Super Awesome Wikipedia Hops Fairness Article

Blandine Seznec, Philip Thruesen, Jaroslav Cechak, and Roel Castano

{bsezne16, pthrue15, jcheca16, rcasta15}@student.aau.dk

May 24, 2016

Abstract

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

This paper explores the user behaviour in

1. Introduction

As technology becomes part of all aspects of human nature and new trends such as the internet of things connect everyday objects to the internet, the amount of data stored in the digital universe has been growing at an outstanding rate. Consequently, processing and analysing data sets has become increasingly difficult and techniques have been developed in the fields of machine learning, big data, and others to facilitate the use of data.

One of the many problems that arises from this growth of digital capacity is information retrieval. We can see examples of this with search engines, recommender systems, bioinformatics, and many more, where it is necessary to rank data sets based on relevance to a certain query taking into account multiple features which influence the relevance of each option. The Learning to Rank algorithm is design to solve this issue and extensive research has been done to improve it in the last decade.

One interesting case where Learning to Rank (L2R) could be applied is to extract the importance of links in Wikipedia articles to prevent Overlinking; Containing excessive number of links, making it difficult to likely to aid the user [1]. Wikipedia is an internet encyclopaedia with more than 38 million articles in over 250 different languages of semi-structured information. It is also the most popular wiki-based website, and is ranked by Alexa as the #6 most popular website in the internet. It allows the collaborative modification of its articles by the users, which is one of the main reasons Wikipedia has grown to such an impressive size.

Being "the free encyclopedia that anyone can edit" has many advantages and disadvantages, and one of the disadvantages is, as mentioned before, Overlinking. As mentioned in a study by Ashwin Paranjape et al. [2]: "in the English Wikipedia, of all the 800,000 links added ... in February 2015, the majority (66%) were not clicked even a single time in March 2015, and among the rest, most links were clicked only very rarely". Since most of the editing of articles is done manually, finding and removing useless links to other articles is hard to do and editors do not focus on this. Given this case, we would like answer the following questions.

- Is it possible reduce the amount of links in Wikipedia articles by ranking the most relevant links based on a set of the most important features?
- How could the features for the L2R algorithm be chosen for maximum effectiveness?
- Will reducing the number of links in a Wikipedia article improve the readability of an article and aid the reader in finding interesting and relevant links?

1.1. Related Work

Previous work has been done with link structure in Wikipedia from which we acquired key concepts for this article. [3] explores the issue of disambiguation and detection of possible links in external texts. In addition to the main focus in automatic cross-reference of external articles, this paper provides an understanding of certain techniques used to detect potential links in articles and the proper reference (disambiguation) of terms in wikipedia. Detecting links relies in a machine learning algorithm similar to the one applied in this article which weights in different features of the articles to provide a score to all potential links and chose the final ones. Examples of features used in this implementation include link probability, relatedness of topics, disambiguation confidence, and many others. Learning to disambiguate links on the other hand, means identifying the correct meaning, for example "crane", as a large bird or a mechanical lifting machine, depending on the context (using unambiguous concepts for example) and probability of said word.
On the other hand, [4] adds on the topic of identifying missing hyperlinks by utilising data sets of navigation paths from wikipedia-based games in which users find paths between articles. This is useful for studying shortest paths between target articles but we believe this does not address the issue of aiding users by presenting them with articles that might be interesting.
[2] comes closer to the goal of pointing out missing references by making use of server logs to weight the usefulness of links that are not yet implemented. By studying the user's paths through Wikipedia they find patterns which could be shortened with missing links.

2. Method

2.1. Learning to Rank Algorithms

In this work, the aim is to look at possibility to rank the links between articles on the Wikipedia by their real click-trough rate. Ranking is an essential part of informational retrieval, but it is not limited to it. Ranking became even more important in recent years when there is much more data than one can process in reasonable amount of time. In order to accomplish this task, we chose to use a family of algorithms called Learning to Rank. These algorithms brings machine learning approach into information retrieval.

2.1.1 Ranking problem formulation

Let q be a query and let's denote an associated set of documents to the query q as $\mathbf{x} = \{x_1, x_2, \ldots, x_m\}$. Every document x_i has its label (relevance evaluation) y_i in the set $\mathbf{y} = \{y_i\}_{i=1}^{m}$. The values of y_i has to be from some totally ordered set $(S, \leq)$. Ranking can be view as a task of finding permutation π on indices $\{1, 2, \ldots, m\}$ given a query q and its associated set of documents $\mathbf{x}$. Permutation π must satisfy that $y_{\pi(j)} \leq y_{\pi(i)}$ for all $1 \leq i < j \leq m$. The sequence $x_{\pi(1)}, x_{\pi(2)}, \ldots, x_{\pi(m)}$ is then ordering of retrieved documents according to their relevance in respect to the query q.

Learning to rank algorithms handles this task as a instance of supervised machine learning problem. Each document is represented by its feature vector. Let q_i where $1 \leq i \leq n$ be the training queries and $\mathbf{x}^{(i)} = \{x_j^{(i)}\}_{j=1}^{m^{(i)}}$ their associated documents, where $m^{(i)}$ is the number of associated documents for the query q_i. Then $\mathbf{y}^{(i)} = \{y_j^{(i)}\}_{j=1}^{m^{(i)}}$ are labels for the associated documents to query q_i (also called ground truth). Test set is $\mathbf{T} = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^{n}$ for the training queries. The algorithm then automatically learns a model for $\mathbf{T}$ in the form of a function $F(\mathbf{x}^{(i)})$ that approximates the real mapping $\hat{F}(x^{(i)}) = y^{(i)}$ on the training set. Such model can later be used to predict relevance of new query instances outside the training set, where the label is unknown.

The common idea behind creating the model in learning to rank algorithms, or machine learning in general, is optimisation of a loss function $L(F(\mathbf{x}^{(i)}), \mathbf{y}^{(i)})$ for $1 \leq i \leq n$. The loss function describes the quality of found ranking in terms of errors made compared to ground truth. In [5] and [6], loss function is used to categorise learning to rank algorithms into three groups; pointwise, pairwise, and listwise.

2.1.2 Pointwise approach

The pointwise algorithms treat each document (feature vector to be precise) as an standalone instance. The input for the resulting model is a feature vector and its output is predicted label. Essentially the problem is simplified to regression, classification, or ordinal regression as each document is treated independently as a point in the feature space. Loss functions corresponding regression, classification, or ordinal loss functions. Limitation pointed out in [5] is assumption that relevance is absolute and does not depend on the query.

2.1.3 Pairwise approach

The pairwise algorithms always look at a pair of documents. The input for the model is a pair of feature vectors and the output is their relative preference, i.e. if the first from the pair should be ranked higher that the second one or vice versa. The loss function in this approach measures discrepancy between preference predicted by the model and actual order in the ground truth.

2.1.4 Listwise approach

The listwise algorithms look at whole document list as a whole. The input for the model is list of feature vectors and the output is either list of labels or a permutation. This approach is the only one where the loss function directly measures the final position/rank of documents.

2.1.5 RankLib

In this work a library called RankLib[1] (a part of The Lemur project [7]) has been used for performing learning to rank. This library implements the following algorithms as stated in [8].

- MART [9] (pointwise)
- RankNet [10] (pairwise)
- RankBoost [11] (pairwise)
- AdaRank [12] (listwise)
- Coordinate Ascent [13] (listwise)
- LambdaMART [14] (listwise)
- ListNet [15] (listwise)
- Random Forests [16]

2.2. Data Sets

There are multiple data sets and sources available which facilitate the interaction with Wikipedia, and due to the massive amount of data managed by wikipedia, it is important to have an organised and manageable representation of this data in our own server for processing. This section describes the data sources used during the research of this article.

2.2.1 Wikipedia Dumps

The most significant data source is the English Wikipedia database dump, which is released at least once a month, and contains a complete copy of the text and metadata of current revisions of all articles in XML format. This dump facilitates the extraction of mu Wikipedia also releases the page views and page counts for all articles.

2.2.2 Wikipedia Clickstream

The Wikipedia Clickstream [17] project contains data sets of $(referer, resource)$ pairs of articles describing user navigation and raw counts on the volume of traffic through the article pairs. This pairs are extracted from the request logs of Wikipedia, in which the referer is an HTTP header field that identifies the webpage from which the resource was requested. Typical referral sites like Google or Facebook are also included and crawler-traffic has been attempted filtered from the raw data.

2.3. Ground Truth

James Kobielus [18], from IBM Big Data and Analytics Hub, describes ground truth as "a golden standard to which the learning algorithm needs to adapt". In most cases, a training data set labeled by human experts is needed to provide data patterns for the learning algorithm to use as baseline. This type of machine learning is referred to as supervised learning. The other two main approaches, unsupervised learning and reinforcement learning, attempt to automate the distillation of knowledge from data not previously labeled by human experts.

In our case we apply a supervised learning model using a ground truth dataset constructed primarily from the clickstream data query logs. We use a subset of 1000 articles having the most outgoing clicks. We refer to these articles as being *prominent* articles. Each of these prominent articles contain links to other articles so that we in total have [2] 186k uniquely referenced articles and 343k article (α, β) pairs where α is a prominent article and β is a *prominent-linked* article.

The clickstream data does not contain pairs where the the number of referrals from α to β is below 10 and therefore we supplement our ground truth data using the

[1] RankLib source and binary files are available in Sourceforge repository at https://sourceforge.net/p/lemur/wiki/RankLib/

[2] FiXme Note: Insert correct numbers

Wikipedia article dumps for the same period of time as the recording of clickstream data. In this way we can extract article pairs that does not appear in the clickstream data because of the 10-click limit (set to protect privacy) – this adds a substantial amount of "unpopular" article links to our set so that we avoid a bias towards more clicked links. [3]

2.4. Notation

We denote our set of prominent articles[4] $A = \{\alpha_1, ..., \alpha_M\}$ where α_i is one of M prominent articles. Belonging to the i^{th} prominent article is a set B_i of linked articles: $B_i = \{\beta_1, ..., \beta_{N_i}\}$ where N_i is the number of referenced articles for the i^{th} prominant article.

Click counts (popularity) for an article pair (α, β) is denoted $pop(\alpha, \beta)$. Eq. 1 defines the overall popularity of an arbitrary prominent article α_k.

$$pop(\alpha_k) = \sum_{j=1}^{|B_k|} pop(\alpha_k, \beta_j) \tag{1}$$

Eq. 2 defines the overall popularity of a prominent-linked article.

$$pop(\beta) = \sum_{i=1}^{|A|} pop(\alpha_i, \beta) \tag{2}$$

In our set of ground truth we have that:

$$rank(\alpha, \beta) = pop(\alpha, \beta) \tag{3}$$

where a higher rank equals the level of popularity and a high rank therefore represents an important article pair.

This notation will be used throughout the rest of this article. [5]

3. Features

This section defines each feature used in the implementation of Learn to Rank algorithm.

3.1. Popularity of Resource

It seems reasonable to assume that users have a tendency to click on links to articles that are describing trending and popular topics. This is the motivation for constructing the feature described in details below where a numerical measure is estimating a degree of user interest for the resource article β in the pair (α, β).

For our model to be applicable to new articles, we should not, in general, use any information on article α that are caused by user behaviour. If we were to use this information we would not in practice be able to apply our model to new articles before sufficient data has been logged.

3.1.1 Description

From the clickstream data it is possible to derive the number of article views on a an article that are as a result of external traffic i.e. traffic coming from outside the Wikipedia domain. This gives the possibility of extracting the number of searches from known search providers: Google, Bing and Yahoo! or the number of visits from social media sites such as Twitter and Facebook. This is traced using the HTTP header information: 'referer'. Note that this information is completely separated from the information used for creating the ground truth – for that only internal click counts were used.

Furthermore we limit ourselves to using traffic data from external and well known sources and not include data from internal Wikipedia searches or when the referrer is missing: This could possibly be due to spoofed referral headers which could introduce a bias to our model.

We consider traffic from search and social media individually as [19] states that users might have different intentions depending on which media they choose to search or browse for content. Users on social media, unlike for

[3] FiXme Note: INCOMPLETE
[4] FiXme Note: A doenst show in pdf
[5] FiXme Note: INCOMPLETE

search engines, tends to look for content that is currently trending.

3.1.2 Extraction

Social networks are known for virally spreading topics that in terms of popularity or 'shares' will follow a power law distribution [20] – this is a consequence of the social network forming scale-free graphs. Assuming interest in articles from both search- and social media sites have emerged from social networks we consider that our distribution of article views from these external sources are as such.

The nature of the power law distribution gives that there are very few popular articles among all the articles. And of those few only a small fraction are highly popular.

To illustrate that our above assumptions for both search- and social media traffic holds figure [6] shows how the probability density function of article views originating from search are linear in a log-log plotting which is characteristic for power law distributions.

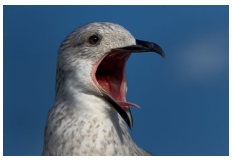

Figure 1: *figure caption*

. The below figure illustrates this. The articles considered is all distinct ðIŻČ-articles for all ÉŚ in A, i.e. all distinct prominent-linked articles.

We see that the popularity originating from search traffic of these mentioned articles follow a power law distribution.

Knowing the distribution of the click data from social medias and search we can preprocess the features including traffic volumes from these sources. An easy and effective way of coping with the non-linearity of the power law distribution is by taking the logarithm twice of the popularity ln(ln(beta)) and thereafter normalizing the feature values using the maximum value. The final outcome of this processing results in the feature having a correlation coefficient of $r = 0,3879$ and a 'coefficient of determination' of $r^2 = 0,15$.

In the below figures the PDF is shown for the search volumes respectively without taking the log, with the log and with the log twice. The distribution of feature values 'flattens out' and signs of any exponential growth or decay disappears.

For the above 3 tests we have a 'coefficient of determination' of respectively 0,0045, 0,113 and 0,150, thereby effectively selecting the 3. option as an input feature for the model.

It must be noted for the cases where the popularity are zero, we simply skip preprocessing because of the nature of the logarithm.

3.2. Popularity Percentile Rank

[7]

Another feature to derive from clickstream data is for an instance pair $(, beta)$ where beta is in B the probability that an arbitrary article inB $beta$ has a lower popularity than beta: $f((, beta)) = pr(pop(beta) > pop())$

Characteristic for this feature is that if $pop(beta)$ is the maximal value for any article in B, then the feature value gives: $f((, beta_max)) = 1$

If $pop(beta)$ is the minimum value for any article in B, then the feature value gives: $f((, beta_min)) = 0$

And if $pop(beta)$ is equal to the median value of the articles in B, where all article popularity values are distinct, then the feature value gives: $f((, beta_med)) = 0.5$

This feature gives an opportunity to extract more knowledge on the distribution of article popularity locally for a prominent article.

The coefficient of determination yields $r^2 = 0,0582$.

3.2.1 Description

x

3.2.2 Motivation

x

3.2.3 Extraction

x

[6] FiXme Note: insert figure

[7] FiXme Note: the following is just raw notes

3.3. Title similarity

[8] A last feature to extract from the clickstream data is title similarity. This feature is fairly trivial and motivated by e.g. the observation that articles on people often have their family members among the most popular reference links and these often share familyname whereas we would observe a higher title similarity. The feature was implemented using a Jaccard similarity measure and normalized from 0 to 1. We achieved a coefficient of determination on $r^2 = 0,0115$.
[9]

3.4. Link Position

3.4.1 Description

Position of a link in an article is the number of characters counted from the beginning of the article up to the link appearance. Because the length of different articles may vary quite substantially, this number is normalized per article.

3.4.2 Motivation

Our intuition behind the Link Position feature is based on two observations. The first one is that given the way Wikipedia articles are structured, the most general description of the article is placed in the first few paragraphs before the table of contents. This section of the Wikipedia article is called the lead [21]. As described in the Wikipedia manual of style, for many people, the lead section is the only section they will read since it summarises the entire article. Later paragraphs only dive deeper into the topics outlined in the description.

The second observation is the way people read web pages. As explained in [22] by Jakob Nielsen, one of the leaders in human-computer interaction, on average, users have time to read at most 28 % of the words on a website. Additionally, most attention is given to the top portion of a page and later sections are merely skimmed through. People looking for different article that is somewhat related to the one currently being read might be interested in more general concepts as they contain the searched term. As explained above, more general terms happen to be heavily abundant in the top portion of a Wikipedia article.

3.4.3 Extraction

Raw texts of the articles in Wiki markup are taken from the Wikipedia dumps described above and links to other articles are found using regular expressions. Links to images, external sources, etc. are ignored. Due to the nature of source, from which is this feature extracted, there might be slight discrepancies in position value and the exact number of characters preceding the links in final article as displays in web browser.

3.5. Link Order

3.5.1 Description

This feature captures the position of link relative to all the other links contained in the same article. The order equal to n means that the link is the n^{th} link in the article. The final value of this feature is again normalised per article.

3.5.2 Motivation

Similar to the previous feature, Link Order is based on the observation that a Wikipedia article's initially describe the topic in general terms. While the link position captures more of a distance between links and their spread, link order is a simplified version of it. It conveys less information, but in a much more straightforward manner.

3.5.3 Extraction

This feature is also extracted from the Wikipedia dumps in a similar manner to the Link Position feature.

3.6. Link Count

3.6.1 Description

Link Count is the number of occurrences of the same link throughout the article.

3.6.2 Motivation

Whenever a link is present multiple times in a single article, it means that the topic is repeated several times in the article as well. This may signify strong relatedness of articles content.

[8] FiXme Note: Just raw notes - and it should be together with the other sim features
[9] FiXme Note: INCOMPLETE

3.6.3 Extraction

The feature is extracted from the Wikipedia dump using the raw article texts as in case of the previous two features.

3.7. Community Membership

3.7.1 Description

Let $G(V, E)$ be a simple graph. Then a community in G is a subgraph $(V', E') = G' \subseteq G$, such that $|E'| \geq |\{\{u, v\} \in E \setminus E' \mid \{u, v\} \cap V' \neq \emptyset\}|$. In our case the G is a graph of Wikipedia, where $V = \bigcup_{i=1}^{M} B_i \cup A$ is a set of articles and E is set of links between them. After detecting all communities, each edge (link) $\{\alpha_i, \beta_j\}$ is given a value of 1 if both α_i and β_j belong to the same community or -1 otherwise.

3.7.2 Motivation

Communities are an implicit clustering of articles on Wikipedia that capture emerging properties of interconnected articles. This feature looks promising in cases where someone is interested in a specific topic, be it music bands of the same genre or fields of study in a certain science. Related articles from the same community might be a good place to look at and will highly likely contain desired article. Furthermore, users reading an article have shown an interest in specific topic and might want to broaden and deepen his or her knowledge of it.

3.7.3 Extraction

Every link extracted from the Wikipedia dump is converted into an edge of a graph. The resulting graph is then loaded and communities are found using igraph package in R. Community detection algorithms used were [23], [24], and [25]. The graph was transformed into a simple graph and then community detection algorithms were run. Even though links have natural orientation, one can relax them as undirected edges. Sheer number of links by it self is an adequate indicator of community belonging regardless of their orientation.

3.8. Symmetric Linking

3.8.1 Description

A link (α_i, β_j) is considered symmetric if, and only if $\beta_j \in B_i$ and link (β_j, α_i) also exists. If the link is symmetric, the feature has a value of 1 and -1 otherwise.

3.8.2 Motivation

Symmetric linking indicates, in some cases, that there exists an important relevance between said articles or highly related topics are being discussed. Examples of this article relationship includes competing presidential candidates, sports team rivals, movies and its actors, etc. As expected, it is common for users to demonstrate interest in these kind of relations between articles.

3.8.3 Extraction

This feature is extracted from the Wikipedia dump file. First, all the links between prominent articles in A and articles in all B_i for $1 \leq i \leq M$ are found. Then for every link (α_i, β_j) a link (β_j, α_i) is looked up.

3.9. HITS and PageRanks

3.9.1 Description

HITS and PageRank characterise nodes in the graph by score based on number of edges and also the scores of neighbours.

Although HITS and PageRank differ in content their main goal is the same, give estimate of article significance. In case of HITS, high hub score may indicate more general topics while high authority score would be indication of very focused article discussing particular term in great depth. PageRank is similar to the hub and authority score combined.

3.9.2 Motivation

The intuition behind this feature comes from the search engine domain. When searching, these scores helps identify the most relevant results. When applied to articles, the scores will help categorise linked articles for the learning to rank algorithm.

3.9.3 Extraction

Similarly to community extraction, HITS and PageRank are computed from graph representation of Wikipedia. The graph is processed in R using package igraph that implements both [26] and [27]. Link directions are preserved in this case.

4. Results

Table 1: *Example table*

Name		
First name	Last Name	Grade
John	Doe	7.5
Richard	Miles	2

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

$$e = mc^2 \tag{4}$$

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

5. Discussion

5.1. Subsection One

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

5.2. Subsection Two

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Donec odio elit, dictum in, hendrerit sit amet, egestas sed, leo. Praesent feugiat sapien aliquet odio. Integer vitae justo. Aliquam vestibulum fringilla lorem. Sed neque lectus, consectetuer at, consectetuer sed, eleifend ac, lectus. Nulla facilisi. Pellentesque eget lectus. Proin eu metus. Sed porttitor. In hac habitasse platea dictumst. Suspendisse eu lectus. Ut mi mi, lacinia sit amet, placerat et, mollis vitae, dui. Sed ante tellus, tristique ut, iaculis eu, malesuada ac, dui. Mauris nibh leo, facilisis non, adipiscing quis, ultrices a, dui.

List of Corrections

References

[1] John C. Dvorak. *Missing Links*. PCMag. Apr. 16, 2002. URL: http://www.pcmag.com/article2/0,2817,33326,00.asp (visited on May 23, 2016).

[2] Ashwin Paranjape et al. "Improving Website Hyperlink Structure Using Server Logs". In: *CoRR* abs/1512.07258 (2015). URL: http://arxiv.org/abs/1512.07258.

[3] David Milne and Ian H. Witten. "Learning to Link with Wikipedia". In: *Proceedings of the 17th ACM Conference on Information and Knowledge anagement*. CIKM '08. Napa Valley, California, USA: ACM, 2008, pp. 509–518. ISBN: 978-1-59593-991-3. DOI: 10.1145/1458082.1458150. URL: http://doi.acm.org/10.1145/1458082.1458150.

[4] Robert West, Ashwin Paranjape, and Jure Leskovec. "Mining Missing Hyperlinks from Human Navigation Traces: A Case Study of Wikipedia". In: *CoRR* abs/1503.04208 (2015). URL: http://arxiv.org/abs/1503.04208.

[5] Tie-Yan Liu. "Learning to Rank for Information Retrieval". In: *Foundations and Trends® in Information Retrieval* 3.3 (2009), pp. 225–331. ISSN: 1554-0669. DOI: 10.1561/1500000016. URL: http://dx.doi.org/10.1561/1500000016.

[6] Hang Li. "A Short Introduction to Learning to Rank". In: *IEICE Transactions on Information and Systems IEICE Trans. Inf. & Syst* E94-D.10 (Oct. 2011), pp. 1854–1862. ISSN: 1745-1361. DOI: 10.1587/transinf.E94.D.1854. URL: https://search.ieice.org/bin/pdf_link.php?category=D&lang=E&year=2011&fname=e94-d_10_1854&abst=.

[7] The Lemur Project. *The Lemur Project*. Center for Intelligent Information Retrieval (CIIR) at the University of Massachusetts, Amherst, and the Language Technologies Institute (LTI) at Carnegie Mellon University. Dec. 19, 2013. URL: http://www.lemurproject.org/ (visited on May 19, 2016).

[8] Van Dang. *RankLib*. Oct. 5, 2013. URL: https://sourceforge.net/p/lemur/wiki/RankLib/ (visited on May 19, 2016).

[9] Jerome H. Friedman. "Greedy function approximation: A gradient boosting machine." In: *Ann. Statist.* 29.5 (Oct. 2001), pp. 1189–1232. DOI: 10.1214/aos/1013203451. URL: http://dx.doi.org/10.1214/aos/1013203451.

[10] Chris Burges et al. "Learning to Rank Using Gradient Descent". In: *Proceedings of the 22Nd International Conference on Machine Learning*. ICML '05. Bonn, Germany: ACM, 2005, pp. 89–96. ISBN: 1-59593-180-5. DOI: 10.1145/1102351.1102363. URL: http://doi.acm.org/10.1145/1102351.1102363.

[11] Yoav Freund et al. "An Efficient Boosting Algorithm for Combining Preferences". In: *J. Mach. Learn. Res.* 4 (Dec. 2003), pp. 933–969. ISSN: 1532-4435. URL: http://dl.acm.org/citation.cfm?id=945365.964285.

[12] Jun Xu and Hang Li. "AdaRank: A Boosting Algorithm for Information Retrieval". In: *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '07. Amsterdam, The Netherlands: ACM, 2007, pp. 391–398. ISBN: 978-1-59593-597-7. DOI: 10.1145/1277741.1277809. URL: http://doi.acm.org/10.1145/1277741.1277809.

[13] Donald Metzler and W. Bruce Croft. "Linear Feature-based Models for Information Retrieval". In: *Inf. Retr.* 10.3 (June 2007), pp. 257–274. ISSN: 1386-4564. DOI: 10.1007/s10791-006-9019-z. URL: http://dx.doi.org/10.1007/s10791-006-9019-z.

[14] Qiang Wu et al. "Adapting Boosting for Information Retrieval Measures". In: *Inf. Retr.* 13.3 (June 2010), pp. 254–270. ISSN: 1386-4564. DOI: 10.1007/s10791-009-9112-1. URL: http://dx.doi.org/10.1007/s10791-009-9112-1.

[15] Zhe Cao et al. "Learning to Rank: From Pairwise Approach to Listwise Approach". In: *Proceedings of the 24th International Conference on Machine Learning*. ICML '07. Corvalis, Oregon, USA: ACM, 2007,

pp. 129–136. ISBN: 978-1-59593-793-3. DOI: 10.1145/1273496.1273513. URL: http://doi.acm.org/10.1145/1273496.1273513.

[16] Leo Breiman. "Random Forests". In: *Machine Learning* 45.1 (2001), pp. 5–32. ISSN: 1573-0565. DOI: 10.1023/A:1010933404324. URL: http://dx.doi.org/10.1023/A:1010933404324.

[17] Ellery Wulczyn. *Wikipedia Clickstreap*. Datahub. Apr. 7, 2016. URL: https://datahub.io/dataset/wikipedia-clickstream (visited on May 19, 2016).

[18] James Kobielus. *The Ground Truth in Agile Machine Learning*. IBM Big Data and Analytics Hub. June 13, 2014. URL: http://www.ibmbigdatahub.com/blog/ground-truth-agile-machine-learning (visited on May 19, 2016).

[19] Jaime Teevan, Daniel Ramage, and Merredith Ringel Morris. # *TwitterSearch: a comparison of microblog search and web search*. 2011, pp. 35–44. URL: http://research.microsoft.com/pubs/154556/wsdm11-twitter.pdf.

[20] Puneet Jain et al. "Scalable Social Analytics for Live Viral Event Prediction." In: (2014). URL: http://www.aaai.org/ocs/index.php/ICWSM/ICWSM14/paper/viewFile/8035/8123.

[21] Wikipedia. *Wikipedia:Manual of Style/Lead section*. May 17, 2016. URL: http://en.wikipedia.org/w/index.php?title=Plagiarism&oldid=5139350 (visited on May 18, 2016).

[22] Jakob Nielsen. *How Little Do Users Read?* Nielsen Norman Group. URL: https://www.nngroup.com/articles/how-little-do-users-read/ (visited on May 18, 2016).

[23] Aaron Clauset, M. E. J. Newman, and Cristopher Moore. "Finding community structure in very large networks". In: *Phys. Rev. E* 70 (6 Dec. 2004), p. 066111. DOI: 10.1103/PhysRevE.70.066111. URL: http://link.aps.org/doi/10.1103/PhysRevE.70.066111.

[24] Usha Nandini Raghavan, Réka Albert, and Soundar Kumara. "Near linear time algorithm to detect community structures in large-scale networks". In: *Phys. Rev. E* 76 (3 Sept. 2007), p. 036106. DOI: 10.1103/PhysRevE.76.036106. URL: http://link.aps.org/doi/10.1103/PhysRevE.76.036106.

[25] M. Rosvall, D. Axelsson, and T. C. Bergstrom. "The map equation". In: *The European Physical Journal Special Topics* 178.1 (2009), pp. 13–23. ISSN: 1951-6401. DOI: 10.1140/epjst/e2010-01179-1. URL: http://dx.doi.org/10.1140/epjst/e2010-01179-1.

[26] Sergey Brin and Lawrence Page. "The anatomy of a large-scale hypertextual Web search engine". In: *Computer Networks and {ISDN} Systems* 30.1–7 (1998). Proceedings of the Seventh International World Wide Web Conference, pp. 107–117. ISSN: 0169-7552. DOI: 10.1016/S0169-7552(98)00110-X. URL: http://www.sciencedirect.com/science/article/pii/S016975529800110X.

[27] Jon M. Kleinberg. "Authoritative Sources in a Hyperlinked Environment". In: *J. ACM* 46.5 (Sept. 1999), pp. 604–632. ISSN: 0004-5411. DOI: 10.1145/324133.324140. URL: http://doi.acm.org/10.1145/324133.324140.