An Improvement of Weighted PageRank to Handle the Zero Link Similarity

Azma Yusuf
Institut Teknologi Batam
Batam, Kepulauan Riau, Indonesia
2022017@student.iteba.ac.id

***Abstract*—Algoritme PageRank yang terkenal memanfaatkan struktur tautan untuk menghitung peringkat kualitas halaman. Ini pada dasarnya memberikan jumlah probabilitas yang sama ke halaman tetangga dari suatu halaman. Sebagai ekstensinya, algoritma PageRank berbobot telah diusulkan yang memberikan bobot berbeda pada tautan keluar dari sebuah halaman. Beberapa algoritma PageRank berbobot menggunakan kesamaan antar halaman sebagai bobot. Di halaman web Korea, kami menemukan bahwa terkadang memiliki nilai nol untuk kesamaan antar halaman dari halaman tetangga karena karakteristik bahasa. Makalah ini mengusulkan peningkatan algoritma PageRank berbobot yang dapat menangani kesamaan antar halaman nol tersebut. Metode yang diusulkan telah diimplementasikan menggunakan paradigma MapReduce untuk penanganan data besar, dan telah dievaluasi melalui halaman web Wikipedia bahasa Korea dan dibandingkan dengan dua metode lainnya.**

***Keywords*—PageRank; Weighted PageRank; Similarity; MapReduce; TFIDF**

I. Introduction

Saat ini, ketika orang ingin mengetahui sesuatu, banyak dari mereka mencoba mencarinya di Internet. Mereka akan kewalahan jika terlalu banyak halaman yang diberikan sebagai halaman yang relevan dalam pencarian web. Dalam information retrieval (IR), pemeringkatan menjadi salah satu isu krusial. Untuk memilah yang berpengaruh di antara halaman yang dicari, berbagai algoritma peringkat telah diusulkan. [1]–[11]

PageRank [1] adalah salah satu algoritma peringkat terkenal yang menggunakan struktur link Web. Ini mengasumsikan bahwa seorang peselancar berjalan secara acak di atas halaman web dan mencoba untuk menentukan distribusi statis dari peselancar. Dengan metafora penderitaan acak, semakin banyak tautan yang dimiliki halaman, semakin tinggi peringkatnya. Di PageRank, penderitaan membuat jalan acak ke halaman tetangga dengan probabilitas yang sama. Kadang-kadang probabilitas yang sama ini tampaknya tidak masuk akal karena beberapa tautan terhubung ke halaman tetangga yang jauh lebih penting.

Untuk mengatasi situasi ini, algoritma PageRank berbobot [2], [3], [8] telah diusulkan. Mereka memperhitungkan baik distribusi jumlah in-link untuk node tetangga, the jumlah kunjungan ke halaman tetangga, atau antar halaman kesamaan. Masing-masing memiliki pro dan kontra. Pembobotan berdasarkan kesamaan antar halaman terdengar baik dalam melayani peringkat berbasis konten. Kami telah mencoba pembobotan berbasis kesamaan antar halaman Algoritma PageRank ke halaman Wikipedia bahasa Korea. Untuk menghitung kesamaan antar halaman, kami menggunakan vektor model [13]. Untuk mendapatkan representasi vektor untuk halaman, pertama-tama kita melakukan analisis morfologi untuk mengekstrak kata-kata. Berbeda dengan bahasa barat, kata benda dalam bahasa Korea paling banyak menyampaikan informasi yang berarti. Karena karakteristik bahasa, kata benda diekstraksi untuk mengidentifikasi kata kunci. Kata kunci diidentifikasi menggunakan istilah frekuensi dan dokumen terbalik informasi frekuensi. Dengan kata kunci, vektor representasi untuk halaman diperoleh. Kemudian, itu terjadi pada memiliki kesamaan nol ketika kesamaan antar halaman dihitung menggunakan produk dalam dari vektor tersebut. Ini sangat canggung untuk halaman tetangga tidak memiliki kesamaan. makalah ini adalah terkait dengan peningkatan pada PageRank berbobot berbasis kesamaan antar halaman untuk menangani kasus dengan nol link kesamaan.

Sisa makalah ini disusun sebagai berikut: Bagian II menyajikan PageRank dan variannya lebih detail. Bagian III memperkenalkan peningkatan pada pembobotan PageRank, dan Bagian IV menunjukkan beberapa hasil eksperimen untuk metode yang diusulkan. Kami menarik kesimpulan di Bagian V.

II. Related works

A. PageRank Algorithm

PageRank [1] adalah algoritma perbatasan yang memberi peringkat halaman dengan mengacu pada struktur link Web. suguhan PageRank halaman sebagai node dan hyperlink sebagai tepi grafik. Setiap simpul memiliki nilai Rank sendiri dan mendistribusikannya secara merata ke tetangganya. Distribusi berulang tanpa batas sampai semua nilai peringkat adalah konvergen. Distribusi stasioner dari Ranks adalah dianggap sebagai skor Peringkat akhir halaman. Untuk mencegah peningkatan tak terbatas dalam nilai Peringkat, itu dibatasi untuk jumlah dari semua Peringkat menjadi 1, dan juga untuk setiap nilai Peringkat menjadi tidak lebih besar dari 1.Nilai peringkat dari simpul j dihitung sebagai berikut:

$$r_j = \sum_{i_{\text{out}}} \frac{r_i}{L_{\text{out}}\,(i)} \tag{1}$$

$$\sum r_i = 1 \tag{2}$$

dimana L_{out} adalah jumlah out-link dari node i.Setiap simpul i mendistribusikan skor peringkatnya r_i secara merata simpul tetangganya j. SEBUAH node j mengumpulkan semua skor Rank yang dikirimkan dari tetangga node dan ambil jumlah mereka sebagai skor Peringkat baru r_j. Peringkat ini proses distribusi dapat dinyatakan dengan kedekatan stokastik matriks M dan vektor pangkat r. Matriks M adalah tetangganya matriks untuk web yang mengkodekan hubungan lingkungan antara halaman dan distribusi stasioner nilai Peringkat. Baru Nilai peringkat $r^{(t+1)}$ dihitung sebagai berikut:

$$r^{(t+1)} = \mathrm{M}\mathrm{r}^{(t)} \tag{3}$$

di mana $t+1$ adalah langkah selanjutnya dari t.Ketika semua nilai peringkat adalah konvergen, $r^{(t+1)}$ sangat mirip dengan $r^{(t)}$. Di sini, konvergen rsesuai dengan vektor eigen dengan nilai eigen 1 untuk matriks M.

B. WeightedPageRank Algorithms

Surfers sebenarnya tidak melakukan random walk seperti di PageRank. Untuk mengakomodasi karakteristik perilaku seperti itu, pembobotan Algoritma PageRank telah diusulkan yang memungkinkan penderitaan untuk membuat transisi probabilistik yang tidak merata ke tetangga halaman. [2], [3], [8]

1) Weighted PageRank Based on the number of in-links of neighboring pages:

Xing dan Ghorbani [2] mengusulkan PageRank algoritma yang memberikan lebih banyak porsi Rank ke tetangga halaman dengan lebih banyak tautan. Ya tidak cukup mencerminkan perilaku peselancar yang sebenarnya, karena hanya informasi struktur topologi digunakan.

2) Weighted PageRank based on Similarity Measure:

Qiaoet al. [3] menyarankan varian PageRank berbobot algoritma, yang disebut SimRank, yang mendistribusikan nilai Rank di porsi kesamaan antar halaman. Untuk menerapkan metode, semua kesamaan halaman berpasangan perlu dihitung lebih awal. Secara komputasi mahal untuk menerapkan metode ini untuk skala besar volume halaman. Oleh karena itu untuk menerapkan metode, perlu infrastruktur komputasi paralel terdistribusi seperti Hadoop Pengurangan Peta [14].

3) Weighted PageRank based on visits of links:

Kumaret al. [8] memperkenalkan algoritma PageRank berbobot di mana node mendistribusikan lebih banyak nilai Rank ke yang keluar tautan yang lebih sering dikunjungi oleh pengguna. Ini algoritme memerlukan data klik tautan di seluruh Web. Oleh karena itu, sangat ideal, tetapi tidak mudah untuk menerapkannya pada skala web.

C. Hub and Authorities Algorithm

Najork [7] mengambil pendekatan yang agak berbeda dari PageRank dan mengusulkan algoritma peringkat yang disebut HITS (hyperlink pencarian topik yang diinduksi). Sementara PageRank mengasumsikan gagasan satu dimensi tentang pentingnya halaman, tampilan HITS halaman penting memiliki dua rasa penting. [15] Itu algoritma karena itu memberikan dua skor untuk setiap halaman. Yakin halaman sangat berharga karena memberikan informasi tentang tema. Halaman-halaman ini disebut otoritas. halaman lainnya adalah berharga karena mereka memberi tahu Anda ke mana harus pergi untuk mencari tahu topik itu. Halaman-halaman ini disebut hub. Algoritma mendefinisikan dua konsep secara rekursif. Sebuah halaman adalah dianggap sebagai hub yang baik jika terhubung dengan otoritas yang baik. Sebuah halaman adalah dianggap sebagai otoritas yang baik jika dihubungkan dengan hub yang baik. [15]

D. Distributed and Parallel Computing

Pengambilan informasi dari repositori data besar, seperti Web dan penyimpanan data besar membutuhkan infrastruktur komputasi yang menyimpan dan memproses data tersebut. Kita dapat menggunakan salah satu dari sistem superkomputer atau komputasi terdistribusi dan paralel sistem.

Hadoop [14] adalah infrastruktur komputasi yang baik yang dapat ditetapkan dalam biaya ekonomi. Ini adalah proyek Apache untuk platform komputasi terdistribusi yang menyediakan seperti sistem file terdistribusi yang disebut HDFS (Hadoop Distributed File System) dan kerangka kerja komputasi paralel terdistribusi yang disebut Kurangi Peta. Kerangka kerja MapReduce mengatur pekerjaan ke dalam Peta tugas dan Mengurangi tugas. Data input dipartisi dan diproses oleh proses Peta, dan hasil pemrosesannya dibentuk menjadi pasangan nilai kunci. Hasil tugas peta dikocok menjadi Reduce tugas sesuai dengan kunci mereka. Kurangi proses agregat nilai dengan kunci yang sama, untuk mendapatkan hasil akhir. Ini kerangka kerja komputasi memungkinkan kita untuk menangani beban berat komputasi seperti komputasi kesamaan halaman berpasangan.

III. THE PROPOSED ALGORITHM

PageRank berbobot berbasis kesamaan antar halaman pada kesamaan tidak dapat menangani situasi yang antar halaman kesamaannya adalah 0. Untuk menghadapi situasi ini, kami mengusulkan a metode untuk menjaga kesamaan antar halaman nol dan untuk menyesuaikan bobot untuk distribusi nilai Rank.

Ekstraksi kata kunci berbasis kata benda seperti di halaman Korea terkadang menemukan kata kunci umum di antara halaman yang ditautkan. Terlepas dari gagasan yang melekat tentang kesamaan antar halaman untuk memperkirakan frekuensi traversal tautan, situasi nol-kesamaan menghalangi penerapan algoritma PageRank berbobot.

Untuk meningkatkan penerapan PageRank berbobot, kami mengusulkan metode untuk menjamin beberapa bobot minimum dan menyesuaikan bobot. PageRank berbobot yang diusulkan berfungsi sebagai berikut, yang pada dasarnya berperilaku dengan cara yang sama seperti Qiao algoritme et al. [3]

Berdasarkan ukuran kemiripan, bobot w_{ij} pada node i ke j dihitung sebagai berikut:

$$w_{ij} = \frac{s_{ij}}{\sum_{k \in L_{\text{out}}\,(i)} s_k} \tag{4}$$

di mana s_{ij} adalah kesamaan antara halaman i dan j dan $L_{\text{out}}\,(i)$ menunjukkan halaman yang ditunjuk oleh halaman i.

Nilai Peringkat r_j dari halaman j diperbarui, hingga semua peringkat nilai konvergen, sebagai berikut:

$$r_j = \sum i \in L_{in}(j) \beta w_{ij} r_i + (1-\beta)\frac{1}{N} \tag{5}$$

di mana β menunjukkan tingkat teleportasi seperti di PageRank, $L_{ln}(j)$ adalah halaman yang mengarah ke halaman j, dan N adalah totalnya jumlah halaman.

Untuk mengukur kesamaan antar halaman, kami menggunakan cosinus jarak antara vektor kata kunci yang elemennya adalah Nilai TFIDF dari lemma. Lemma diekstraksi dari Halaman Korea menggunakan penganalisa morfologi Korea. TF (Frekuensi Istilah) adalah frekuensi lemma dalam satu halaman, dan IDF (Frekuensi dokumen terbalik) adalah frekuensi halaman mengandung lemma. [13] Untuk kata kunci di halaman, TFIDF-nya dihitung sebagai berikut:

$$\text{TFIDF} = \frac{TF}{\log\left(\frac{DF}{N}\right)} \tag{6}$$

Selanjutnya, kesamaan S_{ij} antara halaman i dan j dihitung dengan jarak cosinus antara vektor kata kunci TFIDF nilai:

$$s_{ij} = \frac{K_i \cdot K_j}{|K_i||K_j|} \tag{7}$$

di mana $K_i$adalah vektor kata kunci halaman i.

Menggunakan kesamaan dari Persamaan (7), bobot dihitung sebagai: Persamaan (4). Namun, itu tidak mempertimbangkan situasi bahwa kesamaan antar halaman adalah nol. Oleh karena itu, kami mengusulkan suatu algoritma, yang menanganinya dengan mengalokasikan kesamaan minimum ke tautan ke halaman dengan kesamaan nol.

$$\rho\frac{\min\left(s_{ij}\right)}{\sum_{L_{in}(l)} s_{ik}} = \alpha(1-\rho)ZR \tag{8}$$

di mana ρ adalah parameter yang disediakan pengguna untuk kesamaan nol, dan ZR adalah jumlah tautan kesamaan bukan nol.

Persamaan (8) dikembangkan untuk membuat kesamaan yang disesuaikan dengan bobot kesamaan nol lebih kecil dari kesamaan bukan nol nilai-nilai. ditentukan menurut Persamaan(8), dimana kesamaan minimum dikendalikan oleh α. Bersama dengan yang disesuaikan kesamaan baru, bobotnya hanya perlu dihitung ulang sebagai biasa. Pada akhirnya, nilai peringkat ditentukan seperti pada algoritma PageRank berbobot dengan Persamaan (4).

IV. Experiments

Untuk mengevaluasi kinerjanya, kami menerapkan metode yang diusulkan untuk menentukan peringkat halaman web dalam bahasa Korea Wikipedia. Kami telah mengumpulkan sekitar 300.000 halaman dari Wikipedia bahasa Korea. Gambar 1 menunjukkan sistem percobaan Arsitektur

Seluruh halaman dari ko.wikipedia.org telah dirayapi dan disimpan ke dalam Hadoop HDFS. Halaman diurai menggunakan a Penganalisa morfologi Korea untuk mengekstrak lemma. hadoop Program MapReduce dikembangkan untuk menghitung kejadian dari kata-kata di halaman. Berdasarkan data jumlah kata, TFIDF untuk lemma setiap halaman dihitung untuk menentukan kata kunci, dan vektor kata kunci dibuat untuk setiap halaman. Vektor kata kunci dinyatakan dengan dihitung nilai TFIDF. Kesamaan antara tetangga halaman dihitung menggunakan jarak cosinus. Akhirnya, nilai peringkat ditentukan oleh Persamaan (4) menggunakan bobot yang dihitung.

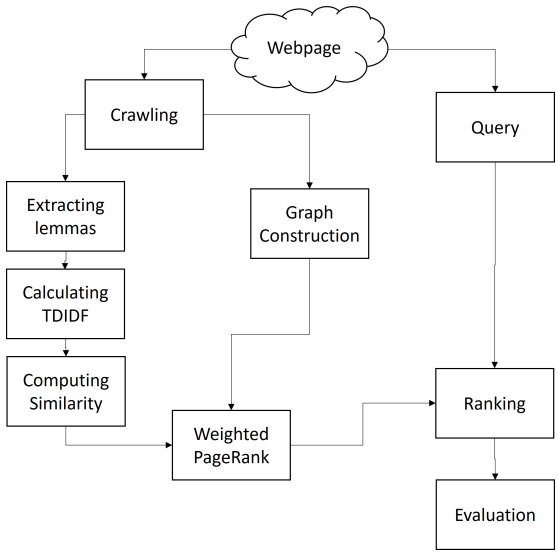

Fig. 1. Experiment System Architecture

Dalam percobaan, kami menggunakan cluster Hadoop dari 5 node untuk menangani volume data yang besar. Semua tugas yang diusulkan metode telah diimplementasikan dalam program MapReduce. Kami melakukan percobaan 10 kali untuk dipilih secara acak 10 kata kunci dari Wikipedia dan menemukan halaman yang mengandung kata kueri dan mengurutkannya menurut urutan menurun dari nilai peringkat mereka. Kemudian kami memilih 20 halaman

teratas dan dievaluasi relevansinya dengan 5 skala level. Kami menghitung Keuntungan Kumulatif Diskon yang Dinormalisasi (NDCG) [13] untuk 20 halaman teratas untuk setiap kueri. NDCG adalah metrik evaluasi yang digunakan untuk mengevaluasi kinerja mesin pencari web. Ini memberikan nilai dari 0,0 hingga 1,0, dan nilai 1,0 adalah yang ideal peringkat entitas. Halaman kebenaran dasar untuk pertanyaan adalah ditentukan dengan memilih halaman tautan keluar dari halaman kueri di penurunan urutan kesamaan.

Gambar 2 menunjukkan hasil percobaan untuk aslinya PageRank, Qias dkk.[3] metode dan metode yang diusulkan dalam hal NDCG. Diamati bahwa metode yang diusulkan telah memberikan peningkatan sekitar 2% rata-rata selama PageRank asli. Selama Qias et al. [3], yang diusulkan metode meningkat di NDCG sekitar 1,3%.

Dari percobaan, kami mengamati bahwa yang diusulkan metode menghasilkan hasil yang sedikit lebih baik secara rata-rata dibandingkan dengan metode lainnya.

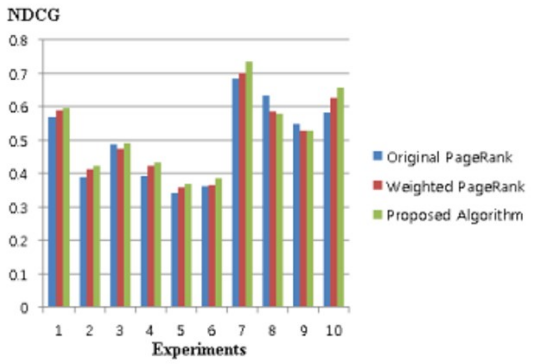

Fig. 2. Comparison of the original PageRank and the proposed Algorithm

V. Conclusion

Dalam penelitian ini, kami menganalisis perilaku tertimbang algoritma PageRank dan mengidentifikasi bahwa kesamaan algoritma PageRank berbobot berbasis tidak dapat bekerja dengan baik di beberapa situasi, terutama, ketika kata kunci kata benda adalah diekstraksi dari halaman Korea untuk perhitungan kesamaan. baru varian dari algoritma PageRank berbobot diusulkan untuk menangani nol kesamaan antar halaman. Untuk volume data yang besar pemrosesan, algoritma yang diusulkan diimplementasikan dalam Program MapReduce dan set data eksperimental adalah diproses pada cluster Hadoop dari 5 node. yang diusulkan algoritma telah diterapkan ke Wikipedia Korea untuk evaluasi kinerja. Dalam percobaan, kami menerapkan tiga algoritma: PageRank asli, Qias et al.'s PageRank berbobot [3], dan metode yang diusulkan. Dari percobaan kami telah mengamati metode yang diusulkan tercapai beberapa peningkatan dalam hal NDCG dibandingkan yang dibandingkan metode.

Acknowledgment

Penelitian ini didukung oleh MSIP (Kementerian Sains, ICT dan Perencanaan Masa Depan), Korea, di bawah the Dukungan ITRC (Pusat Penelitian Teknologi Informasi) program (NIPA-2013-H0301-13-4009) diawasi oleh NIPA (Badan Promosi Industri TI Nasional)

References

[1] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," *Computer networks and ISDN systems*, vol. 30, no. 1-7, pp. 107–117, 1998.

[2] W. Xing and A. Ghorbani, "Weighted pagerank algorithm," in *Proceedings. Second Annual Conference on Communication Networks and Services Research, 2004.* IEEE, 2004, pp. 305–314.

[3] S. Qiao, T. Li, H. Li, Y. Zhu, J. Peng, and J. Qiu, "Simrank: A page rank approach based on similarity measure," in *2010 IEEE International Conference on Intelligent Systems and Knowledge Engineering*. IEEE, 2010, pp. 390–395.

[4] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web." Stanford InfoLab, Tech. Rep., 1999.

[5] K. Kumar and F. D. M. Abhaya, "Pagerank algorithm and its variations: A survey report," *IOSR Journal of Computer Engineering (IOSR-JCE)*, vol. 14, no. 1, pp. 38–45, 2013.

[6] N. Duhan, A. Sharma, and K. K. Bhatia, "Page ranking algorithms: a survey," in *2009 IEEE International Advance Computing Conference*. IEEE, 2009, pp. 1530–1537.

[7] M. A. Najork, "Comparing the effectiveness of hits and salsa," in *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, 2007, pp. 157–164.

[8] G. Kumar, N. Duhan, and A. Sharma, "Page ranking based on number of visits of links of web page," in *2011 2nd International Conference on Computer and Communication Technology (ICCCT-2011)*. IEEE, 2011, pp. 11–14.

[9] D. Nemirovsky and K. Avrachenkov, "Weighted pagerank: cluster-related weights," SAINT PETERSBURG STATE UNIV (RUSSIA), Tech. Rep., 2008.

[10] N. Tyagi and S. Sharma, "Weighted page rank algorithm based on number of visits of links of web page," *International Journal of Soft Computing and Engineering (IJSCE) ISSN*, pp. 2231–2307, 2012.

[11] T. H. Haveliwala, "Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search," *IEEE transactions on knowledge and data engineering*, vol. 15, no. 4, pp. 784–796, 2003.

[12] P. Resnik, "Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language," *Journal of artificial intelligence research*, vol. 11, pp. 95–130, 1999.

[13] H. Schütze, C. D. Manning, and P. Raghavan, *Introduction to information retrieval*. Cambridge University Press Cambridge, 2008, vol. 39.

[14] J. Dean and S. Ghemawat, "Mapreduce: simplified data processing on large clusters," *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, 2008.

[15] A. Rajaraman and J. D. Ullman, *Mining of massive datasets*. Cambridge University Press, 2011.

[16] D. Kim, K.-s. Kim, K.-H. Park, J.-H. Lee, and K. M. Lee, "A music recommendation system with a dynamic k-means clustering algorithm," in *Sixth International Conference on Machine Learning and Applications (ICMLA 2007)*. IEEE, 2007, pp. 399–403.

[17] K. M. Lee, K. M. Lee, and C. H. Lee, "Statistical cluster validity indexes to consider cohesion and separation," in *2012 international conference on fuzzy theory and its applications (ifuzzy2012)*. IEEE, 2012, pp. 228–232.

[18] K. M. Lee, "Adaptive resource scheduling for workflows considering competence and preference," in *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*. Springer, 2004, pp. 723–730.

[19] H. Lee-Kwang, K. A. Seong, and K.-M. Lee, "Hierarchical partition of nonstructured concurrent systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 27, no. 1, pp. 105–108, 1997.

[20] K. M. Lee and J.-H. Lee, "Coordinated collaboration of multiagent systems based on genetic algorithms," in *Pacific Rim International Workshop on Multi-Agents*. Springer, 2003, pp. 145–157.

[21] C. Y. Yoon and K. M. Lee, "An end-user evaluation system based on computing competency and complex indicators," in *2007 IEEE International Conference on Information Reuse and Integration*. IEEE, 2007, pp. 480–485.