Super Awesome Wikipedia Hops Fairness Article

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

{bsezne16, pthrue16, jcheca16, rcasta15}@student.aau.dk

May 19, 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

Wikipedia is an internet encyclopedia with more than 38 million articles in over 250 different languages of semi-structured information (text and multimedia). Even with its advanced searching technology developed and improved over the last 15 years, the sheer amount of information may be overwhelming to most people and confusing at times. Seeking the relevant (related) information for a certain topic or interest among all these articles can prove cumbersome. Currently, most Wikipedia articles provide a "See also" section with some recommendations of related articles. These recommendations are added manually by the contributors. This is, of course, time consuming and error-prone, as the human contributors' recommendations might not be complete nor reflect what readers find relevant.

By utilising the available data, which include Wikipedia Clickstream data sets, specific per second/day page view count, and derived data from article structures, we believe we could improve the "See Also" section to provide unbiased, accurate recommendations.

This has been researched multiple times but we have found some of them to be based almost completely on data from Wikispeedia or The Wiki Game, which both supply paths utilised by players to reach certain, unrelated articles. This does affect how the user navigated Wikipedia and doesn't reflect natural user behaviour.

Our method for this research is based on real user behaviour and Wikipedia data. Using the Learning to Rank algorithm, it is possible to weight the importance of all of the different indicators and recommend the most useful articles for the user. By solving this problem, we would like to get insight into the following questions.

- Is it possible to predict the user's upcoming path utilising generalised information provided by Wikipedia and applied to features in the Learning to Rank algorithm?
- How could this predictions improve the suggestions to users of the Wikipedia network to facilitate finding the most interesting and relevant articles?

1.1. Related Work

Previous work has been done with link structure in Wikipedia from which we acquired key concepts for this article. [1] 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, [2] 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.

[3] 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 [4] and [5], 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 [4] 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 [6]) has been used for performing learning to rank. This library implements the following algorithms as stated in [7].

- MART [8] (pointwise)
- RankNet [9] (pairwise)
- RankBoost [10] (pairwise)
- AdaRank [11] (listwise)
- Coordinate Ascent [12] (listwise)
- LambdaMART [13] (listwise)
- ListNet [14] (listwise)
- Random Forests [15]

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 [16] 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 [17], 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.

For this case, we built the ground truth for our model primarily using the Clickstream data. We use a subset of articles consisting of the 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 our ground truth dataset contains 186k uniquely referenced articles and 343k article (A, B) pairs where A is a prominent article and B is a prominent-linked article.

The clickstream data does not contain pairs were the navigational traffic is below 10 referrals and therefore we use the Wikipedia article dump for the same period of time where we can extract links that does not appear in the clickstream data. A large fraction of existing links are having less than 10 clicks.

INCOMPLETE

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

3. Features

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

3.1. Link Position

Our intuition behind the Link Position feature is based on two reasons. 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 [18]. 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 motivation is the way people read web pages. As explained by Jakob Nielsen [19], one of the leaders in human-computer interaction, on average, users have time to read at most 28% of the words of 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 first portion of Wikipedia article.

To be able to measure this feature, we count the number of characters preceding the occurrence of the link. The text of the articles is taken from the wikipedia dumps previously described and links to other wikipedia articles are found using regular expressions. As expected, links to images, external sources, etc. are ignored. Due to the way this feature is extracted, there might be slight discrepancies in feature value and the exact number of characters preceding the links.

3.2. Link Order

Similar to the previous feature, Link Order is based on the fact that a wikipedia article's initially describe the topic in general terms and the hypothesis?? that the probability of clicking a link is higher the closer it is to the first term. 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. This feature is also extracted from the Wikipedia dumps by counting the number of links in the article. This feature captures the position of link relative to all the other links contained in the same article. The value n means that the link is the n^{th} link in the article.

3.3. Community Membership

This feature captures notion of two article being in the same community. The communities are computed from the graph representation of Wikipedia $G(V, E)$, where V is a set of articles and E is set of links between them. In this scenario community is $(V', E') = G' \subseteq G$ such that $|E'| \geq |\{\{u, v\} \in E \setminus E' \quad | \quad u \in V' \vee v \in V'\}|$. Communities are a implicit way of clustering articles on Wikipedia that capture emerging properties of interconnected articles.

This feature looks promising in cases where someone is interested in a specific topic. This could mean, as mentioned earlier, bands of a music genre, states of a country, fields of study in a certain science, or many others. 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.

3.4. Symmetric Linking

This feature captures the notion of two article being interconnected in both directions. Formally, a link (A, B) between article A and B is symmetric if and only if link (B, A) also exists. 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. By looking and the article relationships, we found in one sample of the most visited articles that 74.9% of the links were non-symmetric and 25.1% were symmetric.

3.5. HITS and PageRanks

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.

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 scores.

The intuition behind this feature comes from the search engine domain. In search, 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.

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.

$$e = mc^2 \tag{1}$$

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.

References

[1] 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.

[2] 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.

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

[4] 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.

[5] 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=.

[6] 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).

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

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

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

[17] 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).

[18] 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).

[19] Jakob Nielsen. *How Users Read on the Web*. Nielsen Norman Group. Nov. 1, 1997. URL: https://www.nngroup.com/articles/how-users-read-on-the-web/ (visited on May 18, 2016).