Discovering Links Using Semantic Relatedness

Johannes Hoffart, Daniel Bär, Torsten Zesch, and Iryna Gurevych

Ubiquitous Knowledge Processing Lab
Computer Science Department, Technische Universität Darmstadt
Hochschulstraße 10, D-64289 Darmstadt, Germany
www.ukp.tu-darmstadt.de

Abstract. We present our approaches for link discovery in document collections with or without existing links. In collections containing links, we discover links using measures of link anchor ranking based on existing links. In collections without links, we gather noun phrases as anchor candidates. To discover targets, we use a measure of semantic relatedness between texts. We find that semantic relatedness is useful to identify targets for ambiguous link anchors. In collections that contain no existing links, using only document titles as anchor candidates can be enhanced by using arbitrary noun phrases extracted from documents.

1 Introduction

Links are a crucial feature of hypertext to navigate document collections, but creating links is a daunting task. It requires a huge effort to decide which phrases are important enough for the reader to serve as link anchor, and which documents are good targets for that phrase. Additionally, knowledge of the best target implies knowledge of the complete document collection, something which is hard to achieve for a human. Thus, automatically discovering links is an important research topic.

We distinguish two types of document collections: those already containing links and those without any links. The information of document collections that already contain links, e. g. which phrases are often used as links or which documents are linked to by which phrase, can be used for discovering new links. One such document collection that contains collaboratively created links is Wikipedia. It has been the subject of a lot of link discovery research [1, 4, 7, 9, 11, 14]. In document collections without links, link discovery can make use only of the textual content of the documents, e. g. using methods of information retrieval [2, 8].

In this paper, we aim at creating link discovery algorithms that work both on document collections that already contain links, as well as on document collections that do not. To coordinate the research efforts in link discovery, there is the Link-the-Wiki track at INEX[1], in which we participated. In this work, we describe our contributions to the Link-the-Wiki track at INEX 2009, and qualitatively analyze the results.

[1] http://www.inex.otago.ac.nz

In the next section, we formally define the task of link discovery and describe related work. In Section 3, we give a brief overview of the tasks in the Link-the-Wiki track, and describe the two document collections used in the track. In Sections 4 and 5, we detail our link discovery approaches and qualitatively analyze the results.

2 Link Discovery in Document Collections

We distinguish between two types of links in document collections, as shown in Figure 1.

Document-level links relate a source document to a target document.

Anchor-level links relate a specific *anchor phrase* in the source document to a target document. Within the target document, a concrete *entry point* may be specified, e. g. section headings or paragraphs. They can be represented as character offset in the document.

Formally defined, let $\mathcal{D}$ be the document collection. The goal of link discovery is to connect a source document $s \in \mathcal{D}$ to a target document $t \in \mathcal{D}$ by means of hyperlinks. Such links are denoted by $l(s,t)$ and are called document-level links. From the perspective of a single document $d \in \mathcal{D}$, outgoing links are links that have d as source, $l(d,t)$, and incoming links have d as target, $l(s,d)$.

Additionally, we classify links in a more fine-grained manner: Links that originate from a specific anchor phrase p in s and link to a target document t are denoted by $l(s_p,t)$. Links from a document s to a certain entry point e in t are denoted by $l(s,t_e)$. Link from p in s to e in t, the most specific links, are denoted by $l(s_p,t_e)$. We call both $l(s_p,t)$ and $l(s_p,t_e)$ anchor-level links.

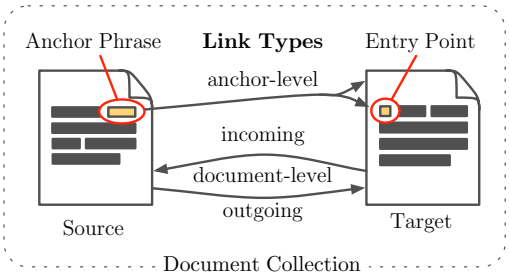

Fig. 1. Link Types in Document Collections

Finally, we define $\mathcal{D}_p$ as the set of documents containing a phrase p and $\mathcal{D}_{l(s_p,t_e)}$ as the set of documents containing the link $l(s_p,t_e)$ where both p and e can be omitted. Documents that do not contain any links at all are called *orphans*.

2.1 Anchor-Level Link Discovery

Discovering anchor-level links comprises two tasks: anchor discovery and target discovery. Figure 2 shows a classification of the approaches to the first task, Figure 3 of the approaches to the second one. Anchor phrases that are relevant to

the reader of the document, and thus should be connected to further information, need to be identified. In the target discovery step, the best matching target is retrieved. However, if there is no valid target in the collection, the link might be rejected.

Anchor Discovery Anchor discovery is done in two steps: identifying anchor candidates, followed by ranking the candidates. Potential **anchor candidates** are:

- **N-Grams**: term groups of length N, used e. g. by Geva [7].
- **Noun phrases (NPs)**: groups of determiners, prepositions, adjectives and nouns, e. g "the president of a country" or "natural language processing".
- **Document or section titles** extracted from the document structure.

Document collections containing links provide an additional source of anchor candidates, namely the phrases that have already been used as a link anchor in some document.

A measure to **rank anchors** that does not rely on an existing link structure but on the distribution of terms is tf.idf, which is used by Csomai and Mihalcea [4]. It can be used to rate both single and multi-term link anchors. These multi-term candidates are assigned the same rating as the single term with the maximum tf.idf value inside the multi-term one.

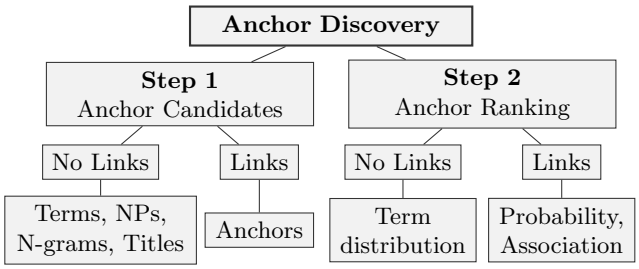

Fig. 2. Discovering Anchors

Ranking anchors can be improved using information about existing links. Thus, most of the approaches for ranking anchor-level links make heavy use of existing links in the collection. Csomai and Mihalcea [4] propose such a measure to rank the anchor candidates. It is called *keyphraseness*, motivated by the notion that using a phrase as a link anchor is a hint that it is a keyphrase in the document. It rates a phrase p according to the probability of p being used as anchor in a collection. The *keyphraseness* of p is the number of times it was used as link anchor in an article, divided by the total number of documents the phrase appears in. It is calculated as follows:

$$keyphraseness(p) = P(anchor|p) \approx \frac{|\mathcal{D}_{l(s_p,t)}|}{|\mathcal{D}_p|} \tag{1}$$

Using Equation 1, Csomai and Mihalcea achieve an f-measure of 0.55 on discovering anchors in Wikipedia.

Itakura and Clarke [9] introduced a measure to rank anchor candidates based on existing links at INEX 2007. First we define $\mathcal{T} := \{l(s_p, t)|t \in \mathcal{T}\}$ as the set

of link targets of a phrase p in the existing document collection, $\mathcal{T} \subseteq \mathcal{D}$. Let $z := \arg\max_{t \in \mathcal{T}} |\mathcal{D}_{l(s_p,t)}|$ be the most frequent target of a link. The *Itakura and Clarke link measure* (*iclm*) is then the number of times a phrase p occurs in a link $l(s_p, z)$, divided by the total number of documents p appears in.

$$iclm(p) = \frac{|\mathcal{D}_{l(s_p,z)}|}{|\mathcal{D}_p|} \tag{2}$$

We introduce a link measure that rates a phrase p according to the strength a link $l(s_p, t)$ anchored on p with target t is associated with its most frequent target. We will first define the *association strength* $as(p, z)$ between p and a specific target $z \in \mathcal{T}$:

$$as(p, z) = \frac{|\mathcal{D}_{l(s_p,z)}|}{\sum_{t \in \mathcal{T}} |\mathcal{D}_{l(s_p,t)}|}$$

This measures the strength of association of phrase p to target s. The measure we introduce takes the *maximum association strength* $as_{\max}(p)$ as the rating of p:

$$as_{\max}(p) = \max\{as(p, t) | t \in \mathcal{T}\} \tag{3}$$

This measure favors phrases with one highly probable link target. Phrases that have multiple equally common targets will get a lower $as_{\max}$-rating.

Target Discovery In document collections without links, targets can be discovered using any information retrieval model that searches for the anchor phrase. Targets can either be whole documents or an entry point in documents, depending on the granularity of the search.

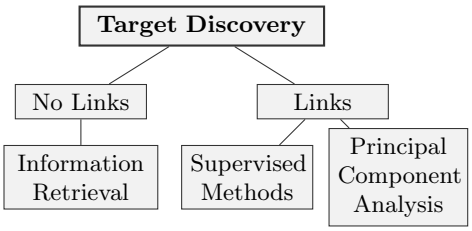

Fig. 3. Discovering Targets

In a document collection containing links, the targets of existing links can be used. However, some phrases are ambiguous, i. e. have different meanings, e. g. "bank" can mean "edge of a river" or "financial institution". In Wikipedia, such phrases have different link targets. Csomai and Mihalcea use this information to train a machine learning classifier for disambiguating targets, achieving an f-measure of 0.87 on Wikipedia.

Milne and Witten [11] propose an approach that **intertwines anchor and target discovery** for Wikipedia links. They train a machine learning classifier for both anchor identification and target disambiguation. One important feature of their target disambiguation classifier is the semantic similarity of two Wikipedia articles, which is measured by comparing article links [10]. The confidence of this disambiguation classifier is used as one of the features for identifying anchors. Among other features for training their anchor classifier are the keyphraseness value (see Equation 1) and the generality of the anchor phrase. Their approach results in an overall f-measure of 0.74 for recreating Wikipedia anchor-level links.

2.2 Document-Level Link Discovery

Unlinked Document Collections One method to discover links on the document level is to generalize anchor-level links. There are also methods that work directly on the document level. They can roughly be classified as shown in Figure 3. Allan [2] uses a standard vector-space information retrieval model to discover targets, using the document text as query. Green [8] tries to improve link discovery by using a semantic relatedness measure based on WordNet [5] and lexical chains. In his evaluation, the measure is compared to links created by vector-space methods like the above mentioned method by Allan, but without significant improvements. Chen et al. [3] propose to link documents that have a high number of overlapping frequent phrases, which achieved the best result in discovering incoming links at INEX 2008.

A measure to calculate the semantic relatedness between texts was introduced by Gabrilovich and Markovitch [6], called *Explicit Semantic Analysis* (ESA). It can be used to identify similar documents as targets. ESA uses the text of all Wikipedia articles to construct a document-level vector representation of a term. The semantic relatedness of two terms can then be computed as the cosine of their corresponding vectors. ESA can be generalized to work with different document collections, e. g. Wiktionary[2] [15].

Linked Document Collections contain additional information based on the assumption that similar documents should have similar links. Adafre and de Rijke [1] employ a two-step process: first they identify topically related documents using the combined vector-space and boolean information retrieval model implemented by Lucene[3]. The second step is to add links that are missing in the source document, but exist in related documents. West et al. [14] discover missing links in documents by reducing the dimensions of an article-link matrix using *principal component analysis*. They argue that the error introduced when reconstructing the original matrix from the dimension-reduced one is a good indicator of which links are missing in an article. In their evaluation, they show that humans judge the quality of their links better than the ones created by the Milne and Witten [11] approach described above.

3 INEX Link-the-Wiki

INEX 2009 Link-The-Wiki track is an international link discovery competition. We participated in two tasks:

Link-the-Wiki Discover incoming and outgoing document-level links for 5000 orphans (existing Wikipedia articles with links removed). Formally, given a document d, the task is discovering links $l(d, \cdot)$ and $l(\cdot, d)$.

[2] http://www.wiktionary.org
[3] http://lucene.apache.org

Link-Te-Ara Discover outgoing anchor-level links for all Te Ara articles. Formally, given a document d, the task is discovering links $l(d_p, t_e)$ from phrases p in d to entry points e in target document t.

Our interest in this challenge was to experiment with methods of link discovery in both linked and unlinked document collections. Wikipedia is a good collection to test methods that rely on an existing, well-gardened link structure, whereas Te Ara — without any links at all — is well suited to test methods relying on the textual content alone.

3.1 Wikipedia

Wikipedia is a large-scale, general-purpose encyclopedia. The articles are collaboratively created by a large community of users. The community also maintains a very dense and well-gardened link structure connecting the articles. For the INEX challenge, an XML dump of the English Wikipedia was provided[4]. It comprises 2,666,190 individual encyclopedia articles that have been converted from wiki syntax to semantically enriched, well-formed XML documents [12]. Structural elements like sections, paragraphs, etc. are preserved. The annotated XML data accounts for 50.7 GB in size.

3.2 Te Ara

Contrary to Wikipedia, which covers general knowledge, the *Te Ara Encyclopedia of New Zealand*[5] solely focuses on topics related to matters and facts relevant to New Zealand. The texts are authored by local experts of the Ministry of Culture and Heritage[6]. For the challenge, an XML dump of the Te Ara Encyclopedia was provided. It is a lot smaller than Wikipedia, comprising only 438 articles. This results in a total of 3180 XML files (due to each individual article usually consisting of more than one file).

The structure of Te Ara's XML files differs from the Wikipedia collection. Each article is not represented by a single self-contained file, but rather consists of a main file along with one or more resource description documents for multimedia content. Same as the Wikipedia dump, structural elements like headings or paragraphs are annotated in the textual parts of the files. The main difference to the Wikipedia collection is that Te Ara texts do not include any reference links to related articles other than links to resources.

The main file is comprised of a metadata header, an article abstract, and followed by a numbered list of subentries. Subentry, though, is misleading in this case: It actually addresses a certain part of the article and is intended for being rendered on a separate web page. Each subentry encloses an internal index number, a title element and the text body. Any resources—like photos along with their titles and descriptions—are described in separate XML files which are named according to the subentry index they refer to.

[4] Snapshot from Oct 8, 2008
[5] http://www.teara.govt.nz
[6] http://www.mch.govt.nz

UKP_LTWF2F	*Anchor Ranking*			*Target Ranking*	
Experiment ID	**Keyphraseness**	**as$_{\max}$**	**TF.IDF**	**Frequency**	**ESA**
out_k_esa	•				•
out_s_esa		•			•
out_sk_esa	•	•			•
out_sk_freq	•	•		•	
out_tfidf_freq			•	•	

Table 1. Configurations for discovering outgoing document-level links in Wikipedia

4 Link-the-Wiki: Wikipedia Document-Level Links

4.1 Discovering Outgoing Links

To discover outgoing links, we first identify potential anchors, followed by an appropriate target for each anchor, making use of the existing link structure. We combine the *keyphraseness* and $as_{\max}$ measure to rank anchor candidates, and expect this to improve the ranking quality, as the measures are complementary: *keyphraseness* prefers phrases that are often used as anchor, and $as_{\max}$ prefers phrases that have one highly probable link target. To disambiguate between potential targets, we use the ESA semantic relatedness measure based on Wiktionary [15], which captures the article relatedness on a conceptual level. As the measure is based on domain-independent information from a general-purpose document collection (in this case Wiktionary), we do not need to train it for any specific application.

Experiment Configurations We ran different combinations of anchor and target identification in each experiment to discover outgoing document-level links in Wikipedia (Table 1 gives an overview). For each orphan, we perform the following steps:

1. **Preprocessing**: Tokenize, lemmatize[7], and remove stop-words.
2. **Determine anchor candidates**: Each word or phrase in the orphan article that corresponds to a Wikipedia article title (excluding the disambiguation string, denoted as the part in braces in the title) or has been used as link anchor in Wikipedia at least five times (like in [4]).
3. **Prune anchor candidates**: Overlapping anchor candidates are pruned, removing all anchors that are fully contained in another candidate.
4. **Rank anchor candidates**: Using *keyphraseness* (denoted as *k* in the run id), $as_{\max}$ (*s* in the run id), the arithmetic mean of *keyphraseness* and $as_{\max}$ (*sk* in the run id), or tf.idf, denoted as *tfidf*.
5. **Identify targets**: Potential targets are all link targets of the phrase in the existing collection.

[7] Using the TreeTagger [13]

6. **Rank targets**: Take the most frequent target in the collection, denoted as *freq*, or compare the orphan text to all potential target texts using ESA, denoted as *esa*.
7. **Generalize to document level**: Take the highest ranked link target of the highest ranked anchors, until 250 distinct targets have been accumulated.

4.2 Discovering Incoming Links

To discover incoming links, we ran two experiments. In the run with the id *UKP_LTWF2F_in_lucene*, we execute a full-text search for the article title (excluding the disambiguation string) using the combined vector-space and boolean retrieval model as implemented by Lucene, and take the top 250 results as sources of incoming links. In our second experiment, *UKP_LTWF2F_in_esa*, we re-rank the top 2000 search results returned by the Lucene search using ESA, taking into account the semantic relatedness between the orphan and the potential source. Due to the size of Wikipedia and the large computational effort entailed by ESA, we could not use it for the complete retrieval process.

4.3 Qualitative Results

The heuristic of using the most frequent link target for a given anchor works fine for many link anchors, but lacks the ability to adapt to a given context. This is why we used an ESA ranking model which also includes context information as described in Section 2.2. Consider, for example, the Wikipedia article about *Bülent Arınç*, a Turkish politician. In the article text he is said to be born in a city called *Bursa* in Turkey. The most frequent target for this phrase is the article *Bursa Province*, not the city *Bursa*. ESA, on the contrary, identifies the correct target.

Infrequently, both models identify inadequate documents as potential targets. For example, the word *track* in context of a Silverstone Formula One race is linked to an article about *track cycling*, a bicycle racing sport, instead of *Silverstone Circuit*, both by using the most frequent target and using ESA.

In some cases, ESA performs worse than the baseline. A sample sentence is *"As such, it is used for cervical cancer screening in gynecology."* with the underlined word being our link anchor candidate. The baseline approach links to *Screening (Medicine)* which is the correct target. ESA provides a higher value to the *Halftone* article instead, which describes a graphics reproduction technique.

In conclusion, ESA seems to work as expected to disambiguate targets, but not in all cases. The context for ESA based text similarity is the whole article where the anchor appears, which might be too broad and could e. g. be restricted to one paragraph. This will be subject to further research.

5 Link-Te-Ara: Discovering Anchor-Level Links

In this task, the goal is to create links for the complete collection. Outgoing links for all documents include incoming links, so we do not need to distinguish

UKP_LTAraA2B	*Anchor Candidates*		*Target Identification*		*Target Ranking*	
Experiment ID	**Titles**	**Noun Phrases**	**Titles**	**Full Text**	**Lucene**	**ESA**
c_esa	•		•			•
nc_esa		•	•			•
cnc_esa	•	•	•			•
cnc_lucene	•	•	•		•	
cnc_lucene_full	•	•		•	•	

Table 2. Configurations for discovering anchor-level links in Te Ara

between them anymore. In Wikipedia, we gathered anchor candidates based on existing links. To make up for the smaller number of candidates — which are restricted to article titles only because of missing links in Te Ara — we use noun phrases as anchor candidates, in addition to article titles. We use Lucene or ESA to discover appropriate link targets. Using ESA should improve the link discovery in cases where bag-of-word approaches fail due to the vocabulary gap.

5.1 Experiment Configurations

We ran different combinations of anchor identification and target ranking to discover anchor-based links (Table 2 gives an overview). For each document in the Te Ara collection, we do the following:

1. **Preprocessing**: Tokenize, PoS tag, determine noun chunks[8], and remove stop-words.
2. **Determine anchor candidates**: Document and section titles (denoted as *c* in the run id), noun phrases in the document (denoted as *nc*), or a combination of both (denoted as *cnc*) are anchor candidates. Document and section titles are all elements of type *title* in Table 2, described by the XPath. The annotation is restricted to the elements of type *content* in Table 2, as we assume that anchors should only appear in the content parts of a document.
3. **Prune anchor candidates**: Overlapping anchor candidates are pruned, removing all anchors that are fully contained in another candidate.
4. **Rank anchor candidates** using tf.idf.
5. **Remove superfluous anchors**: Take either the best 50 or 6% of terms in the document, whichever yields the lowest number. The Link-Te-Ara task limits the anchor links to a maximum of 50. For shorter documents, we restrict this number even further (in accordance with [4]).
6. **Target identification**: Search for each anchor phrase, either in the titles (see elements specified by XPath in Table 2), or in the complete documents. When searching in titles only, the title's position is the entry point. When searching in complete documents, the entry point is set to the beginning of the document.

[8] Using the TreeTagger [13]

Type	Elements	Namespace
Title	/Entry/Name | //SubEntry/Name	enz.govt.nz/Entry
	//EnglishName	enz.govt.nz/Resources
	//TopicBox/Heading	enz.govt.nz/SubEntrySectionElements
	//h3	w3.org/1999/xhtml
Content	//p | //li | //blockquote	w3.org/1999/xhtml
	//Text	enz.govt.nz/Entry

Table 3. Title and content XML elements in Te Ara

7. **Target ranking**: Using the combined vector-space and boolean retrieval model implemented by Lucene[9] (denoted as *lucene* for title search and *lucene_full* for document search), or using ESA to compare anchor phrases to all titles (denoted as *esa*). We use the top 5 results as targets.

5.2 Qualitative Results

Anchor Identification A snippet of the results of our Te Ara anchor identification is shown in Figure 4, the first paragraph of the article *Wine*.

> Sauvignon blanc, with its grassy smell, put *New Zealand wine* in the international spotlight in the 1980s. Since then, wine *exports* have boomed, with pinot noir another big hit. But for many years, tough *licensing* laws and New Zealanders' taste for fortified wines limited the wine *industry*.

Fig. 4. Comparison of top ranked noun phrases (*nc*) to top ranked *extracted candidates* (*c*) in the abstract of the article "Wine"

All annotated phrases, except *licensing*, are appropriate anchor candidates. The noun phrases are more precise, though, describing more specific concepts. Whereas the titles only allow for the identification of the term *industry*, the noun phrase is in this case more appropriate to the topic of the overall article: the wine industry. In the experimental configuration where we use both types of candidates, only the noun phrase remains. This is because anchor candidates are pruned, and the longer phrase is preferred. Licensing is ranked too low by tf.idf, and not annotated as anchor in the combined run, showing that combining both anchor candidate types can improve the result.

[9] http://lucene.apache.org

Target Identification We will now discuss all three target identification methods (full-text Lucene search, or Lucene search and ESA text relatedness in document and section titles) exemplarily using the anchor "invasive species" taken from the *Biosecurity* article. When executing a full-text Lucene search for the query "invasive species", the *Marine invaders* article is the first target, which discusses invasive species from the seas. It is relevant to the anchor, and although it does not cover the generic concept of invasive species but rather a special case, it includes a perfect definition of the generic concept of "invasive species". The other found articles, *Ants-3*, *Ants*, *Marine invaders-2*, and *New Zealand fauna and flora overseas-5*, are also relevant.

Restricting the search to titles only, the second target includes the section "Introduced and invasive species" in the *Marine invaders* article, a perfect definition for the term. The last target, "Endemic species" in the article *Butterflies & moths*, though, is unrelated to the "invasive" part of the query. Here, a problem of keyword-based search becomes evident — partial matches of the title are often misleading, even though the heads of the noun phrases match.

The last target identification method, title-focused ESA text similarity, has "More new species?" in the *Acclimatization* article as target, which is specifically about the problem of invasive species, and thus is very relevant as background information. The rest of the results are only related to a part of the query, namely "species", and not relevant — the same problem as exhibited by the title-focused search detailed above.

To conclude, we can say that the full-text search is appropriate to get relevant article targets, but the results are sometimes too broad. Restricting the search to titles improves this, but the restriction introduces more results that are irrelevant to the anchor. In our examples, ESA did not improve the result. However, quantitative evaluation results at a larger scale are necessary to draw any final conclusions.

6 Summary

In this paper, we presented our experiments in the Link-the-Wiki track at INEX 2009. We participated in two tasks: discovering document-level links for Wikipedia orphans and discovering anchor-level links in Te Ara. For Wikipedia, we combined *keyphraseness* and *maximum association strength* for anchor ranking with Explicit Semantic Analysis for target disambiguation. For Te Ara, which in contrast to Wikipedia contains no links, we used noun phrase extraction to identify link anchor candidates. To identify entry points, we restricted the search to the article and section titles, using Lucene and Explicit Semantic Analysis to rank them. We concluded with a qualitative discussion of our results. In Wikipedia, Explicit Semantic Analysis was useful to identify the correct link target for ambiguous anchors. In Te Ara, noun phrases serve as good anchor candidates. Restricting the search to document and section titles correctly determined relevant entry points in the majority of examples we analyzed.

7 Acknowledgments

This work has been supported by the Volkswagen Foundation as part of the Lichtenberg-Professorship Program under grant No. I/82806, and by the Klaus Tschira Foundation under project No. 00.133.2008.

References

1. Adafre, S.F., de Rijke, M.: Discovering Missing Links in Wikipedia. In: LinkKDD '05: Proceedings of the 3rd international workshop on Link discovery. pp. 90–97. ACM, New York, NY, USA (2005)
2. Allan, J.: Building Hypertext Using Information Retrieval. Information Processing & Management 33(2), 145–159 (1997)
3. Chen, M.L.E., Nayak, R., Geva, S.: Link-the-Wiki: Performance Evaluation Based on Frequent Phrases pp. 326–336 (2009)
4. Csomai, A., Mihalcea, R.: Linking documents to encyclopedic knowledge. IEEE Intelligent Systems 23(5), 34–41 (2008)
5. Fellbaum, C.: WordNet: An Electronic Lexical Database. MIT Press (1998)
6. Gabrilovich, E., Markovitch, S.: Computing Semantic Relatedness using Wikipedia-based Explicit Semantic Analysis. In: Proceedings of The Twentieth International Joint Conference for Artificial Intelligence. pp. 1606–1611. Hyderabad, India (2007)
7. Geva, S.: GPX: Ad-Hoc Queries and Automated Link Discovery in the Wikipedia. In: Fuhr, N., Kamps, J., Lalmas, M., Trotman, A. (eds.) INEX. Lecture Notes in Computer Science, vol. 4862, pp. 404–416. Springer (2007)
8. Green, S.J.: Building hypertext links by computing semantic similarity. IEEE Transactions on Knowledge and Data Engineering 11(5), 713–730 (Sep 1999)
9. Itakura, K.Y., Clarke, C.L.A.: University of Waterloo at INEX2007: Adhoc and Link-the-Wiki Tracks. In: Focused Access to XML Documents. Lecture Notes in Computer Science, vol. 4862, pp. 417–425. Springer (2008)
10. Milne, D., Witten, I.H.: An Effective, Low-Cost Measure of Semantic Relatedness Obtained from Wikipedia Links. In: Proceedings of the AAAI 2008 Workshop on Wikipedia and Artificial Intelligence (WIKIAI 2008). Chicago, IL (2008)
11. Milne, D., Witten, I.H.: Learning to Link with Wikipedia. In: Proceedings of the 17th ACM Conference on Information and Knowledge Mining. pp. 509–518. ACM, New York, NY, USA (2008)
12. Schenkel, R., Suchanek, F., Kasneci, G.: YAWN: A semantically annotated Wikipedia XML corpus. In: 12. GI-Fachtagung für Datenbanksysteme in Business, Technologie und Web (BTW 2007). Lecture Notes in Informatics, vol. 103, pp. 277–291. Gesellschaft für Informatik, Aachen, Germany (2007)
13. Schmid, H.: Probabilistic Part-of-Speech Tagging Using Decision Trees. In: Proceedings of the International Conference on New Methods in Language Processing. pp. 44–49 (1994)
14. West, R., Precup, D., Pineau, J.: Completing Wikipedia's Hyperlink Structure through Dimensionality Reduction. In: CIKM '09: Proceeding of the 18th ACM conference on Information and knowledge management. pp. 1097–1106. ACM, New York, NY, USA (2009)
15. Zesch, T., Müller, C., Gurevych, I.: Using Wiktionary for Computing Semantic Relatedness. In: Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008. pp. 861–867. Chicago, Illinois, USA (July 2008)