works on the document text or only the document titles, the method uses *text knowledge* or *title knowledge*. If only document titles have been used to select possible target documents, finding the correct target boils down to Word Sense Disambiguation [21]. For example, the anchor *"Bank"* may match the document titles *"Bank (river)"* and *"Bank (money)"*[3]. We now have to determine which article is meant by measuring the similarity between the textual content in which the anchor phrase is used and the textual content of the documents.

*Link Knowledge:* One way to incorporate link knowledge into target ranking is to count how often the selected anchor phrase points to that target (*target strength*). Formally, we define:

$$target\,strength(p, d_t) = \frac{l(p, d_t)}{\Sigma_d\, l(p, d)} \tag{3}$$

Consider the anchor phrase "Washington" in Figure 3 with the possible target documents $d_1$ *"George Washington"* and $d_2$ *"Washington, D.C."*: The target strength for *"George Washington"* is $\frac{2}{3}$ and for *"Washington, D.C."* it is $\frac{1}{3}$.

## IV. Classification of Existing Approaches

In this section, we give an overview of state-of-the-art link discovery approaches and classify them according to which knowledge they utilize.

### A. Gevas Page Name Matching (GPNM)

GPNM [15] combines methods which use text, title, and link knowledge, as shown in Figure 4(a). Every title from the document collection that can be found in the source document is considered as an anchor candidate that is then ranked according to their length. Targets are selected and ranked using link knowledge, i.e. possible targets are limited to those that have already been linked using this anchor. The score is set according to how often they have been linked using this anchor.

### B. Itakura & Clark Link Mining (ICLM)

As shown in Figure 4(b), this approach [17] completely relies on knowledge derived from existing links. In the source document, all phrases that have at least once been used in the document collection as a link anchor are considered as anchor candidates. The candidates are ranked using *anchor strength*. Target candidates are all documents that have been previously linked using this anchor. The more often a target has been linked by the anchor, the better its rank in the list of targets.

### C. Text-based approach

We propose an approach that only uses text knowledge, but no prior knowledge about the document collection. Hence, we do not make use of any document titles or existing links. Figure 4(c) gives an overview of the system configurations used in this paper. We experiment with three anchor selection methods (tokens, n-grams, and noun phrases) and three anchor ranking methods (anchor length, tf.idf weight, and cooccurrence graphs). For target selection, we perform a full text search in the document collection with the anchor text as the query. We use the target relevance scores returned by Terrier[4] for target ranking.

## V. Evaluation

Automated evaluation of link discovery approaches requires a document collection that already contains links. Previously, Wikipedia has been used that purpose, e.g. for evaluation in the context of the INEX 2009 Link-the-Wiki-Track [22]. We follow this approach, as Wikipedia is publicly available and contains high quality links that were collaboratively added and verified by a large number of Wikipedia contributors.[5]

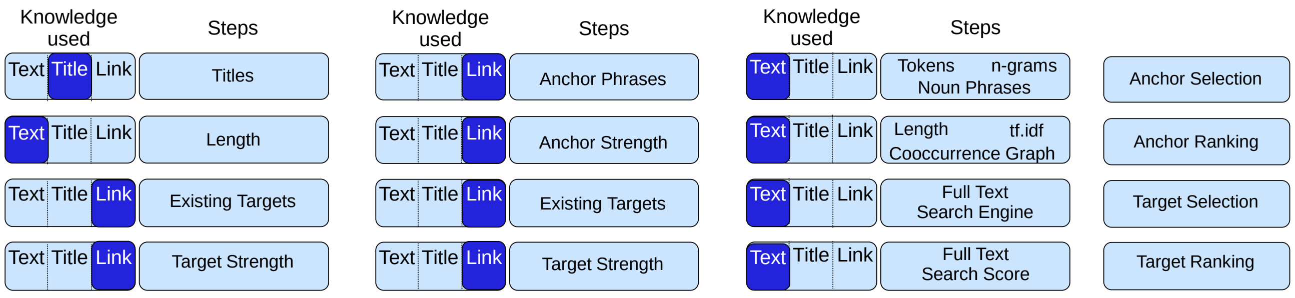
We use a Wikipedia snapshot from October 8, 2008 containing 2,666,190 articles and 135,478,255 links (this is the same version that was used for the evaluation at INEX 2009). The dataset used for testing consists of 2,709 articles (every 1,000th article) containing 140,143 gold standard links. The remaining articles are used to provide link and title knowledge.

We evaluate the performance of *anchor discovery* and *target discovery* separately, as in the intended use case (supporting users when adding links to unstructured document collections) the two steps are also separated: First, anchors are highlighted

[3] Such situations often occur in Wikipedia, as there is a high probability that more than one article exists for a certain term. For example, there are 8 different articles for "Einstein" in the English Wikipedia (as of September 8th, 2010).

[4] http://www.terrier.org/

[5] As the Wikipedia community agrees to follow certain guidelines when adding links, we can consider Wikipedia as a corpus that is annotated by the community with high quality.



(a) GPNM system

(b) ICLM system

(c) Text-based system

Fig. 4. Overview of link discovery approaches and the type of knowledge used.

in a document, and valid anchors are selected by the user. Second, for this set of user verified anchors, the best targets are computed and presented to the user who then decides which target to link to.

## A. Anchor Discovery Results

In order to allow a fair comparison of the different anchor discovery approaches, we evaluate all approaches using the same preprocessing and postprocessing steps. We limit the number of suggested anchor candidates per document, as some anchor selection methods like *n-gram* create too many candidates. We use a threshold relative to the document length instead of a fixed threshold, as the document length varies considerably in the test collection. For example, adding only 10 links to a long document might not be sufficient, while adding 10 links to a short document might already be too much. Following [12], we use a threshold of 6% (i.e. approximately 1 out of 17 tokens is used as a link anchor) as an upper bound. However, in our Wikipedia evaluation dataset, we found that the average number of tokens used as links is only 1.7%. Thus, we decided to also use a threshold of 1% (i.e. 1 out of 100 tokens is used as a link anchor) as a lower bound.

We use two baselines: (i) selecting all tokens as anchor candidates, and (ii) selecting all noun phrases. In both cases, we rank the candidates according to their position in the document (the earlier a candidate appears, the better it is ranked). We compare the baselines with the state-of-the-art approaches GPNM (title-based) and ICLM (link-based), as well as a wide range of text-based configurations as explained in the previous section. Table I displays the obtained results in terms of precision[6], recall[7], and f-measure[8] at the two linking threshold levels 1% and 6%.

The overall precision of all approaches is rather low. However, it is known that using Wikipedia for evaluation underestimates the actual precision of unsupervised approaches [23], as Wikipedia contains many anchors like dates or numbers which are not easily captured by text-based approaches. Also, we do not require perfect precision, as the user will select valid link anchors from the highlighted set of anchor candidates. Thus, we consider this level of precision to be sufficient for providing link support, which was also confirmed by our practical experiments and user studies.

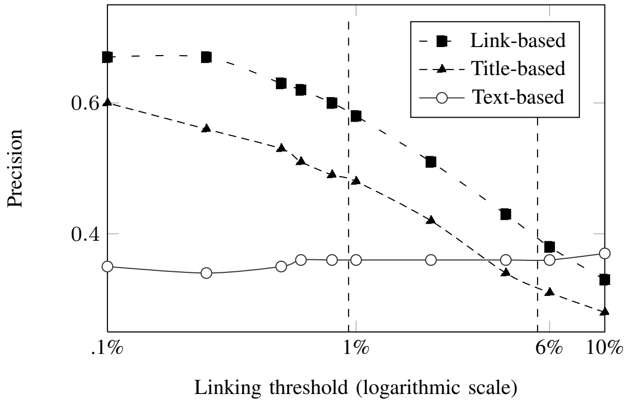


Fig. 5. Anchor discovery precision depending on linking threshold (i.e. ratio of document's text that is linked). The threshold 1% and 6% are indicated by vertical lines.

The best performing text-based approach is a combination of token candidates using cooccurrence graph ranking. Cooccurrence graph based ranking generally outperforms the other ranking approaches *length* and *tf.idf*.

For a linking threshold level of 1% (few links per document) the title-based (ICML) and link-based approach (GPNM) perform much better than any text-based approach. However, for a linking threshold of 6% (many links per document) the text-based approach outperforms the title-based approach, and the distance to the link-based approach is much smaller. Given these large differences between the results at the two threshold levels, we systematically analyzed the influence of the linking threshold on the precision of the different approaches. Figure 5 shows the precision of the link-based, title-based, and the best text-based approach for different linking thresholds. It shows that link-based and title-based approaches perform well when discovering only few links (the smallest threshold is .1%, i.e. 1 in 1,000 tokens are used as anchor phrases). As the unsupervised text-based approach is independent of the linking threshold, it performs slightly better than the other approaches when discovering many links (10% or 1 in 10 tokens).

[6] # correct anchors retrieved / # total anchors retrieved

[7] # correct anchors retrieved / # anchors in gold set

[8] $F = \frac{2PR}{P+R}$

TABLE I
RESULTS OF ANCHOR DISCOVERY APPROACHES ON THE WIKIPEDIA TEST COLLECTION.

|  | Anchor | Anchor | Info | 1% |  |  | 6% |  |  |
|---|---|---|---|---|---|---|---|---|---|
| Name | selection | ranking | Type | P | R | F | P | R | F |
| Baselines | Tokens | Position | Text | .11 | .03 | .05 | .11 | .09 | .10 |
|  | Noun phrases | Position |  | .23 | .06 | .09 | .22 | .12 | .16 |
| Text-based | Tokens | Coocc graph | Text | .36 | .10 | .16 | .36 | .25 | .29 |
|  | N-grams |  |  | .28 | .08 | .12 | .30 | .23 | .26 |
|  | Noun phrases |  |  | .27 | .07 | .11 | .26 | .14 | .18 |
|  | Tokens | tf.idf | Text | .19 | .05 | .08 | .16 | .13 | .14 |
|  | N-grams |  |  | .16 | .04 | .07 | .16 | .13 | .15 |
|  | Noun phrases |  |  | .25 | .06 | .10 | .23 | .13 | .17 |
|  | Tokens | Length | Text | .12 | .03 | .05 | .11 | .09 | .10 |
|  | N-grams |  |  | .13 | .03 | .05 | .13 | .10 | .11 |
|  | Noun phrases |  |  | .23 | .06 | .09 | .22 | .12 | .16 |
| GPNM | Page titles | Length | Title | .48 | .13 | .21 | .31 | .26 | .28 |
| ICLM | Link anchors | Anchor strength | Link | .58 | .16 | .26 | .38 | .31 | .34 |

**Available training data** As we have seen in Table I, the link-based ICML approach performs best. However, it heavily depends on the number of existing links in the collection. The large amount of links in Wikipedia is clearly a special case that is due to the highly motivated voluntary editors. We simulate the case of a less linked document collection by reducing the available training data in Wikipedia, thus controlling the amount of link knowledge that is used by the ICML system. We randomly remove links until only .001% of the original links (∼ 1,000 links) are left in the training data.[9]

Figure 6 shows how precision changes with decreasing number of links available for training. In contrast to link-based anchor discovery, text-based and title-based approaches are not influenced by the amount of available link data. Thus, they appear as horizontal lines. The link-based approach performs best when all the training data (over 130 million links) is available, but quickly drops below the text-based approach (at around 50% of training data) and finally also below the title-based approach (at 1% of training data). As the performance of link-based approaches deteriorates that quickly, they cannot be used to reliably predict link anchors in most other document collections, where less training data is available.

**Domain transfer** As we have seen above, the link-based approach only works well if a large number of links is available for training, which is hardly the case for most document collections. An obvious idea would be to use the knowledge about links and document titles from Wikipedia to improve anchor detection in other document collections. However, this turns into an issue of domain transfer, and will not work in many cases.

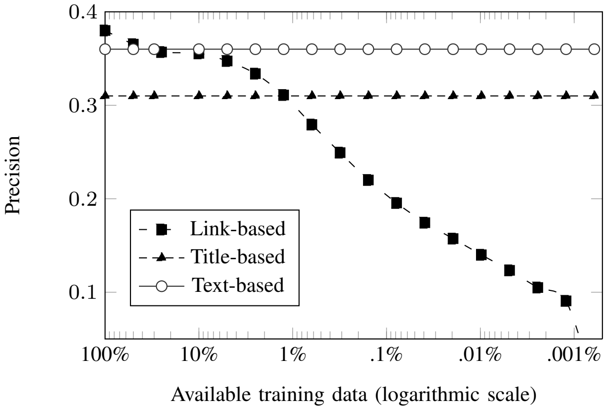


Fig. 6. Precision of link based anchor discovery depending on the available training data at 6% threshold

For example, title-based anchor discovery uses the list of all articles in a collection as candidate anchors. Applying the list of all Wikipedia articles to another collection may not capture domain-specific anchors. For example, Wikipedia does not contain an article for each university professor. Thus, in a document collection about one specific university, the knowledge from Wikipedia will not be useful to select an anchor candidate for a link to the professors' personal homepages.

Likewise, using link-based anchor discovery it is not possible to capture domain-specific anchor phrases that do not occur as a link in the training data. For example, if link information from Wikipedia is used to discover anchors in a corporate environment, names of a company's products can probably not be discovered, as product names in Wikipedia are usually not considered worth linking, except for very well known products.

## B. *Target Discovery Results*

Target discovery cannot be evaluated fully independently of anchor discovery, as we need a list of source anchors for which to discover the correct targets. Thus, we select the best performing anchor discovery configuration for each approach from the 6% threshold case, and perform target discovery using the anchors candidates output by using the same type of knowledge[10]. Using the best set of anchors for each approach allows for a fair comparison.

[9] Randomly removing links is only a rough approximation of the real growth (shrinkage) process of Wikipedia that can be modeled by preferential attachment [24].

[10] We also tested anchor candidate sets produced by other approaches, but it did only marginally influence results.

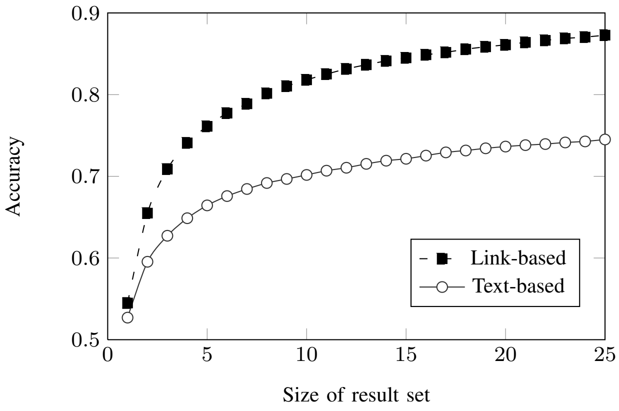


Fig. 7. Accuracy of target discovery depending on the size of the result set. An approach correctly identified a target document if it is contained in the result set.

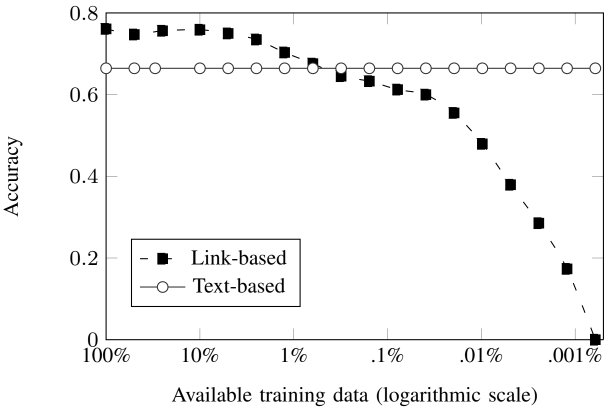


Fig. 8. Accuracy of target discovery depending on the available training data. (Result set size = 5)

As taking into account target discovery for wrong anchor candidates would yield misleading results, we only consider correct anchors. We also filter anchors that are dates or numbers. This way, we mimic our setting of first selecting an anchor, and then choosing the best matching target, where a user would only select valid anchors from the full list of suggested anchors.

As evaluation metric for target discovery, we use a relaxed version of accuracy: We compute a result set with target suggestions which is defined to be correct if it contains the gold target. This relaxed definition mimics the user's view on the result set, as we expect the user to identify the correct target given a list of suggestions. Obviously, this is limited to a result set up to a certain length. Hence, we limit the result set to 10 target suggestions, as this is the number of suggestions returned by common search engines. The overall accuracy is then calculated as the number target sets containing the correct target to the total number of gold targets.

Figure 7 shows the accuracy of the text-based and link-based approaches depending on the size of the result set. Note, that the two state-of-the-art approaches GPNM and ICLM are both treated as link-based approaches, as they use the same steps for target discovery (compare Figure 4(a) and 4(b)). As we can see from the Figure 7, the link-based approach outperforms the text-based approach for larger result sets, but they perform comparably for very small result sets. If we only consider the single top-ranked target document, the accuracy is rather low (around 50%), i.e. in only 50% of all cases can the correct target document be found on the first rank. However, if we return 10 target suggestions (which we expected to be a good size of the result set), the performance of the text-based approach improves to 70%, and that of the link-based approach to 80%. Further experiments show that accuracy rises nearly linear for more than 10 targets, but stays below 90% even for a result set size of 1,000. However, users are not expected to view such a large number of results.

**Available training data** We further analyzed the influence of the available amount of link knowledge by randomly reducing the number of links used for training. Figure 8 shows the accuracy of target discovery approaches depending on the amount of available training data. The results are very similar to those for anchor discovery. The link-based approach outperforms the text-based approach when all the training data (over 130 million links) is available, but drops below the text-based approach if the amount of training data is reduced. As a consequence, a large number of links is required to yield acceptable performance using the link-based approach. This means that adding a few links does not help, which makes the approach vulnerable to the "slow-start" problem. As the text-based approach is not affected by a low number of links, it can already provide link suggestion support even for document collections without existing links.

**Domain transfer** Knowledge about already existing links in Wikipedia is very useful for creating new links in Wikipedia. However, it will not help much for other document collections, as can be shown using a simple example. Inside Wikipedia, it is valuable knowledge to know that the anchor phrase *"Java 5"* almost always points to the article about the programming language *"Java"*. However, this does not help us to decide in other document collections, where there might be no such document or in collections where there are more specific documents. In a document collection about programming languages there probably exists one page for every version of Java. This cannot be captured by using the knowledge derived from Wikipedia.

### C. Evaluation summary

We have shown that link-based link discovery performs best for a document collection that contains a large amount of training data. However, the large amount of links in Wikipedia is clearly a special case (due to the highly motivated voluntary editors) which we cannot expect to be available for other document collections. Thus, we further analyzed the influence of the available amount of link training data and found that the performance of link-based approaches quickly drops if less training data is available.

Another problem with link-based approaches trained on Wikipedia is that they cannot be easily transferred to other document collections. Knowledge about good anchors in one domain is highly specific and cannot be applied to domains with another focus. Knowledge about link targets is only useful for target discovery if the same target document exists in both collections, which is rather unlikely for most domains.

For arbitrary document collections, only text-based approaches can be used for reliable link discovery.

## VI. Summary

In this paper, we evaluated the performance of link discovery approaches and presented a classification scheme for link discovery approaches with respect to the type of knowledge being used. We evaluated them on a test collection derived from Wikipedia, and showed that the link-based approach outperforms all other approaches if it can draw knowledge from a huge number of already existing links. However, other document collections normally contain much less links, and thus provide less knowledge about good link anchors and targets. As a consequence, link-based approaches suffer from the "slow start" problem, i.e. in a collection with only a few links they do not provide helpful link suggestions. Their performance only gets acceptable when a large number of links is manually added to the collection. In contrast, the text-based and title-based approaches are able to provide linking support, even if no links have been added so far.

Furthermore, we argued that knowledge from Wikipedia which is needed for title-based and link-based approaches is not necessarily transferable to other domains. Thus, text-based approaches are the best choice for reliable link discovery in arbitrary document collections.

## Acknowledgements

This work has been supported by the Volkswagen Foundation as part of the Lichtenberg-Professorship Program under grant No. I/82806. The project has been supported by the Klaus Tschira Foundation under project No. 00.133.2008. We thank Petra Stegmann for her helpful feedback.

## References

[1] J. Kleinberg, "Authoritative sources in a hyperlinked environment," in *Proceedings of the 9th ACM-SIAM Symposium on Discrete Algorithms*, 1998.

[2] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank Citation Ranking: Bringing Order to the Web," Stanford InfoLab, Technical Report 1999-66, November 1999. [Online]. Available: http://ilpubs.stanford.edu:8090/422/

[3] K. Nakayama, T. Hara, and S. Nishio, "Wikipedia Mining for an Association Web Thesaurus Construction," in *Web Information Systems Engineering WISE 2007*, ser. Lecture Notes in Computer Science, 2007, vol. 4831, pp. 322–334.

[4] A. Majchrzak, C. Wagner, and D. Yates, "Corporate Wiki Users: Results of a Survey," in *WikiSym '06: Proceedings of the 2006 international symposium on Wikis*, Odense, Denmark, 2006, pp. 99–104. [Online]. Available: http://doi.acm.org/10.1145/1149453.1149472

[5] M. Buffa, "Intranet wikis," in *Proceedings of the IntraWebs Workshop 2006 at the 15th International World Wide Web Conference*, 2006.

[6] E. Frank, G. W. Paynter, I. Witten, C. Gutwin, and C. G. Nevill-Manning, "Domain-Specific Keyphrase Extraction." in *Proceedings of the 16th International Joint Conference on Aritificial Intelligence (IJ-CAI)*, San Mateo, USA, 1999, pp. 668–673.

[7] P. D. Turney, "Learning Algorithms for Keyphrase Extraction," *Information Retrieval*, vol. 2, no. 4, pp. 303–336, 2000.

[8] C. D. Manning and H. Schütze, *Foundations of statistical natural language processing*. The MIT Press, 1999.

[9] H. Schmid, "Probabilistic Part-of-Speech Tagging Using Decision Trees," in *International Conference on New Methods in Language Processing*, Manchester, UK, 1995.

[10] J. R. Finkel, T. Grenager, and C. Manning, "Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling," in *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, 2005, pp. 363–370. [Online]. Available: http://www.aclweb.org/anthology/P05-1045

[11] S. Tucker and S. Whittaker, "Have A Say Over What You See: Evaluating Interactive Compression Techniques," in *Proceedings of the 2009 International Conference on Intelligent User Interfaces*, Sanibel Island, USA, 2009, pp. 37–46.

[12] R. Mihalcea and A. Csomai, "Wikify!: Linking Documents to Encyclopedic Knowledge," in *Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management, CIKM*, Lisbon, Portugal, 2007, pp. 233–242. [Online]. Available: http://doi.acm.org/10.1145/1321440.1321475

[13] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*. McGraw Hill, 1983.

[14] R. Mihalcea and P. Tarau, "TextRank: Bringing Order into Text," in *Proceedings of EMNLP-04and the 2004 Conference on Empirical Methods in Natural Language Processing*, 2004, pp. 404–411. [Online]. Available: http://www.aclweb.org/anthology/W04-3252

[15] S. Geva, "GPX: Ad-Hoc Queries and Automated Link Discovery in the Wikipedia," in *Preproceedings of the INEX Workshop*, 2007, pp. 404–416.

[16] D. Milne and I. H. Witten, "Learning to Link with Wikipedia," in *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management*, New York, USA, 2008, pp. 509–518.

[17] K. Y. Itakura and C. L. A. Clarke, "University of Waterloo at INEX2007: Adhoc and Link-the-Wiki Tracks," in *INEX 2007 Workshop Preproceedings*, vol. 4862, 2007, pp. 417–425. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-85902-4_35

[18] J. Hoffart, D. Bär, T. Zesch, and I. Gurevych, "Discovering Links Using Semantic Relatedness," in *Preproceedings of the INEX Workshop*, 2009, pp. 314–325.

[19] I. Gurevych, C. Müller, and T. Zesch, "What to be? - Electronic Career Guidance Based on Semantic Relatedness," in *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, Prague, Czech Republic, 2007, pp. 1032–1039. [Online]. Available: http://www.aclweb.org/anthology/P/P07/P07-1130

[20] O. Sunercan and A. Birturk, "Wikipedia Missing Link Discovery: A Comparative Study," in *AAAI Spring Symposium on Linked Data Meets Artificial Intelligence (Linked AI 2010)*, ser. AAAI Spring Symposium, A. S. Symposium, Ed., Stanford, USA, 2010. [Online]. Available: http://www.aaai.org/ocs/index.php/SSS/SSS10/paper/view/1072

[21] R. Navigli, "Word Sense Disambiguation: A Survey," *ACM Computing Surveys*, vol. 41, no. 2, pp. 1–69, 2009.

[22] D. W. C. Huang, S. Geva, and A. Trotman, "Overview of the INEX 2009 Link the Wiki Track," in *INEX*, ser. Lecture Notes in Computer Science, vol. 6203, 2009, pp. 312–323. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-14556-8

[23] W. C. Huang, A. Trotman, and S. Geva, "The Importance of Manual Assessment in Link Discovery," in *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, New York, USA, 2009, pp. 698–699. [Online]. Available: http://doi.acm.org/10.1145/1571941.1572084

[24] A. Capocci, V. D. P. Servedio, F. Colaiori, L. S. Buriol, D. Donato, S. Leonardi, and G. Caldarelli, "Preferential Attachment in the Growth of Social Networks: The Internet Encyclopedia Wikipedia," *Physical Review E*, vol. 74, p. 036116, 2006.