

Listwise Learning to Rank Approach Using ListNet

^{1st} Alfendo S.P. Situmorang
Fakultas Informatika dan Teknik Elektro
Institut Teknologi Del
Balige, Indonesia
alfendosps@gmail.com

^{2th} Reinheart Christian Simanungkalit
Fakultas Informatika dan Teknik Elektro
Institut Teknologi Del
Medan, Indonesia
reinheartcs@gmail.com

^{3th} Frans Daniel Yehuda Sitohang
Fakultas Informatika dan Teknik Elektro
Institut Teknologi Del
Balige, Indonesia
franssitohang12@gmail.com

^{4th} Dewi Purnama Napitupulu
Fakultas Informatika dan Teknik Elektro
Institut Teknologi Del
Porsea, Indonesia
fleurapurnama27@gmail.com

Abstract—*Learning to Rank* adalah penerapan pembelajaran mesin yang biasanya menerapkan *supervised machine learning* untuk melakukan pemeringkatan. Pada paper ini kelompok mengusulkan algoritma *ListNet* yang merupakan salah satu teknik dimana suatu peringkat akan diubah menjadi *listwise classification*. Algoritma ini akan mempertimbangkan pasangan *query-document* untuk melakukan pemeringkatan.

Keywords—*Learning to Rank*, *ListNet*, *listwise*.

I. PENDAHULUAN

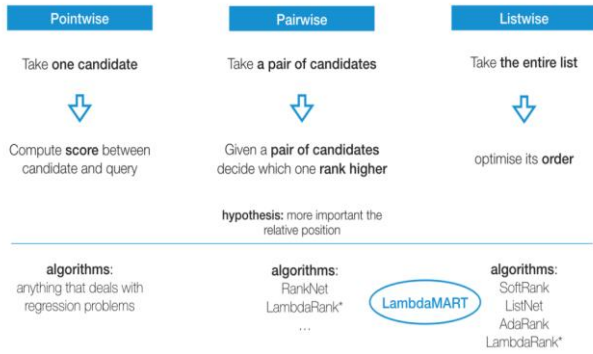
Information Retrieval (IR) adalah proses mencari dan memperoleh informasi yang dibutuhkan oleh pengguna. IR membantu memuaskan para pengguna menemukan sumber informasi yang relevan dari kumpulan sumber informasi yang tersedia. Kebutuhan informasi pengguna ditentukan oleh *query* yang diberikan ke sistem yang telah memiliki kumpulan sumber informasi. Setiap sumber informasi yang memenuhi *query* memberikan tingkat relevansi yang bervariasi antar yang satu dengan yang lain sehingga dapat disimpulkan dalam pengambilan *ranking* sesuai tingkat relevansi yang diberikan.

Hasil *ranking* yang memiliki kualitas yang tinggi sangat berpengaruh untuk *web search*, sehingga kebutuhan informasi pengguna yang diharapkan akan ditampilkan di bagian atas mesin pencarian [1]. Salah satu bagian terpenting dari sistem IR adalah model yang dapat memberi peringkat pada dokumen, *Learning to Rank* (LTR) adalah salah satu bidang yang menyediakan teknik dan algoritma untuk pemeringkatan. [2].

Learning to Rank (LTR) merupakan salah satu metode yang dapat diterapkan secara efektif untuk menyelesaikan tugas pembuatan model pemeringkatan dalam *Information*

Retrieval. *Learning to Rank* membantu memecahkan masalah IR seperti *document retrieval*, *collaborative filtering*, *sentiment analysis*, *computational advertising*, dll. Metode *Learning to Rank* bertujuan untuk mempelajari model yang diberikan *query* dan sekumpulan dokumen kandidat untuk menemukan peringkat dokumen yang sesuai sesuai dengan relevansinya. *Learning to Rank* adalah metode dengan pendekatan *supervised learning* yang menggunakan pasangan *query* dan dokumen yang dilengkapi dengan skor relevansi sebagai labelnya. *Learning to Rank* akan membuat model pemeringkatan secara otomatis dari kumpulan data yang digunakan [3]. *Ranking model* biasanya digunakan untuk menetapkan skor peringkat ke sekumpulan dokumen baru dengan relevansi yang tidak diketahui. Performa teknik *Learning to Rank* bergantung pada jumlah data pelatihan yang tersedia. Ada banyak algoritma yang tersedia untuk menyelesaikan masalah *Learning to Rank*. Pada kesempatan ini, kelompok menggunakan algoritma *ListNet* yang memiliki ide dasar yang menggabungkan pemodelan *Neural Network* dan algoritma *Gradient Descent* untuk menjalankan *listwise loss function*. Kinerja metode tersebut terbukti mampu melebihi metode *pairwise* seperti *Ranking SVM*, *RankBoost*, dan *RankNet* [4]. Tahapan *Learning to Rank* dibagi menjadi dua bagian, yakni *learning system* dan *ranking system* [6]. Pada tahap *learning system* akan dilakukan pelatihan terhadap *data train* menggunakan suatu metode *ranking*, dengan tujuan untuk memperoleh suatu model pemeringkatan yang paling optimal [6]. Ukuran optimal untuk sebuah model adalah ketika memperoleh akurasi pemeringkatan yang tinggi [6]. Terdapat tiga pendekatan yang dilakukan dalam menyelesaikan permasalahan pemeringkatan [6]. Ketiga pendekatan tersebut

adalah *pointwise approach*, *pairwise approach* dan *listwise approach*. Perbedaan dari ketiga pendekatan dapat dilihat pada gambar di bawah ini [5].



Gambar 2. Perbedaan ketiga Pendekatan Learn to rank

Sumber: Learning to (Retrieve and) Rank — Intuitive Overview, 2017

II. METODOLOGI

A. Pendekatan Listwise

Pada bagian ini akan dibahas terkait pendekatan *listwise* pada *Learn to Rank*, dimana pendekatan ini peringkat diubah menjadi *listwise classification*. Pendekatan ini menggenerasi beberapa daftar dokumen induk berdasarkan satu *query*. Masing-masing daftar *ranking* akan melakukan penghitungan skor, lalu melakukan perbandingan skor.

Pendekatan ini memiliki performa yang lebih baik dibandingkan dengan pendekatan lain terutama pada perhitungan data latih. Namun pendekatan ini cukup kompleks.

B. ListNet

Listnet adalah metode pembelajaran yang berfokus pada optimasi *listwise loss function*. Metode ini menerapkan model *Neural Network* dan algoritma *Gradient Descent*. Dengan menekankan probabilitas tertinggi k, didapatkan

$$\Delta \omega = \frac{\partial L(y^{(i)}, z^{(i)}(f_{\omega}))}{\partial \omega} = - \sum_{j=1}^{n^{(i)}} P_{y^{(i)}}(x_j^{(i)}) \frac{\partial f_{\omega}(x_j^{(i)})}{\partial \omega} + \frac{1}{\sum_{j=1}^{n^{(i)}} \exp(f_{\omega}(x_j^{(i)}))} \sum_{j=1}^{n^{(i)}} \exp(f_{\omega}(x_j^{(i)})) \frac{\partial f_{\omega}(x_j^{(i)})}{\partial \omega}$$

untuk nilai k = 1.

C. Dataset

Pengujian terhadap metode ListNet dilaksanakan dengan dataset LETOR 4.0 yang didapatkan dari *Microsoft IR Research Group* [7]. Dataset telah dibagi ke dalam 5 fold untuk mempermudah analisis.

III. HASIL

Dari pengujian metode yang diusulkan, didapatkan hasil sebagai berikut.

```
INFO:root:Loading dataset
INFO:root:loaded 471 sets for training
INFO:root:loaded 157 sets for dev
INFO:root:loaded 156 sets for test
INFO:root:[ 0] T/loss=0.142821 T/acc=0.451238 D/loss=0.148315 D/acc=0.487140 lr=0.000700
INFO:root:[ 1] T/loss=0.137200 T/acc=0.468169 D/loss=0.147032 D/acc=0.496971 lr=0.000696
INFO:root:[ 2] T/loss=0.136525 T/acc=0.475992 D/loss=0.146266 D/acc=0.506027 lr=0.000693
INFO:root:[ 3] T/loss=0.135640 T/acc=0.477695 D/loss=0.147637 D/acc=0.515359 lr=0.000690
INFO:root:[ 4] T/loss=0.135373 T/acc=0.482647 D/loss=0.148527 D/acc=0.522586 lr=0.000686
INFO:root:[ 5] T/loss=0.134315 T/acc=0.485358 D/loss=0.143353 D/acc=0.516742 lr=0.000683
INFO:root:[ 6] T/loss=0.132754 T/acc=0.480787 D/loss=0.149556 D/acc=0.501629 lr=0.000679
INFO:root:[ 7] T/loss=0.132787 T/acc=0.484308 D/loss=0.146763 D/acc=0.493562 lr=0.000676
INFO:root:[ 8] T/loss=0.133133 T/acc=0.483089 D/loss=0.147581 D/acc=0.508903 lr=0.000672
INFO:root:[ 9] T/loss=0.132194 T/acc=0.496119 D/loss=0.149219 D/acc=0.512974 lr=0.000669
INFO:root:[10] T/loss=0.131840 T/acc=0.484854 D/loss=0.145007 D/acc=0.515641 lr=0.000666
INFO:root:[11] T/loss=0.130412 T/acc=0.482546 D/loss=0.145197 D/acc=0.516779 lr=0.000662
INFO:root:[12] T/loss=0.130120 T/acc=0.493569 D/loss=0.149141 D/acc=0.487824 lr=0.000659
INFO:root:[13] T/loss=0.128984 T/acc=0.491816 D/loss=0.147632 D/acc=0.509254 lr=0.000656
INFO:root:[14] T/loss=0.127938 T/acc=0.496348 D/loss=0.153018 D/acc=0.495097 lr=0.000653
INFO:root:[15] T/loss=0.128657 T/acc=0.497189 D/loss=0.153891 D/acc=0.512954 lr=0.000649
INFO:root:[16] T/loss=0.126758 T/acc=0.492667 D/loss=0.153465 D/acc=0.487291 lr=0.000646
INFO:root:[17] T/loss=0.126986 T/acc=0.494533 D/loss=0.149780 D/acc=0.510668 lr=0.000643
INFO:root:[18] T/loss=0.128086 T/acc=0.501695 D/loss=0.147584 D/acc=0.510836 lr=0.000640
INFO:root:[19] T/loss=0.125542 T/acc=0.492811 D/loss=0.150435 D/acc=0.501242 lr=0.000636
INFO:root:[20] T/loss=0.125497 T/acc=0.497087 D/loss=0.151781 D/acc=0.510658 lr=0.000633
INFO:root:[21] T/loss=0.125004 T/acc=0.503915 D/loss=0.152182 D/acc=0.510668 lr=0.000630
INFO:root:[22] T/loss=0.123756 T/acc=0.504938 D/loss=0.148343 D/acc=0.498576 lr=0.000627
INFO:root:[23] T/loss=0.123647 T/acc=0.502738 D/loss=0.146208 D/acc=0.510448 lr=0.000624
INFO:root:[24] T/loss=0.122756 T/acc=0.505678 D/loss=0.152791 D/acc=0.493319 lr=0.000621
INFO:root:[25] T/loss=0.122737 T/acc=0.501166 D/loss=0.150823 D/acc=0.502078 lr=0.000618
INFO:root:[26] T/loss=0.121759 T/acc=0.512659 D/loss=0.151895 D/acc=0.500165 lr=0.000614
INFO:root:[27] T/loss=0.121403 T/acc=0.502772 D/loss=0.153618 D/acc=0.489144 lr=0.000611
INFO:root:[28] T/loss=0.120426 T/acc=0.513349 D/loss=0.151285 D/acc=0.501996 lr=0.000608
INFO:root:[29] T/loss=0.118362 T/acc=0.507774 D/loss=0.151392 D/acc=0.507934 lr=0.000605
INFO:root:[30] T/loss=0.120486 T/acc=0.519071 D/loss=0.152343 D/acc=0.497378 lr=0.000602
INFO:root:[31] T/loss=0.119420 T/acc=0.506527 D/loss=0.150613 D/acc=0.490004 lr=0.000599
```

Gambar 3. Perhitungan *minimum average loss* dari dataset LITOR 4.0.

Metode ListNet memeringkatkan setiap dokumen dengan mengiterasi perhitungan *minimum loss*. Setelah semua dokumen selesai diiterasi, dihasilkan perhitungan *minimum average loss* untuk menunjukkan rerata *loss* yang dihasilkan dari fungsi *listwise loss* yang telah dieksekusi.

Penerapan ListNet memiliki *test accuracy* awal sebesar 0.451238, *test loss* awal sebesar 0.142821, *dev accuracy* awal sebesar 0.487140, dan *dev loss* awal sebesar 0.148315.

IV. KESIMPULAN DAN SARAN

Metode ListNet memiliki skor kinerja yang baik dalam mengolah dan memeringkatkan dokumen dataset LITOR 4.0. Namun, kinerja yang baik tersebut memiliki kekurangan, yaitu durasi proses implementasi yang lebih lama.

REFERENCE

- [1] S. D. N. SILVA, E. S. DE MOURA, P. P. CALADO and A. S. DA SILVA, "Effective Lightweight Learning-to-Rank Method Using Unified Term Impacts," no. 8, pp. 70420 - 70437, 2020.
- [2] M. Modry, *Learning to Rank Algorithms*, 2014.
- [3] M. Pereira, E. Etemad and F. Paulovich, "Iterative Learning to Rank from Explicit Relevance Feedback," p. 698–705, 2020.
- [4] Cao, Zhe, et al. "Learning to rank: from pairwise approach to listwise approach." *Proceedings of the 24th international conference on Machine learning*. 2007.

[5]	M. Trevisiol, "job and talent Engineering," 04 April 2017. [Online]. Available: https://jobandtalent.engineering/learning-to-retrieve-and-rank-intuitive-overview-part-iii-1292f4259315 . [Accessed 30 April 2021].
-----	---

[6]	A. A. Abdillah, H. Murfi and Y. Satria, " <i>Uji Kinerja Learning to Rank Dengan Metode Support Vector</i> ," <i>IndoMS Journal on Industrial and Applied Mathematics</i> , vol. 2, no. 1, p. 16, 2015.
[7]	Tao Qin and Tie-Yan Liu. <i>Introducing LETOR 4.0 Datasets</i> , arXiv preprint arXiv:1306.2597.