Automatic Link Detection: A Sequence Labeling Approach

James Gardner
Department of Mathematics and Computer Science
Emory University
jgardn3@emory.edu

Li Xiong
Department of Mathematics and Computer Science
Emory University
lxiong@mathcs.emory.edu

ABSTRACT

The popularity of Wikipedia and other online knowledge bases has recently produced an interest in the machine learning community for the problem of automatic linking. Automatic hyperlinking can be viewed as two sub problems – link detection which determines the source of a link, and link disambiguation which determines the destination of a link. Wikipedia is rich corpus with hyperlink data provided by authors. It is possible to use this data to train classifiers to be able to mimic the authors in some capacity. In this paper, we introduce automatic link detection as a sequence labeling problem. Conditional random fields (CRFs) are a probabilistic framework for labeling sequential data. We show that training a CRF with different types of features from the Wikipedia dataset can be used to automatically detect links with almost perfect precision and high recall.

Categories and Subject Descriptors

I.2.7 [**Artificial Intelligence**]: Natural Language Processing – *text analysis.*; I.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing – *linguistic processing.*

General Terms

Algorithms, Design, Experimentation.

1. INTRODUCTION

Collaborative online encyclopedias or knowledge bases such as Wikipedia[1] have become extremely popular because of their open access, comprehensive and interlinked content, rapid and continual updates, and community interactivity. Wikipedia's popularity has made it the largest, most visited encyclopedia in history. The concepts in Wikipedia are understood more easily due to the hypertextual links between concepts that authors have provided in their articles. This allows a user to "jump" to requisite concepts in the network all the way "down" to the concepts that are evident to the reader's intuition in order to more fully understand the current one.

[1]http://www.wikipedia.org

Problem Definition. Hyperlinking a corpus of articles consists of two sub-problems. Determining the source of a link is called *link detection* or *anchor detection* and determining the destination of a link is called *link disambiguation.* The main analytic challenges for hyperlinking lie in how to accurately determine which terms or phrases to link and which articles to link to.

Existing Solutions and Their Limitations. The existing approaches for hyperlinking can be mainly classified into three categories, namely, *manual linking*, *semi-automatic linking*, and *automatic linking.* Manual linking refers to the linking technique where both the link source and link target are explicitly defined, e.g. anchor tags in html documents. Most web pages use the manual approach. Semi-automatic linking refers to the technique where the link source is manually marked by the author and the system performs the link disambiguation. Many online collaborative systems (including Wikipedia) use the semi-automatic approach. Automatic linking refers to the technique where the system automatically performs link detection and link disambiguation.

The existing automatic approaches for hyperlinking can be further classified into two main categories. The NNexus system [1] uses meta data and a *rule-based* approach for determining the link sources (any concept that is defined in the corpus will be linked) and link targets (the link candidate that is closest to the link source in the classification tree is the link target). While the system has demonstrated its effectiveness through the PlanetMath corpus, it is not feasible for a corpus such as Wikipedia in which virtually every single term is defined in the corpus and applying NNexus directly will produce an extremely overlinked system.

The popularity of Wikipedia has also recently produced an interest of *machine learning* based techniques for the problem of automatic linking. The existing manually linked pages in Wikipedia are highly accurate [9] and provide a good training set for machine learning based automatic linking [8, 7, 6]. While the machine learning approach using Wikipedia has shown its effectiveness [8], link detection remains a difficult problem because it is much harder to determine the importance and link-worthiness of a term (whether it should be linked) than determining the most relevant meaning of a term in a document (where it should be linked to). In addition, the current machine learning approaches for link detection [8] only detect the concepts that should be linked for a document but *not* which occurrence of the concepts should to be linked (when there are multiple concepts and different authors may have different styles of linking).

Contributions. This paper investigates the problem of automatic link detection using machine learning approaches. In particular, we model automatic link detection as a sequence labeling problem. We try to learn how the authors link, inline, rather than learn what concepts or key phrases (n-grams) should be linked. In contrast to the previous approaches, we are able to mimic the authors' linking style from the corpus and determine exactly which terms or phrases in the document should be linked. We experimentally evaluate our system using subsets of the Wikipedia corpus. We also compare our approach with the previous approaches and show the effectiveness of our approach.

2. METHODOLOGY

In this section, we describe how we model the problem as a sequence labeling problem and the classifier and features we use for our approach.

2.1 Overview

We model link detection as a sequence labeling problem. It is aimed to learn how the authors link, inline, rather than learn what concepts or key phrases (n-grams) the authors typically link. In other words, we do not extract all the n-grams from the text but instead treat the text inline and use only features of the terms without having chosen predetermined n-grams.

We now describe the sequence labeling problem in detail. Given a sequence of data, we want to assign a label to each element in the sequence. The sequence labeling problem is often seen in Natural Language Processing tasks. Labeling each word in a sentence with the appropriate part of speech is an example of a sequence labeling problem.

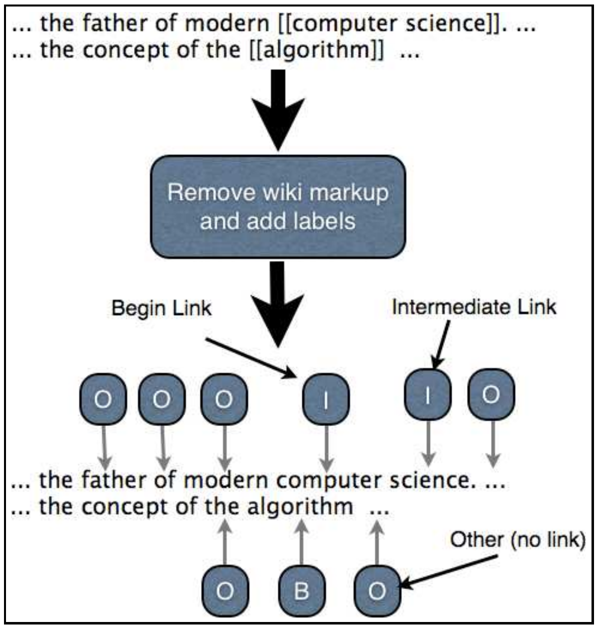
... the father of modern [[computer science]]. ...
... the concept of the [[algorithm]] ...
Remove wiki markup and add labels
Begin Link
Intermediate Link
O
O
O
I
I
O
... the father of modern computer science. ...
... the concept of the algorithm ...
Other (no link)
O
B
O

Figure 1: Generating Sequence Labeling Data

We inherently model arbitrarily long n-grams by using the following labels. We label each token (discussed below) in the text as either Other (no link), B-link (begin link), or I-link (intermediate link). If we are given the sentence "the father of modern computer science" and wish to link computer science, we would want to assign a label of Other to "the", "father", "of", and "modern". We would then label "computer" with B-link and "science" with I-link. Similarly the term after "science" would be labeled with Other. See Figure 1 for a visual example of converting Wikipedia markup into sequence labeling data. Each term is associated with a label of either Begin Link, Intermediate Link, or Other (no link).

The sequence labeling data is in the form of feature vectors. Initially each feature vector contains a label and a term. In order to help the classifier determine which label to assign to a term we add a variety of features to the feature vector based on the term.

2.2 Features

In order to construct a good classifier for link detection we must find which features are important in determining whether a term should be linked in the text. We use three types of features: local features, document features, and corpus features.

Local Features. Local features are features of the term in the text. Local features can further be broken into syntactic and semantic features. Semantic features are derived from the term and maybe from surrounding terms but the feature itself is specific to the current term, e.g. POS. POS would not be possible to calculate for a specific term without taking into account surrounding terms, but the POS is associated with only the term itself. We call these features local semantic features. Features that are the result of regular expressions match, e.g. the existence of special characters and capitalization (sometimes referred to as the shape of a term in NLP literature) are called local syntactic features. We also consider word stems to be syntactic features because the stem replacement is based on regular expressions.

Document Features. Document features can be computed from the current document or string of text being used to train the classifier. Determining whether a term should be linked is dependent upon context. One easy way to obtain context for a term is looking at the n previous and next words. Document features include n previous and next words, occurrence statistics (e.g. as first occurrence, second occurrence, . . .), and other features computed on the previous and next tokens.

Corpus Features. Corpus features are features that are computed on the entire training set. Typically, terms that are relatively important in a document should be linked. The *tf-idf* (term frequency - inverse document frequency) score is often used in information retrieval and gives a measure of how important a term is in a document collection. Let N be the number of document in the collection. Let $tf_{t,d}$ be the number of times a term t occurs in the document d divided by the number of terms in document d. Let df_t be the number of documents that contain term t. Let $idf_t = \log(N/df_t)$. Then $\text{tf-idf}_{t,d} = tf_{t,d} \cdot idf_t$ is the tf-idf weight of term t in document d. The label probability is the probability that a term is labeled with the label in the training set. This feature gives us a prior inclination to whether the term should or should not be linked in the document.

External Features. We use external features to refer to features that are computed over datasets that are not part of the training data. These features can include the corpus-based features but computed over external knowledge bases such as the the entire Wikipedia corpus (not just the training set) or other external data that can be used to supplement the training data. These can include tf-idf, link structure, link probability, and relatedness (how related a concept target is to a destination term [8])

It is worth noting that supplementing training data with external knowledge bases such as Wikipedia may not always be a good idea for the link detection problem. This is especially true if the training data has a completely different linking style from Wikipedia. As a result, this paper does not focus on using external data to enhance the classifier but instead introduces the link detection model and other features of the model.

2.3 Classifiers

A conditional random field (CRF) [3] is an advanced discriminative probabilistic model that is shown to be effective in labeling sequential data including natural language text. A CRF takes as input a sequence of feature vectors generated from the text where each feature vector consists of a label, token and set of features based on the sequence. Given a token from the sequence it calculates the probabilities of the various possible labelings (whether it is a B-link, I-link, or Other) and chooses the one with maximum probability. The probability of each label is a function of the feature set associated with that token. More specifically, a CRF is an undirected graphical model that defines a single log-linear distribution function over label sequences given the observation sequence. The CRF is trained by maximizing the log-likelihood of the training data. Given that CRFs have been shown to be effective at solving a variety of sequence labeling problems [3, 4], we use them as our classifier for link detection.

The process of detecting links involves generating the features and training a Conditional Random Field model based on those features. The model is then used to predict the labels of the terms (feature vectors) that the classifier has not seen. Section 3 shows that training a CRF with different types of features from the Wikipedia dataset can be used to automatically detect links with almost perfect precision and high recall.

3. EXPERIMENTS

This section describes a set of initial experiments verifying the feasibility and benefit of the sequence labeling approach. We created a set of data conversion and feature generation modules, and a CRF classifier based on the Mallet toolkit [5].

3.1 Datasets

We constructed our training and testing dataset from Wikipedia. Wikipedia uses a semi-automatic linking system. Wikipedia's linking system does not completely disambiguate a link but instead relies on disambiguation pages as a substitute to full disambiguation. It is interesting to note that artificial hubs are created in the Wikipedia link network because of disambiguation pages. This may have impact on some algorithms that use the link structure of a semantic network such as HITS [2]. Disambiguation pages paradoxically add ambiguity to the data because the link structure is modified and it encourages authors not to find the correct target for a link. However, even though Wikipedia links do not provide an exact link destination it provides a well marked dataset for our problem of link detection. That is, because all the links in Wikipedia are manually provided by the authors we can assume to a certain degree that Wikipedia is a gold standard.

The experimental dataset was taken from an xml dump of Wikipedia's content (08/13/2008). We then extracted subsets of the Wikipedia pages based on category. We only extracted pages that contained at least three links. We selected articles with more than three links to avoid short articles such as image and special pages. We extracted subsets of 500 articles in 7 different categories (Random, Biology, Business, Health, Language, Mathematics, and People).

Each token in the sample wiki text was converted into a feature vector and associated with a label of either I-link, B-link, or O (no link) based on the wiki markup indicating links in each article as discussed in Section 2. We used k-fold cross validation for our experiments. We generated k subsets of each category sample. We then added features to each test and training set based on our discussion in Section 2.2. When calculating the corpus features for each training set we ensure that the statistics are computed only on the training set and not the testing set. The corpus features for the testing sets were computed over the testing and training sets.

3.2 Evaluation Metrics

We report the precision, recall, and f-score of each label in the sequence labeling. Precision is defined as the number of correct labels divided by the number of labels output by the classifier. Recall is defined as the number of correct labels divided by the number of labels in the testing set. The f-measure is the harmonic mean of precision and recall computed as $F = 2 * P * R/(P + R)$.

In order for a better understanding of the accuracy of the link detection and for comparison to other approaches we converted the B-link and I-link labels into phrases. A phrase is simply all the terms starting from a B-link up to the last I-link. We also report precision, recall, and f-measure for the phrases linked in the articles.

The results reported for the sequence labeling performance indicate the accuracy of linking in-line. That is, if "The United States" is linked in the article in two places and we only link one of them, then our performance results will be lower. When considering phrase detection we would only need to link "The United States" in one of the locations and the precision and recall will not suffer. This is the same as reported in [8].

3.3 Impact of Different Categories

We ran our system for the datasets of different categories and the results are shown in Tables 1 and 2. In general, the classifier performs consistently across different categories of the Wikipedia corpus. It performed best on the Biology category achieving precision of .83, recall of .65, and f-score of .67 in the phrase linking experiments. The Business category proved hardest to disambiguate with a precision of .72,

Category	Precision	Recall	F-Score
Random	.78	.60	.67
Biology	.83	.65	.73
Business	.72	.56	.63
Health	.79	.59	.68
Language	.79	.61	.69
Mathematics	.76	.55	.64
People	.73	.62	.67

Table 1: Comparison of Categories: Phrase Linking Results

	B-Link			I-Link		
Category	Prec	Rec	F	Prec	Rec	F
Random	.96	.64	.77	.99	.65	.78
Biology	.97	.67	.80	.99	.64	.78
Business	.96	.58	.72	.99	.63	.77
Health	.98	.60	.74	.99	.62	.76
Language	.97	.64	.78	.99	.64	78
Mathematics	.95	.57	.72	.99	.64	.78
People	.96	.65	.78	.98	.67	.80

Table 2: Comparison of Categories: Sequence Labeling Results

recall of .56, and f-score of .63. The precision, recall, and f-score values of each of the categories is not far different from the results on the Random category.

3.4 Sampling (Impact of the Class Distribution)

As we can see from the results so far, our classifier has relatively high precision but still suffers from low recall for the class of B-links and I-links. One possible reason is that the B-links and I-links are rare compared to the Other labels (non-linked terms) and the classifier are trained to recognize the Other labels dominantly. This is a common problem in classification and a proven technique for addressing the non-uniform misclassification cost is the cost-proportionate sampling approach [10]. We experimented with a variation of the sampling technique in order to improve the results of the classifier by biasing it more toward selecting links. In order to bias the classifier we split the text in the training sets into sentences. If the sentence contained a link then it is selected and if the sentence did not contain a link we selected it with probability .1. The sampled training sets were then used to train CRF classifiers. The CRF models were then used to predict the labels of the original test sets (no sampling was done on the test sets).

The results of biasing the class distribution towards more links can be seen in Table 3. The sentence sampling increased both the precision and recall of the classifier. The precision increased from .758 to .918, the recall increased from .597 to .675, and the f-score increased from .668 to .778.

	Precision	Recall	F-Score
Milne	.744	.738	.741
CRF (w/ Sampling)	.918	.675	.778

Table 3: Comparison of CRF Sequence Labeling Approach to Milne

3.5 Comparison with Milne Approach

Although our original goal was to detect the actual location of links, we can compare our phrase linking experiments to those reported in [8]. Table 3 shows the results of our classifier with sentence sampling to the results reported in [8]. Milne and Witten trained a C4.5 classifier and reported recall of .738, precision of .744 and f-score of .741. Our method has much higher precision but lower recall. This can partly be explained by the fact that our algorithm does not specifically test all possible phrases (n-grams) in a document. However, we do have the advantage of determining the exact occurrences of the phrases to link (when there are multiple occurrences of the same phrase in the article). In addition, we expect the sampling technique with a higher sampling rate will help further improve our results.

4. CONCLUSIONS AND FUTURE WORK

We have described our sequence labeling based approach for automatic link detection. Our initial results show that it achieves almost perfect precision and high recall. Our work continues along several directions. First, we plan to conduct a thorough error analysis and incorporate new features to further enhance the recall of our system while maintaining the high precision. Second, we are investigating possible techniques to address the performance concerns of computing the features over a large dataset and training the CRF over a large feature vector set. Finally, we are developing an easy to use sequence labeling toolkit based on the techniques described in this paper. We are releasing the dataset and our system as open source software for others to be able to run their own tests, reproduce our results, and enhance the system for their own use.

5. REFERENCES

[1] James Gardner, Aaron Krowne, and Li Xiong. NNexus: An automatic linker for collaborative web-based corpora. *IEEE Transactions on Knowledge and Data Engineering*, 21(6):829–839, 2009.

[2] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, 1999.

[3] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, pages 282–289, 2001.

[4] Andrew McCallum. Efficiently inducing features of conditional random fields. In *UAI*, pages 403–410, 2003.

[5] Andrew K. Mccallum. Mallet: A machine learning for language toolkit. 2002.

[6] O. Medelyan, I. H. Witten, and D. Milne. Topic indexing with wikipedia. In *In Proc of Wikipedia and AI workshop at the AAAI-2008 Conference*, 2008.

[7] Rada Mihalcea and Andras Csomai. Wikify!: linking documents to encyclopedic knowledge. In *CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 233–242, New York, NY, USA, 2007. ACM.

[8] David Milne and Ian H. Witten. Learning to link with wikipedia. In *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management*, pages 509–518, New York, NY, USA, 2008. ACM.

[9] Gabriel Weaver, Barbara Strickland, and Gregory Crane. Quantifying the accuracy of relational statements in wikipedia: a methodology. In *JCDL '06: Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries*, pages 358–358, New York, NY, USA, 2006. ACM.

[10] Bianca Zadrozny, John Langford, and Naoki Abe. Cost-sensitive learning by cost-proportionate example weighting. In *ICDM '03: Proceedings of the Third IEEE International Conference on Data Mining*, page 435, Washington, DC, USA, 2003. IEEE Computer Society.