

An Exploration of Learning to Link with Wikipedia: Features, Methods and Training Collection

Jiyin He and Maarten de Rijke

ISLA, University of Amsterdam, Science Park 107, 1098 XG Amsterdam, The Netherlands
{j.he, derijke}@uva.nl

Abstract. We describe our participation in the Link-the-Wiki track at INEX 2009. We apply machine learning methods to the anchor-to-best-entry-point task and explore the impact of the following aspects of our approaches: features, learning methods as well as the collection used for training the models. We find that a learning to rank-based approach and a binary classification approach do not differ a lot. The new Wikipedia collection which is of larger size and which has more links than the collection previously used, provides better training material for learning our models. In addition, a heuristic run which combines the two intuitively most useful features outperforms machine learning based runs, which suggests that a further analysis and selection of features is necessary.

1 Introduction

The aim of the LTW track is to automatically identify hyperlinks between documents. We only participated in the task of outgoing link generation within Wikipedia (A2B). In our experiments, we focus on exploring machine learning methods and learning material for link detection. The main purpose of our experiments is two-fold. First, we want to test how our learning methods work on the LTW task, especially how the results learnt from the Wikipedia ground truth will be judged by human assessors. On top of that, since the LTW task is defined as a ranking problem for recommendation purposes, we want to see how a learning to rank approach works as it directly optimizes the rankings instead of assigning binary decisions to candidate links as a classification method would do. Second, we trained our models with different versions of Wikipedia. The two versions used, namely Wikipedia 2008 and Wikipedia 2009, differ in the amount of articles they contain as well as in the amount of links. Presumably, the 2009 version contains more link information but is also more noisy in terms of missing target pages (as some pages are deleted as time passes by). We experiment with both collections so as to see the impact of the training material used. In addition, we explore a set of features for constructing the classifiers/rankers. In order to examine the effectiveness of the features, we also submit a heuristic run that combines the two intuitively most useful features without sophisticated learning methods.

More specifically, we have following research questions:

- When the A2B task is viewed as a ranking problem, is a learning to rank approach more effective than a binary classification approach?

Learning Stage	N-gram	N-gram-target	Target	N-gram-topic	Topic-target	1st-stage
Candidate targets ranking		x	x		x	
Candidate links ranking	x	x	x	x	x	x

Table 1. Features and their corresponding application in different learning stages.

- Do different versions of the Wikipedia collection (with, potentially, differences in collection size, numbers of links, etc.) result in performance differences when used as training material?
- Are the features used for learning the models effective? Are there single features whose contribution to the linking results is dominant?

The rest of the paper is organized as follows: Section 2 introduces the learning approaches applied to our task, Section 3 describes the features and Section 4 presents the runs we submitted plus an analysis of the evaluation results. We conclude in Section 5.

2 A Two-Stage Learning Procedure

Following [5], we consider the linking task as a two stage procedure, namely *candidate target identification* and *link detection* [2]. First, we extract all n-grams in a topic page and train a target-detector to rank the potential target pages for each n-gram, which we refer to as *candidate target pages*. Then we train a link detector on the (n-gram, target) pairs for the final results.

We experiment with two types of learning methods, viz. classification and learning to rank. For classification, we use SVM to classify the instances in both stages and rank the results by the probability of an instance being positive. For our learning to rank approach, we use RankingSVM [3] to directly optimize the ranking of an instance. In the candidate target identification stage, we train a ranker to rank the target candidates for each n-gram and in the link detection stage a ranker is trained to rank the n-gram target pairs.

For learning both the binary SVM and the RankingSVM, we randomly sample 500 pages from the Wikipedia collection for training and 100 pages for validation. For both SVMs we use the linear kernel and tune the regularization parameter on the validation set. To learn the model for target detection, we use only the real anchor texts and their corresponding candidate target pages as instances, while for training the link detectors we use all n-gram-candidate target pairs as instances.

3 Features

We identify 6 types of feature for learning a preference relation between the candidate links. Table 1 specifies in which stage each type is used and Table 2 lists the features. Here, we discuss the motivations for using them and detail the formulation of some.

N-gram features The n-gram features suggest how likely a given n-gram would be marked as an anchor text, without any other information such as its context in the topic

page, which includes its length, IDF score, the number of candidate targets associated with it, and its *ALR* (Anchor Likelihood Ratio) scores. IDF is calculated as

$$\log \left(\frac{|D|}{|\{d_i : ng \in d_i\}_{i=1}^N|} \right),$$

where ng is a n-gram, d_i is a page containing this n-gram, and $|D|$ is the total number of pages in the Wikipedia collection. The *ALR* score can be interpreted as a model selection between two models, the anchor model and the collection model, from which an n-gram is generated. To calculate the probability of an n-gram being generated by either model, the maximum likelihood is used. Specifically, it is calculated as

$$ALR(ng) = \frac{|ng \in A|}{|A|} \cdot \frac{|C|}{|ng \in C|}, \quad (1)$$

where A is the collection of all anchor texts in the Wikipedia collection and C is the collection of all n-grams in the Wikipedia collection, and $|\cdot|$ means the total number of anchor texts/n-grams in the given collection. A large *ALR* value indicates that the n-gram is more likely to have been generated from the anchor model, i.e., this n-gram is more likely to be an anchor text than a common word sequence from the background collection.

N-gram-target features The n-gram-target features describe how well an n-gram and its corresponding candidate target page are related. On the assumption that each Wikipedia page is about a specific concept that is usually denoted by its title, the first feature we use is the match between an n-gram and the candidate target page. The second type of feature in this category consists of indicators of how likely a given n-gram ng and a candidate target page $ctar$ are linked, which is expressed by the following two scores: *RatioLink* and *RatioAnchor*. The former is the ratio between the number of times ng and $ctar$ are linked and the number of times $ctar$ is being linked as a target page in the collection. The latter, i.e., *RatioAnchor* is the ratio between the number of times ng and $ctar$ are linked and the number of times ng is used as an anchor text in the collection. Moreover, we adopt retrieval scores between the n-gram and the candidate target pages as features (n-gram as query), which is an obvious description of the relatedness of the two:

$$RatioLink(ng, ctar) = \frac{|link(ng, ctar)|}{|inlink(ctar)|} \quad (2)$$

$$RatioAnchor(ng, ctar) = \frac{|link(ng, ctar)|}{|ng \in A|} \quad (3)$$

Here, $|link(ng, ctar)|$ denotes the number of times that n-gram ng and $ctar$ are linked in Wikipedia, and $|inlink(ctar)|$ denotes the number of times that $ctar$ is used as a target page and linked with some anchor texts in Wikipedia.

Target features The target features are indicators of how likely a candidate target page alone would be linked with some anchor text in the collection. To this end we explore features such as counts of the inlinks and outlinks within the candidate target page, as well as the Wikipedia category information associated with it.

N-gram features	
Length(ng)	Number of words contained in the n-gram
IDF(ng)	The IDF score of the n-gram
ALR(ng)	The ALR score of the n-gram, as detailed in Eq. 1
#Cand(ng)	Number of candidate target pages associated with the n-gram
N-gram - target features	
TitleMatch(ng, ctar)	Three values - 2: exact match; 1: partial match (i.e., either the title contains the n-gram, or the n-gram contains the title); 0: no match
RatioLink(ng, ctar)	Link ratio of the n-gram and the candidate target page, see Eq. 2
RatioAnchor(ng, ctar)	Anchor ratio of the n-gram and the candidate target page, see Eq. 3
Ret_uni(ng, ctar)	Retrieval score with unigram model, i.e., BM25 with default parameter settings
Ret_dep(ng, ctar)	Retrieval scores with dependency model, i.e., Markov Random Field model as described in [4]
Rank_dep(ng, ctar)	The rank of the target page with the dependency retrieval model
Target features	
#Inlinks(ctar)	Number of in-links contained in the candidate target page
#Outlinks(ctar)	Number of out-links contained in the candidate target page
#Categories(ctar)	Number of Wikipedia categories associated with the candidate target page
Gen(ctar)	Generality of the candidate target page as described in [5]
N-gram - topic features	
TFIDF(ng, topic)	The TFIDF score of the n-gram in the topic page
First(ng, topic)	Position of first occurrence of the n-gram in the topic page, normalized by the length of the topic page
Last(ng, topic)	Position of last occurrence of the n-gram in the topic page, normalized by the length of the topic page
Spread(ng, topic)	Distance between first and last occurrence of the n-gram in the topic page normalized by the length of the topic page
Topic-target features	
Sim(ctar, topic)	Cosine similarity between the candidate target page and the topic page
Ret_unigram(ctar, topic)	Retrieval score using the title of the candidate target page as query against the topic page; using BM25 as retrieval model
First stage scores	
score(ng, ctar)	The output of the ranker for the candidate target page given the n-gram
rank(ng, ctar)	The rank of the candidate target page according to the learnt ranker

Table 2. Features used for learning the preference relation, where ng: n-grams; C: collection; ctar: candidate target pages; topic: topic page.

N-gram-topic features This type of feature describes the importance of the n-gram within its context, i.e., topic page. One would assume that an n-gram being selected as an anchor text should be somewhat important to the understanding of the whole topic page as well as being content-wise related. Here, we use the TFIDF score of the n-gram and its location within the topic page as an indication of the importance of a n-gram within a topic page.

RunID	Description
UvAdR_LTWA2B_01	Binary classification, trained on wiki08
UvAdR_LTWA2B_02	Ranking SVM, trained on wiki08
UvAdR_LTWA2B_03	A heuristic run, combine the ALR and IDF for link ranking, but using rankingSVM for target ranking
UvAdR_LTWA2B_04	Binary classification, trained on wiki09
UvAdR_LTWA2B_05	Ranking SVM, trained on wiki09

Table 3. Submitted runs.

Topic-target features The topic-target features describe the relatedness between a topic page and a candidate target page. One obvious feature is the similarity between the two pages. In addition, as a candidate target page itself is about a concept, we could measure how important this concept is, or in other words, how well this concept is being expressed in the topic page. We measure it by using the title of the candidate target page as a query and calculating the retrieval score against the topic page.

First stage score Once the target ranking has been completed (during the first stage), we can get the ranking score for each candidate target, as well as their ranks. In the second stage, we select the top X candidate targets to construct the candidate links with their corresponding n-grams, where the scores and ranks from the first stage are used as features.

4 Five Runs

4.1 Submitted runs

We submitted 5 runs for the LTW task, as specified in Table 3. We use two Wikipedia collections, wiki08 [1] and wiki09 [6]. Note that we have a heuristic run UvAdR.-LTWA2B_03 that does not use a learning method for link ranking; it only uses RankingSVM for target identification. For link ranking, we filter the candidate links whose ALR score is less than 0.2, and rank the remaining ones with their IDF scores. This run serves as a baseline for other machine learning based approaches. The heuristics used in this run, i.e., the *ALR* and *IDF* scores, however, are the features that are most close to human intuitions, where *ALR* represents how likely an n-gram is involved in a link based on the observation of existing links and *IDF* represents the uncommonness of a n-gram.

For all 5 runs, for each detected link, we set the best entry point to 0, as intuitively, the first paragraph of a Wikipedia page gives a good summary of the concept of the linked anchor text.

4.2 Results and analysis

Figure 1 shows the results of our submitted runs using a Precision-Recall plot.

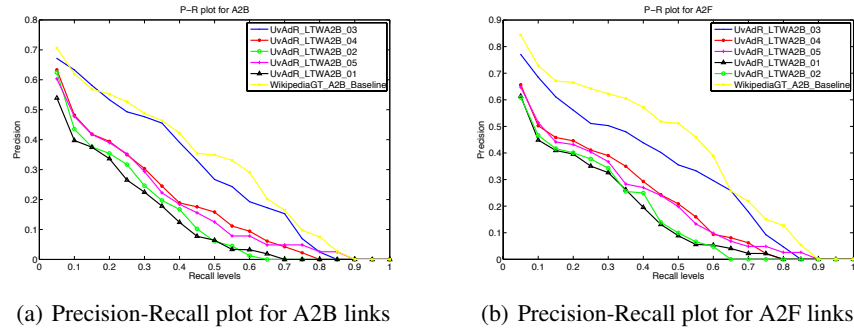


Fig. 1. Precision-Recall plots for the submitted runs.

In this year’s evaluation, the A2B runs are evaluated at two levels, i.e., at the Anchor-to-BEP level as well as at the Anchor-to-File (A2F) level. For the A2F evaluation, all BEP positions are set to 0. Since we set all BEP positions to 0 in our submitted runs, it is more natural to focus on the results of the A2F evaluation. Table 4 lists the interpolated precision scores at different recall levels. The Wikipedia ground truth is included as a pseudo run and evaluated against the manual assessment.

RunIDs	R@0.05	R@0.1	R@0.2	R@0.5
UvAdR_LTWA2B_03	0.77	0.68	0.56	0.35
UvAdR_LTWA2B_04	0.65	0.50	0.44	0.21
UvAdR_LTWA2B_05	0.64	0.51	0.43	0.20
UvAdR_LTWA2B_01	0.61	0.44	0.39	0.09
UvAdR_LTWA2B_02	0.60	0.46	0.40	0.10
WikipediaGT_A2B_Baseline	0.84	0.73	0.66	0.51

Table 4. Average precision at different recall levels, evaluated with A2F.

For both levels of evaluation (using A2B and using A2F), the best run is the heuristic run, which outperforms all sophisticated learning methods. This indicates that the two features, ALR and IDF, are very strong features that probably dominate the contribution to the learned models. In addition, this observation suggests that a detailed analysis and selection of features should be conducted.

Next, from the A2F evaluation, we see that runs trained on the Wikipedia 09 collection (i.e., 04 and 05) outperform runs trained on the Wikipedia 08 collection (i.e., 01 and 02). This suggests that a larger collection with more (existing) links provides better training materials. Also, we see that the runs based on binary classification methods (01 and 04) do not differ a lot from the learning to rank based runs (02 and 05). This may be due to the fact that the training examples from Wikipedia do not contain very strong ranking information, i.e., we only have two levels of relevance: relevant (is a link) and non-relevant (not a link).

Finally, none of our runs outperforms the Wikipedia ground truth. This is no surprise, since the models are learned from the Wikipedia ground truth. In order to outper-

form the Wikipedia ground truth with a learning method, sufficiently many examples with manual labeling should be collected.

5 Conclusions

We have described our approaches and submissions for this year's participation in the INEX Link-the-Wiki track. We submitted 5 runs to the A2B outgoing links detection task. Our main focus was to explore the effectiveness of applying machine learning approaches for the task. Specifically, we experimented with two types of learning approaches, namely classification and learning to rank. We also evaluated the learning material for the task, where we use different sets of training data (based on different versions of Wikipedia). On top of that, we used a heuristic run to exam the impact of the features that are intuitively most effective.

We have found that the learning to rank based approach and the binary classification approach do not differ a lot. The new (2009) Wikipedia collection which is of larger size and has more links than the old (2008) collection, provides better training material for learning the models. In addition, the heuristic run outperforms all machine learning based runs, which suggests that a further analysis and selection of features is necessary.

Acknowledgements This research was supported by the European Union's ICT Policy Support Programme as part of the Competitiveness and Innovation Framework Programme, CIP ICT-PSP under grant agreement nr 250430, by the DuOMAn project carried out within the STEVIN programme which is funded by the Dutch and Flemish Governments under project nr STE-09-12, and by the Netherlands Organisation for Scientific Research (NWO) under project nrs 612.066.512, 612.061.814, 612.061.815, 640.004.802.

Bibliography

- [1] L. Denoyer and P. Gallinari. The Wikipedia XML Corpus. *SIGIR Forum*, 2006.
- [2] J. He and M. de Rijke. A ranking approach to target detection for automatic link generation. In *SIGIR '10: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM New York, NY, USA, 2010.
- [3] R. Herbrich, T. Graepel, and K. Obermayer. *Large margin rank boundaries for ordinal regression*. MIT Press, Cambridge, MA, 2000.
- [4] D. Metzler and W. Croft. A markov random field model for term dependencies. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 472–479. ACM New York, NY, USA, 2005.
- [5] D. Milne and I. H. Witten. Learning to link with wikipedia. In *CIKM '08: Proceedings of the 17th ACM conference on Information and knowledge management*, pages 509–518, New York, NY, USA, 2008. Acm.
- [6] R. Schenkel, F. Suchanek, and G. Kasneci. YAWN: A semantically annotated Wikipedia XML corpus. In *BTW2007*, 2007.