Humboldt-Universität zu Berlin
Mathematisch-Naturwissenschaftliche Fakultät
Institut für Informatik

In-site Hypertext Link Prediction

Exposé

Author: Alpay Yilmaz

Evaluators: Prof. Dr. Ulf Leser

submitted at: .....

Introduction

A company owns an online magazine with 12.629 articles. These articles contain hyperlinks, which were set manually by the author of the article (approx. 33.000 hyperlinks). Those hyperlinks point to other articles of the online magazine and enable the readers to quickly find/read relevant articles. The phrases (the string in which the link is placed) contains of one or more words. The main goal of this procedure is to maximizes the session duration of the readers.

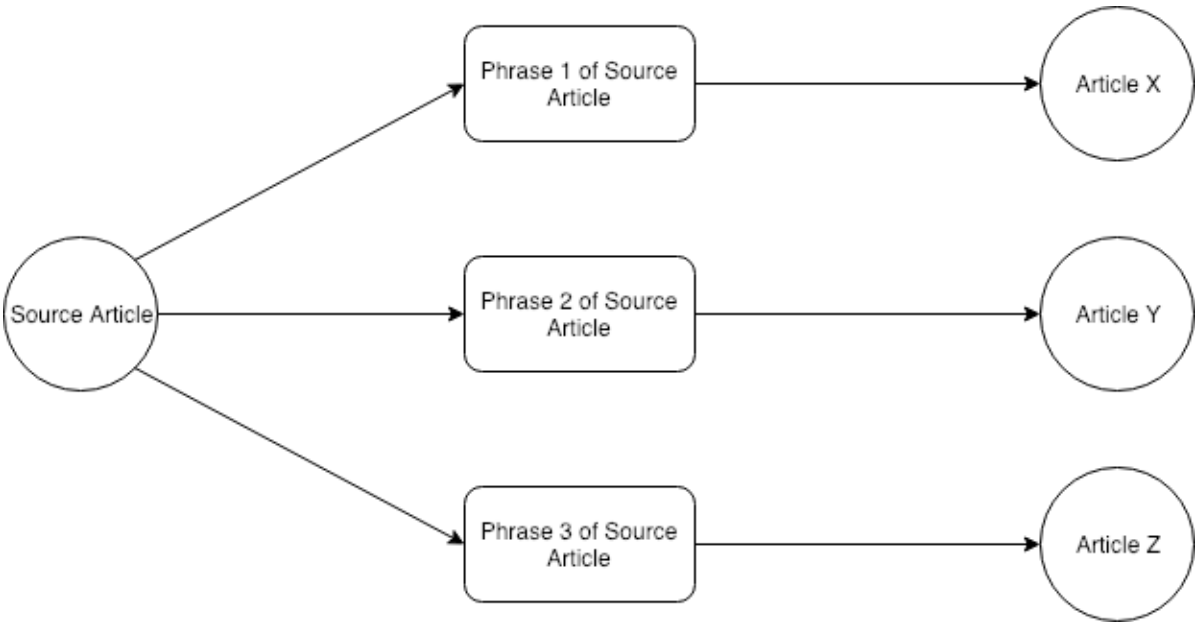
All articles are stored in a structured relational database and anonymous user data is collected via Google Analytics.

Manual linking of articles costs the company resources, mainly time. To save these resources, we consider an automated approach. The goal is to automatically place the hyperlinks for each article, be it newly written or already published, in a way similar to how an experienced author or a SEO-expert would place it.

We will analyze how different factors influence the authors decision in linking the articles with a key phrase.

Objectives

Figure 1: Selecting links for an article



We can separate our problem in two separate problems:

The first one is choosing the most optimal phrase(s). A phrase is the string in the article which contains the hyperlink. When choosing a phrase we have to consider/optimize following points:

- **Selection of strings**
  Which string do we choose? How many words should a phrase consist of?
- **Relevance of phrase**
  How relevant should the phrase be to the article containing it?
- **Number of phrases**
  How many phrases do we choose in an article?
- **Distance between phrases**
  How long should the distance between phrases be?

The second problem is choosing the article X to which the phrases links to. When choosing the article X we have to consider/optimize following:

- **Relevance of linked article to the phrase**
  How relevant should the linked article be to the phrase?
- **Similarity of linked article to the source-article**
  How similar should the source-article be to the linked article.
- **Relevance of the word containing the link**
  Which words should be links?
- **Performance of linked article**
  How performant should the linked article be?

We will analyze how the authors weighted these factors for linking articles based on their past data.

Related Work

Key-Phrase Extraction

Graph Based Approach

The basic idea behind a graph-based approach is to build a graph from the input document and rank its nodes according to their importance using a graph-based ranking method [3]. **Textrank [6]** is one of the most well-known graph-based approaches to key-phrase extraction [3]. Each node of the graph corresponds to a candidate key-phrase from the document and an edge connects two related candidates. The edge weight is proportional to the syntactic and/or semantic relevance between the connected candidates. For each node, each of its edges is treated as a "vote" from the other node connected by the edge. A node's score in the graph is defined recursively in terms of the edges it has and the scores of the neighboring nodes. The top-ranked candidates from the graph are then selected as key-phrases for the input document [3] [6].

Topic-Based Clustering

Another approach to key-phrase extraction involves grouping the candidate key-phrases in a document into topics, such that each topic is composed of all and only those candidate key-phrases that are related to that topic [3]. **KeyCluster** [3] adopt a clustering-based approach that cluster semantically similar candidates using Wikipedia and co-occurrence-based statistics. The underlying hypothesis is that each of these clusters corresponds to a topic covered in the document, and selecting the candidates close to the centroid of each cluster as key-phrases ensures that the resulting set of key-phrases covers all the topics of the document [5].

Topical PageRank [4] is an approach that overcomes one weakness of KeyCluster: It does not give each topic equal importance [4]. It runs TextRank multiple times for a document, once for each of its topics induced by a Latent Dirichlet Allocation. By running TextRank once for each topic, TPR ensures that the extracted key-phrases cover the main topics of the document. The final score of a candidate is computed as the sum of its scores for each of the topics, weighted by the probability of that topic in that document.

Bipartite Link Prediction

In Figure 1 we see that our problem can be viewed as a bipartite graph. Therefore we can use link-prediction method based on bipartite graphs to determine where our phrase should link to. Only few works address the bipartite case, but here we present 2 approaches.

One method is a **supervised machine learning approach** applied to link prediction in bipartite (social) networks [1]. This is achieved by introducing new variations of topological attributes to measure the likelihood of two nodes to be connected. The approach used in [1] consists of expressing the link prediction problem as a two class discrimination problem. Classical supervised machine learning approaches can then be applied in order to learn prediction models.

Another approach is link prediction using **common neighbors and the local-community-paradigm** [2]. By defining a common neighbor index and local-community-paradigm for bipartite networks, they [2] are able to introduce the first node-neighborhood-based and LCP-based models for topological link prediction. By using the vector representation of a bipartite graph, they [2] predict the likelihood of links between nodes. For instance, if $A = a_1, a_2, a_3$ and $B = b_1, b_2$ are the nodes from the two classes A and B present in the bipartite network, a vector $b_1 = (0, 1, 0)$ indicates that node b_1 interacts with node a_2 only. Then, the missing links $b_1 \Leftrightarrow a_1$ and $b_1 \Leftrightarrow a_3$ can take likelihood scores based on how similar (according to a certain metric) the vector representations of a_1 and a_3 are to the a_2 vector, which is the node interacting with b_1 [2].

Procedure

In this Bachelor thesis we will analyze how authors set hyperlinks in articles. To do so we will proceed as follows:

1. **Describe Assumptions**
   We will describe the assumptions we made about how an author chooses to link articles based on the factors described in **Objectives**.

2. **Performance Ranking of Articles**
   Using Google Analytics data from the online magazine, we create a ranking for articles which have the highest influence on a long session duration. With this

information we can optimize our factor ***performance of the linked article***. We will use Python to analyze the data.

3. **Optimal Link Placement**

   We need to figure out how the authors set the hyperlinks in an article. In order to do that we'll look at the best performing articles to learn how to place well performing links. However, this will only help us with the ***distance between phrases*** and the ***number of phrases*** in an article.

   To determine which ***phrases we should select in an article***, we first have to extract relevant key-phrases. After extracting all relevant key-phrases, we will use the knowledge gained from the previous step to weight the extracted key-phrases and use the highest ranking ones as our phrases.

   In order to determine which ***article a phrase should link to***, we have to calculate similarities between articles, secondly look at the performance of the chosen articles and thirdly how ***relevant the linked article is to the key phrase***. The relevance of selected article in relation to the source article will be determined with the collected past data. We have to weight these 3 attributes in order of importance for optimal results.

4. **Discussing assumptions with findings**

   We will compare our findings with the previous assumptions we made.

References

[1] N. Benchettara, R. Kanawati, and C. Rouveirol. Supervised machine learning applied to link prediction in bipartite social networks. In *2010 International Conference on Advances in Social Networks Analysis and Mining*, pages 326–330, Aug 2010.

[2] Simone Daminelli, Josephine Maria Thomas, Claudio Durán, and Carlo Vittorio Cannistraci. Common neighbours and the local-community-paradigm for topological link prediction in bipartite networks. *New Journal of Physics*, 17(11):113037, nov 2015.

[3] Kazi Saidul Hasan and Vincent Ng. Automatic keyphrase extraction: A survey of the state of the art. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1262–1273, 2014.

[4] Zhiyuan Liu, Wenyi Huang, Yabin Zheng, and Maosong Sun. Automatic keyphrase extraction via topic decomposition. In *Proceedings of the 2010 conference on empirical methods in natural language processing*, pages 366–376. Association for Computational Linguistics, 2010.

[5] Zhiyuan Liu, Peng Li, Yabin Zheng, and Maosong Sun. Clustering to find exemplar terms for keyphrase extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 257–266. Association for Computational Linguistics, 2009.

[6] Rada Mihalcea and Paul Tarau. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, 2004.