LLAMA-B: Automatic Hyperlink Authoring in the Blogosphere

Dong Zhou
University of Nottingham
Nottingham
United Kingdom
dxz@cs.nott.ac.uk

Mark Truran
University of Teesside
Middlesbrough
United Kingdom
m.a.truran@tees.ac.uk

Tim Brailsford
University of Nottingham
Nottingham
United Kingdom
tjb@cs.nott.ac.uk

Helen Ashman
University of South Australia
Adelaide
Australia
helen.ashman@unisa.edu.au

Amir Pourabdollah
University of Nottingham
Nottingham
United Kingdom
axp@cs.nott.ac.uk

ABSTRACT

Viewed collectively, the sum of all blog entries recorded to date (usually referred to as the *blogosphere*) represents a prodigiously rich collection of commentary and opinion, a dizzying mixture of fact and speculation, subjective opinion and objective data. This paper introduces a hypermedia authoring tool intended to simplify the process of navigating this chaotic environment. The tool works by adding additional hyperlinks to blogs, links which connect blog entries addressing similar topics. These hyperlinks are generated by an algorithm that uses statistical language modeling and graph based analysis to exploit the implicit associative structure of the *blogosphere*. An evaluative exercise, centred upon the unsupervised labeling of blog articles, confirms the effectiveness of this approach.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing—*Linguistic processing*; I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*Language models*

General Terms

Algorithms, Measurement, Performance, Experimentation

Keywords

Hypertext generation, link generation, blogosphere, tagging.

1. INTRODUCTION

The rapid growth of the *blogosphere* has not escaped the attention of the hypermedia research community, and countless initiatives aimed at searching, mining and analyzing this idiosyncratic information space have been developed [1, 7, 11, 8]. However, there is one important area that has so far escaped widespread scrutiny - this is the potential application of link authoring tools to the *blogosphere*.

The development of link authoring tools effectively predates the invention of the blog by at least two decades [13]. Historically, their existence can be seen as a response to the various shortcomings of manually authored hypertext/hypermedia links [2]. Combining fallibility with finite attention spans, human beings will frequently manufacture broken hyperlinks, and tend to design sets of links that perform poorly in terms of completeness [12]. Various tools engineered over the last twenty years have attempted to shore up this uncertain authoring process with sound algorithmic alternatives, but the problems inherent in manual link creation persist, and are readily apparent in even very modern forms of hypermedia.

With this in mind, we have recently developed a software tool capable of autonomously authoring hyperlinks which associate related blog articles. We call this application LLAMA-B (Linking LAnguage Models Algorithm in the Blogosphere). This system is a direct descendant of an earlier prototype named LLAMA, which was presented in [16]. This paper introduces a number of modifications to the original system designed to improve linking performance. These refinements include an alternative authoring technique based on the PageRank algorithm [5], use of a local (rather than global) linking strategy, and a more sophisticated utilisation of language models [15].

2. RELATED WORK

Autonomous hyperlink authoring tools published to date have tended to occupy one of three categories [13]. The first type of system is the structural authoring tool [10]. Structural authoring tools exploit the internal logical structure of text to create links. Typically, this involves identifying significant structural elements in a text document (e.g. a title or a section heading), then linking these elements to another document. The second type of system is the statistical authoring tool [4, 16]. A statistical tool concerns itself with mathematical functions related to the frequency of terms in

HT'08, June 19-21, 2008, Pittsburgh, Pennsylvania, USA.
Copyright 2008 ACM 978-1-59593-985-2/08/06 ...$5.00.

the documents analysed. These functions are resolved into a determination of what each document is about, and this determination guides the link creation process. The third type of the system is the semantic authoring tool [14]. Semantic authoring tools address the conceptual meaning of terms found in the raw text. The construction of a semantic authoring tool typically involves the creation of formal representations of domain knowledge which can be used to inform the instantiation of hyperlinks.

Viewed collectively, publications addressing the topic of automatic hyperlink authoring form an extensive and mature body of theoretical work. However, despite this wealth of experience and technique, no concerted attempt has yet been made to make use of these tools in the *blogosphere*. Analytical work in this area has so far tended to concentrate on the formation of explicit (user-specified) links rather than implicit (algorithmically-defined) links [1, 7, 11, 8]. A strong argument in favour of developing our application was the opportunity to remedy this situation and study the performance of an authoring tool within this unusual and divergent information space.

3. DESCRIPTION OF LLAMA-B

LLAMA-B is a reiteration of the link authoring algorithm described in [16] combined with a number of recent improvements. In the following section, the core algorithm together with each of these improvements will be explained in detail. Our explanation will use the following notation:

t : *a single term*

d : *a document, representing a single blog entry text*

f : *a feature (a concept, a single term, multiple terms, a sentence, a paragraph in a document or even the document itself)*

C : *a collection of documents*

$LM_{f'}f$: *The language model scoring for a feature f that is induced from a feature $f^{'}$*

$Dom(f)$: *The Dominance score of a given feature f*

$Sub(f)$: *The Subordinance score of a given feature f*

The LLAMA-B algorithm

1: Input a collection of plain text blog entries C.

2: Extract every feature $f \in C$, creating links between any two possible features to form a weighted directed graph $G < V, E >$ where each V represents a feature f and each E represents a link between features with non-negative weighting $w(f_1 \longrightarrow f_2)$. Refer to [16] for a full description of the various weighting functions that can be applied at this point.

3: For each f, compute a *Dominance* score and *Subordinance* score. This gives you $Dom(f)$ and $Sub(f)$ for every single feature in the collection C.

4: Select a set of features with the strongest subordinate scores, $Strongest_{Sub}$, from the collection C as link anchors (which are constrained to *terms* here). This set can be formed in one of two ways - either by retaining a fixed number of the features with the highest $Sub(f)$ scores (as we have done in the experiments discussed in this paper), or by retaining only those features with a subordinance above a certain threshold.

5. For each of the subordinate features that were retained, $f \in Strongest_{Sub}$, determine a measure of linkage fitness $Fit(f, f^{'})$ between it and all possible target features (which are constrained to *documents* here). The fitness function we apply at this stage is defined in [16].

6: Generate the preferred link endpoint for each feature which is determined by the best fitness measure: $< f, f^{'} >$: $MaxFit(f, f^{'})$. It is worth noting that this will generate a single end link for each retained subordinate feature.

7. Collate and output the document collections C with link anchors and targets inserted.

3.1 MLE Smoothing

In the original LLAMA algorithm [16], we used a simple approach based on KL divergence that employed raw language models to compute the association between text features. However, MLE is generally known to underestimate the probabilities attached to a term that does not occur in a specific document (which will therefore have an MLE value of zero) [15]. In order to address this issue, the new LLAMA-B algorithm uses a strategy called *smoothing*, which assigns a non-zero probability to unseen words, thereby improving the accuracy of word probability estimation. Specifically, smoothing is adopted in the following way: given the MLE measure of term t over context d, a *Dirichlet-smoothed* estimate is defined as:

$$Dir_d t = \frac{c(t, d) + \mu \times MLE_C t}{\sum_{t'} c(t^{'}, d) + \mu} \tag{1}$$

Where μ is the smoothing parameter which controls the degree of reliance on relative frequencies in the corpus rather than on the counts in d (μ is usually set to 1000), and C is the document corpus.

3.2 PageRank Measure

The original LLAMA algorithm used an adaptation of Kleinbergs's work on hubs and authorities [9] to identify suitable link anchors and targets (referred to as *subordinate* and *dominant* text features respectively). LLAMA-B also implements the Kleinberg measure (*S-D*), but this is complemented by an alternative method for determining the end points of a link which is modeled on the PageRank algorithm [5]. When the *P-R* measure is applied, we calculate the *dominance* or *subordinance* scorings for all available text features in the following way:

For a given feature f, let $\{f\}_{IN}$ be a set of features that point to it and let $\{f\}_{OUT}$ be a set of features that f points to. Then, the *dominance* score of f is defined as follows:

$$\begin{aligned} Dom(f) &= (1 - \lambda) \times \frac{1}{|F|} \\ &+ \lambda \times \sum_{f^{'} \in \{f\}_{IN}} \frac{w(f^{'} \longrightarrow f)}{\sum_{f^{''} \in \{f^{'}\}_{OUT}} w(f^{'} \longrightarrow f^{''})} \\ &\times Dom(f^{'}) \end{aligned} \tag{2}$$

where λ is a dampening factor which integrates the probability of jumping from one text feature to another at random (normally set to 0.85) and $|F|$ is the total number of features in the collection. Starting with an arbitrary value assigned to every feature in the graph, this calculation is guaranteed to iterate until convergence below a certain threshold.

3.3 Global vs. Local Linking

A graph-based authoring algorithm can follow one of two general approaches. The first, which we refer to as the *global linking* approach, considers the document collection collectively, meaning that a single graph is built over the feature space which considers all of the available terms and documents simultaneously. The second approach, which we call *local linking*, iteratively constructs a unique graph for each document connecting it to the rest of the collection. In other words, a local linking algorithm will produce a number of graphs equal to the number of documents in the collection.

The original LLAMA algorithm followed the global linking approach. However, later evaluation of this algorithm revealed the potential for generating a hypertext containing 'dangling documents' (i.e. documents with no link anchors or endpoints at all, a microcosm of the Web's 'dark matter' [3]). Arguably, the existence of these dangling documents within a hypertext effectively *reduces* the quality of the overall navigational structure, since these documents represent informational dead ends, comparable to scientific articles containing no citations or references. The absence of anchors or endpoints also implies their omission from those search engine indices which are reliant on associative structures. One further criticism addresses the issue of scalability - a local linking approach is considerable more amenable to parallelisation than its global linking alternative, and is therefore more appropriate for larger document collections. For these reasons, the LLAMA-B algorithm implements a local linking approach, with α link anchors authored per article.

3.4 Title Weighting

In its original formulation, LLAMA had no understanding of document composition [16], and was incapable of exploiting structural elements relevant to the authoring process (e.g. document titles, sub-headings, footnotes etc.). This defect has been partially addressed in the new LLAMA-B algorithm, which integrates information extracted from blog titles when calculating the weighting between two features. The modified weighting function we apply is as follows:

$$w(f_1 \longrightarrow f_2) = LM_{f'}^{DT} f = (1-\delta) LM_{D(f')} f + LM_{T(f')} f \tag{3}$$

where $D(\bullet)$ indicates the content of the article, $T(\bullet)$ indicates the title of the article and δ is the parameter used to control the distribution of weights between the two components.

4. EVALUATION

This section describes an experiment designed to evaluate the LLAMA-B algorithm. Given the difficulty of assessing the intrinsic quality of an authored hypertext link [13], this assessment will instead centre upon the automatic production of tags, text labels appended to blog articles for the purposes of categorization. Specifically, it will compare blog tags derived from the LLAMA-B authoring process with blog tags which have been generated manually (by real users) and automatically (using term frequency analysis).

4.1 Experimental Methodology

The first step of the evaluative process involved the creation of an experimental test collection. Using Technorati's RESTful API, we obtained a list of the 100 most frequently used blog tags. After filtering these tags to exclude non-English content, we selected the top ten remaining labels. We then retrieved the first 200 articles recommended by Technorati for each of the top 10 blog tags, excluding duplicate articles, foreign language articles, and articles with no text content. This produced a test bed of 2000 articles, nominally divided into 10 broad categories. The text of each article was manually extracted, passed through a stop list containing 571 terms[1] and stemmed. Completing the pre-processing stage, the entire collection was then indexed using the Lemur Toolkit[2].

The next stage of our experiment examined the *coherence*[3] of each grouping of blog articles. Considering each of the 10 categories as a cluster, we applied the following calculation [6]:

$$Cohesion(Cluster_i) = \frac{\sum_{d_1, d_2 \in Cluster_i, d_1 \neq d_2} Sim(d_1, d_2)}{\sum_{j=1}^{|Cluster_i|} j} \tag{4}$$

where the similarity between two documents d_1 and d_2 was calculated using:

$$Sim(d_1, d_2) = \frac{\sum_{t \in d_1 \bigcup d_2} TFIDF_{d_1}(t) \times TFIDF_{d_2}(t)}{\sqrt{\sum_{t' \in d_1} TFIDF_{d_1}(t')^2 \times \sum_{t' \in d_2} TFIDF_{d_2}(t')^2}} \tag{5}$$

and $TFIDF$ is a score determined for each term in the articles collection in the following way:

$$TFIDF_{d_1}(t) = c(t, d_1) \times \log(\frac{|C|}{docFreq(t)}) \tag{6}$$

where $|C|$ is the number of articles in the collection and $docFreq(t)$ indicates how frequently a term appears in the corpus. The net result of these calculations was a scoring between the ranges of 0 and 1 for each cluster, a measurement indicating the cohesiveness (or interrelatedness) of the articles it contained.

Having examined manually applied blog tags, we then considered blog tags which are generated automatically. Brook and Montanez [6] described a technique for assigning 'autotags' to a blog article. Selecting articles at random from the Technorati search engine, they extracted the three terms with the highest $TFIDF$ scorings as the article's autotags. Subsequently, they grouped those articles that shared a common autotag to form a number of small clusters. Their experimental results indicated that articles clustered using autotags generated in this fashion formed more cohesive groups than articles clustered using manually generated tags.

So, the next stage of the experiment involved duplicating the methodology adopted by Brook and Montanez, using $TFIDF$ scorings to generate 3 autotags for each of the 2000 articles in the test collection. Subsequently, we grouped

[1] ftp://ftp.cs.cornell.edu/pub/smart/

[2] http://www.lemurproject.com

[3] The word *coherence* in this context indicates similarity in terms of content.

those articles sharing identical autotags using a hard clustering scheme to produce several disjoint clusters. We then measured the cluster cohesion score for each grouping using the calculations show above.

The final stage of the experiment involved examining the cluster cohesion of articles grouped using tags derived from our authoring tool. These tags were produced by harvesting the text content of automatically authored link anchors, which can be considered a type of *implicit tag.* The logic at work here is as follows - the higher the frequency with which a given term occurs as a link anchor pointing at a particular article, the greater the likelihood that this term represents a 'good' tag for that article.

Formally, let $inTerms(d)$ indicate all the terms pointing to d from all the other articles in the collection. We counted the number of articles $c(t_i) : t_i \in inTerms(d)$ containing the text anchor t_i as the score for that term w.r.t d. We then arbitrarily chose the top 3 terms with the highest scores as the implicit tags for that article. We then grouped those articles which shared implicit tags to form a number of small clusters. Applying the cluster cohesion calculations once more, we compared the results with the scorings recorded for manually applied tags and keyword based autotags.

4.1.1 Parameters of the authoring runs

We authored two different versions of the test collection as a precursor to implicit tag generation. The first authoring run used the original method for calculating feature dominance and subordinance *(S-D)*. The second method used the PageRank inspired calculation *(P-R)*. To enforce parity, both authoring runs used the same authoring parameters. The number of link anchors per article (denoted α) was fixed at 8. The number of link targets γ was set to 5. The weighting function we employed was the successful *StrengthWeighting* variant [16], with parameter β set to 10. Parameters δ, μ and λ, used for title weighting, MLE smoothing and as part of the *P-R* calculation, were set to 0.25, 1000 and 0.85 respectively.

4.2 Results and Discussion

The cluster cohesion scorings for articles grouped using manually applied blog tags are shown in Figure 1. All illustrated, the cohesion scores for the 10 article groups are very low, rarely exceeding 0.03. This was a fairly surprising result, since previous work had reported figures closer to 0.3 [6]. We ascribe this rather significant variation to the different mechanisms used to extract the text of each article. We used a manual approach to extract the article text during the pre-processing stage of our procedure. However, in the experiment described in [6], article text was extracted using automatic means. It could be that this automatic text extraction process inadvertently introduced some degree of noise into the test collection, thereby skewing the cluster cohesion scorings.

The cluster cohesion scorings for the articles grouped using *TFIDF* generated autotags are also shown in Figure 1. For the most part, these autotags outperform the manually applied tags. However, the cohesion scores vary considerable across the clusters, ranging from a maximum of 0.4574 to a (repeated) minimum of 0.

The cluster cohesion scorings for the implicit tags derived from the two permutations of the LLAMA-B algorithm are shown in Figure 2. The average cohesion scores for each technique, calculated by adding the cohesion scores of each cluster and dividing it by the total number of clusters, are shown in Table 1. As illustrated, both of the permutations of the LLAMA-B algorithm exceeded the average coherency scorings for the *TFIDF* generated clusters. The percentage increase in average coherency over the *TFIDF* approach is, in both cases, quite modest (approaching 11% and 3% respectively). However, these two statistics should be read in combination with a consideration of the observed frequency of incohesive clusters. 68% of the clusters generated using the *TFIDF* methodology were completely incohesive (i.e. they scored zero when the cluster cohesion test was applied). Clusters generated using tags derived from the LLAMA-B authoring process had significantly lower rates of incohesion. One interpretation is this - the LLAMA-B authoring tool produces blog tags resulting in more coherent groups of articles than the autotagging approach of Brook and Montanez, and is considerably less prone to error.

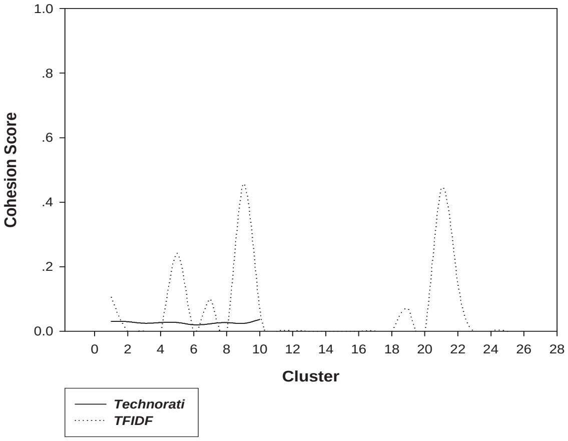

Figure 1: Cohesion scorings for clusters generated using (a) manually generated tags and (b) TFIDF generated tags

4.2.1 Validating the modifications

As discussed in section 3, the LLAMA-B algorithm features a number of improvements which separate it from its theoretical progenitor, the authoring tool known as LLAMA. To verify the effectiveness of these modifications, we conducted another three additional authoring runs, each a variation of the successful *S-D* permutation. In the first authoring run, denoted *S-D-G*, we adopted a global, rather than a local, linking approach. In the second authoring run, denoted *S-D-U*, we ran the LLAMA-B algorithm without MLE smoothing. In the third authoring run, denoted *S-D-NT*, we ran the LLAMA-B algorithm without title weighting. In each of the three cases the average cluster cohesion dropped when our algorithmic improvements were disabled. This finding verifies our decision to incorporate MLE smoothing and title weighting into a local context authoring algorithm.

5. CONCLUSIONS AND FUTURE WORK

In this paper we have discussed an automated authoring tool intended for the blogging community. Given the well-known difficulty of assessing the output of an authoring tool,

Table 1: Blog cluster cohesion scores

	MANUAL	TFIDF	LLAMA-B(S-D)	LLAMA-B(P-R)
Average cohesion	0.027	0.065	0.072	0.067
Incohesive clusters	0%	68.00%	27.50%	32.50%

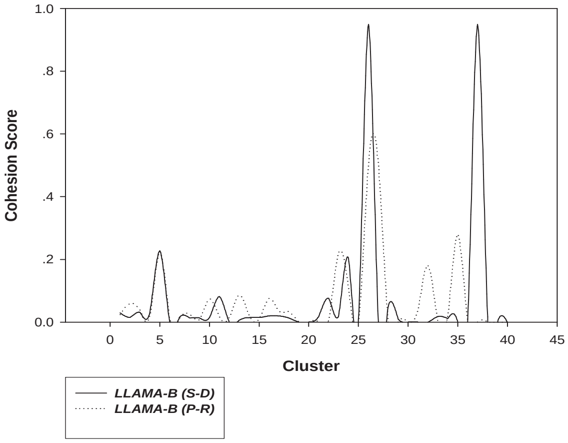

Figure 2: Cohesion scorings for clusters generated using (a) LLAMA-B(S-D) and (b) LLAMA-B(P-R)

we have created an evaluative framework centred upon the unsupervised generation of blog tags. Our experimental results suggest that implicit blog tags, derived from the link authoring process, create more cohesive groups of articles than manually applied or keyword driven tags. We take this to be a very positive result indicating the need for further development of the LLAMA-B algorithm.

There are two main ways in which this study can be extended. Firstly, a user-centred trial to examine the usefulness of the additional authored links is strongly indicated. Secondly, the LLAMA-B algorithm should be examined as a mechanism facilitating community discovery. The results described above comfortably demonstrate that LLAMA-B can accurately identify interconnected groups of blog articles. Its application to larger collections of blogs, possibly the *blogosphere* in its entirety, could yield substantial gains in terms of basic connectivity and community awareness.

6. REFERENCES

[1] E. Adar and L. A. Adamic. Tracking information epidemics in blogspace. In *WI '05: Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 207–214, Washington, DC, USA, 2005. IEEE Computer Society.

[2] H. Ashman. Electronic document addressing: dealing with change. *ACM Comput. Surv.*, 32(3):201–212, 2000.

[3] P. Bailey, N.Craswell, and D. Hawking. Dark matter on the web. In *Proceedings of the 9th International World Wide Web Conference, Poster Track*, 2000.

[4] J. Blustein. Automatically generated hypertext versions of scholarly articles and their evaluation. In *Proceedings of the eleventh ACM on Hypertext and hypermedia*, pages 201–210, San Antonio, Texas, United States, 2000. ACM Press. 336364.

[5] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Proceedings of the 7th International World Wide Web Conference, reprinted in Eds H.Ashman and P.Thistlewaite Comput. Netw. ISDN Syst.*, 30(1-7):107–117, 1998. 297827.

[6] C. H. Brooks and N. Montanez. Improved annotation of the blogosphere via autotagging and hierarchical clustering. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 625–632, New York, NY, USA, 2006. ACM.

[7] Y. Chi, S. Zhu, X. Song, J. Tatemura, and B. L. Tseng. Structural and temporal analysis of the blogosphere through community factorization. In *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 163–172, New York, NY, USA, 2007. ACM.

[8] D. Gruhl, R. Guha, D. Liben-Nowell, and A. Tomkins. Information diffusion through blogspace. In *WWW '04: Proc. of the 13th Int. conf. on World Wide Web*, pages 491–501, New York, NY, USA, 2004. ACM.

[9] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, 1999. 324140.

[10] C. Nentwich, L. Capra, W. Emmerich, and A. Finkelstein. xlinkit: A consistency checking and smart link generation service. *ACM Trans. on Internet Technology*, 2(2):151–185, 2002.

[11] A. Qamra, B. Tseng, and E. Y. Chang. Mining blog stories using community-based and temporal clustering. In *CIKM '06: Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 58–67, New York, NY, USA, 2006. ACM.

[12] P. Thistlewaite. Automatic construction and management of large open webs. *Inf. Process. Manage.*, 33(2):161–173, 1997.

[13] M. Truran, J. Goulding, and H. Ashman. Autonomous authoring tools for hypertext. *ACM Computing Surveys (CSUR)*, 39(3):8, 2007.

[14] E. Valle, P. Castagna, and M. Brioschi. Towards a semantic enterprise information portal. In *2nd Int. Conf. on Knowledge Capture, (Workshop on Knowledge Management and the Semantic Web).* ACM Press, 2003.

[15] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Trans. Inf. Syst.*, 22(2):179–214, 2004. 984322.

[16] D. Zhou, J. Goulding, M. Truran, and T. Brailsford. Llama: automatic hypertext generation utilizing language models. In *HT '07: Proceedings of the 18th conference on Hypertext and hypermedia*, pages 77–80, New York, NY, USA, 2007. ACM Press.