

Pairwise Learning to Rank Approach with Neural Networks

Andri Reimondo Tamba
Fakultas Informatika dan Teknik
Elektro
Institut Teknologi Del
Medan, Indonesia
andriraymondo@gmail.com

Enjelin Ida Hutahaean
Fakultas Informatika dan Teknik
Elektro
Institut Teknologi Del
Medan, Indonesia
enjelinhutahaean250@gmail.com line

Martin Sinaga
Fakultas Informatika dan Teknik
Elektro
Institut Teknologi Del
Medan, Indonesia
Martin.halomoan12@gmail.com

Feber Manuel Sianipar
Fakultas Informatika dan Teknik
Elektro
Institut Teknologi Del
Medan, Indonesia
sianipar.feber@gmail.com

Abstract—Pada penelitian ini kami menyajikan pendekatan Pairwise untuk menentukan peringkat berdasarkan neural network, yang disebut DirectRanker, yang menggeneralisasi RankNet architecture. Kami menunjukkan secara matematis bahwa model kami reflexive, antisymmetric, dan transitive yang memungkinkan untuk mensimplikasi training dan meningkatkan performa.

Keywords— Information Retrieval, Machine learning, Learning to rank.

I. INTRODUCTION

Information Retrieval atau temu balik sistem informasi merupakan salah satu aplikasi terpenting dari *machine learning techniques* beberapa tahun terakhir. *Information retrieval* merupakan sistem pencarian untuk menemukan kembali sebuah informasi. Tujuan dari *Information Retrieval* yaitu untuk mengambil informasi atau data serta menampilkan dokumen yang relevan. Jumlah data yang sangat besar yang digunakan pada kehidupan sehari-hari, seperti data penelitian, ekonomi, dan lain-lain. Sehingga hanya perlu untuk mengambil data yang relevan dengan kebutuhan user. Salah satu masalah utama dalam pencarian informasi adalah masalah *learning to rank*. Diberikan kueri q dan satu set dokumen d_1, \dots, d_n seseorang ingin menemukan *ranking* yang memberikan urutan dokumen sesuai dengan relevansinya tergantung pada q (*query*).

Information retrieval akan mempermudah dalam pencarian data yang dibutuhkan. Metode *machine learning* merupakan metode yang paling umum dalam menangani pembelajaran untuk menentukan rank problem. Pada makalah ini, peneliti menggunakan file *artificial neural net*, dalam sepasang dokumen, untuk menemukan dokumen yang lebih relevan. Pendekatan ini dikenal sebagai pairwise ranking approach, yang kemudian dapat digunakan untuk mengurutkan daftar dokumen. *Architecture neural net* yang dipilih akan menampilkan sifat-sifat tertentu yang secara signifikan meningkatkan kinerja dibandingkan dengan pendekatan lain. Pada dasarnya struktur neural net sama dengan yang ada di RankNet. Namun beberapa kendala yang digunakan disana dan menggunakan algoritma pengoptimalan yang lebih modern. Penelitian ini mengarah pada kinerja yang ditingkatkan secara signifikan dan menempatkan pendekatan kami head-to head dengan metode yang lebih canggih. Hal ini sangat luar biasa mengingat struktur model yang relatif sederhana dan waktu pelatihan

serta pengujian yang singkat. Lebih lanjut, peneliti menggunakan formasi yang berbeda untuk mendeskripsikan properti model kami dan menemukan bahwa model tersebut secara inheren, reflektif, antisimetris, dan transitif. Singkatnya, kontribusi dari makalah ini adalah:

1. Kami mengusulkan skema sederhana dan efektif untuk *neural network structures* untuk pairwise ranking, yang disebut DirectRanker yang merupakan generalisasi dari RankNet.
2. Analisis teoritis menunjukkan komponen mana dari struktur jaringan tersebut yang memunculkan propertinya dan apa persyaratan pada data pelatihan untuk membuatnya berfungsi.
3. Menjaga esensi RankNet dan mengoptimalkannya dengan metode modern, eksperimen menunjukkan bahwa, bertentangan dengan kepercayaan umum, metode *Pairwise* masih dapat bersaing dengan metode listwise yang lebih baru dan jauh lebih kompleks.
4. Menunjukkan bahwa metode tersebut dapat digunakan pada prinsipnya untuk klasifikasi. Secara khusus, untuk penemuan substruktur dalam satu kelas.

II. METODOLOGI

A. Pendekatan Pairwise

Pairwise ranking digunakan dengan cara membuat tabel dan membuat komparasi antara satu ide dengan yang lainnya berdasarkan variabel penentu keamanan pelaksanaan. Dalam pendekatan *pairwise*, pasangan dokumen dianggap sebagai *input* untuk sistem pembelajaran. Tujuannya adalah bukan untuk menentukan skor relevansi setiap dokumen dalam kaitannya dengan kueri, tetapi dokumen mana yang lebih relevan daripada yang lain [1].

B. Neural Network

Neural network adalah arsitektur berbasis *Neural Network*, yang pada dasarnya terdiri dari beberapa tahapan, jaringan tersebut dapat dilatih untuk melakukan tugas-tugas terkait klasifikasi teks. Berikut merupakan tahapan dari neural network adalah sebagai berikut [1]:

1. Convolutional layers

Lapisan konvolusional ini merupakan salah satu lapisan penting dalam CNN. Lapisan ini berisi sejumlah matriks kernel. Pada lapisan ini, konvolusional biasanya dilakukan oleh matriks kernel pada input dan keluaran sebagai matriks fitur nilai tambah bias dihasilkan. Bobot kernel dan bias dipelajari dengan menggunakan prosedur pembelajaran karena bobot koneksi dibagi di antara neuron.

2. Pooling layers

Pooling layers merupakan elemen fundamental dari CNN. tujuan utama dari lapisan ini adalah untuk melakukan pengurangan dimensi input, yang mengurangi jumlah yang dihasilkan secara acak variabel sehingga proses analisis data lebih cepat dan sederhana. Subsampling dari lapisan konvolusi keluaran dilakukan oleh lapisan penyatuan dengan menggabungkan elemen tetangga. Fungsi penggabungan maksimal adalah fungsi penggabungan yang paling umum digunakan yang biasanya mengambil nilai maksimum di antara lingkungan lokal.

3. Embedding Layer

Embedding layer ini merupakan lapisan khusus dari CNN yang berfungsi untuk melakukan tugas-tugas terkait klasifikasi teks. Tujuan dari lapisan ini adalah untuk mengubah dokumen teks masukan menjadi format yang tepat dan sesuai untuk CNN. Pada lapisan ini, setiap kata dari dokumen teks masukan diubah menjadi vektor padat ukuran tetap

4. Fully Connected Layer

Fully connected layer merupakan lapisan tersembunyi dari FNN (*Feed-Forward*) dan juga merupakan lapisan terakhir yang paling banyak digunakan pada tahap CNN. Lapisan ini juga bisa sebagai lapisan konvolusi unik yang berisi matriks kernel dengan ukuran 1x1. Jenis lapisan ini adalah anggota grup yang berisi bobot lapisan yang bisa dilatih.

Numerical Weight (Bobot Numerik) pada setiap masukan ditetapkan berdasarkan data historis sebelum proses pelatihan (*training process*). Dalam proses pelatihan, bobot optimal diselesaikan dengan mengurangi *mean square error* (E_m). Berikut merupakan rumus mencari *mean square error* (E_m):

$$E_m = \frac{1}{p_1} * n_0 \sum_{p_1}^{p=p_1} * \sum_{s_1}^{s=n_0} e_s^2(p)$$

Keterangan :

Dimana, n_0 merupakan jumlah neuron dari lapisan keluaran dan e_s^2 merupakan error dari keluaran s_{th} untuk pola p_{th} set pelatihan. Untuk meminimalkan fungsi kesalahan e_m digunakan algoritma *mini-batch stochastic gradient descent* (m-SGD). Pada m-SGD ini koefisien model dan estimasi kesalahan model dilakukan dengan membagi *training set* menjadi sejumlah *batch* kecil. Algoritma m-SGD merupakan

algoritma yang paling banyak digunakan di bidang *deep learning*.

C. DirectRank

Pendekatan DirectRank terhadap pengurutan peringkat dari jenis *Pairwise*, yaitu dibutuhkan dua dokumen dan memutuskan mana yang lebih relevan daripada yang lain. Pendekatan ini memiliki beberapa kesulitan, karena, untuk mencapai peringkat yang konsisten dan unik, model harus menentukan urutan. Dalam pendekatan Direct Rank, penulis menerapkan *quasi order* pada fitur tersebut ruang F sedemikian rupa sehingga penentuan peringkat unik, kecuali untuk dokumen yang setara, yaitu dokumen dengan label relevansi yang sama. *Quasi Order* ini memenuhi persyaratan berikut untuk semua $x, y, z \in F$:

(A) Reflektivitas: $x \leq x$

(B) Anti Simetri: $x \leq y \Rightarrow y \not\leq x$

(C) Transitivitas: $(x \leq y \wedge y \leq z) \Rightarrow x \leq z$

Penulis menerapkan perintah seperti itu menggunakan fungsi peringkat $r: F \times F \rightarrow \mathbb{R}$ dengan mendefinisikan

$$x \leq y: \Leftrightarrow r(x, y) \geq 0. \quad (1)$$

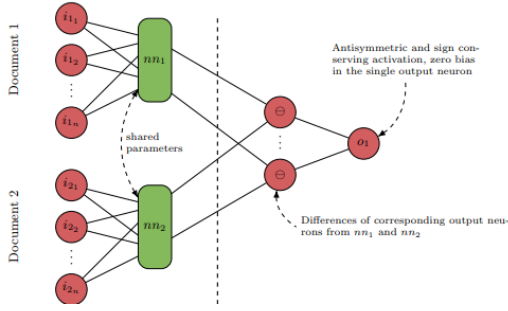
Ketentuan (A) - (C) untuk untuk r :

(I) Reflektivitas: $r(x, x) = 0$ (II) Anti Simetri: $r(x, y) = -r(y, x)$

(III) Transitivitas: $(r(x, y) \geq 0 \wedge r(y, z) \geq 0) \Rightarrow r(x, z) \geq 0$

Dalam kasus ini, r adalah keluaran dari *Neural Network* dengan struktur khusus untuk memenuhi persyaratan yang telah dibuat diatas. Seperti yang ditunjukkan oleh, antisimetri dapat dengan mudah dijamin dalam pendekatan *Neural Network* dengan menghilangkan bias neuron dan memilih fungsi aktivasi antisimetri. Tentu saja, hasilnya hanya akan menjadi antisimetris, jika fitur yang dimasukkan ke jaringan adalah fungsi antisimetris dari dokumen itu sendiri, yaitu jika dua dokumen A dan B harus dibandingkan dengan jaringan, fitur yang diekstrak dari pasangan dokumen harus antisimetris dalam pertukaran A dan B. Hal ini mengarah pada kesulitan pertama karena mengekstrak fitur-fitur yang mengandung cukup informasi tentang dokumen bukanlah hal yang sepele [2]. Model kami menghindari masalah ini dengan mengambil fitur yang diekstrak dari masing-masing dokumen dan mengoptimalkan fitur antisimetrik yang sesuai sebagai bagian dari jaringan itu sendiri selama proses pelatihan. Ini dilakukan dengan menggunakan struktur yang digambarkan pada Gambar 1.

dan aktivasi antisimetris dan konservasi tanda.



Gambar 1. Skema DirectRanker

Fitur yang sesuai dari dua dokumen dimasukkan ke dalam dua subnet nn 1 dan nn 2, masing-masing. Jaringan ini dapat berupa struktur yang berubah-ubah, namun harus identik, yaitu berbagi struktur dan parameter yang sama seperti bobot, bias, aktivasi, dll. Perbedaan keluaran subnet dimasukkan ke dalam subnet ketiga, yang selanjutnya hanya terdiri dari satu neuron output dengan aktivasi antisimetrik dan tanpa bias, mewakili fungsi yang didefinisikan di atas r . Dengan teorema berikut penulis menunjukkan bahwa jaringan ini memenuhi kondisi (I) sampai (III):

Teorema 1. Membiarkan f menjadi keluaran dari jaringan neural arbitrer yang mengambil sebagai masukan vektor fitur $x \in F$ dan mengembalikan nilai $f(x) \in R^n$. Membiarkan H_1 menjadi neuron tunggal dengan antisimetri dan fungsi aktivasi pelestarian tanda dan tanpa bias pengambilan R^n -masukan berharga. Neural network $(f(x) - f(y))$ untuk $x, y \in F$ kemudian memenuhi (I) sampai (III). dapat dikenakan dalam bentuk kondisi berikut:

Fungsi aktivasi neuron keluaran menjadi $\tau: R \rightarrow R$ dengan $\tau(-x) = -\tau(x)$ dan tanda tangan $(\tau(x)) = \text{tanda}(x)$ seperti yang dipersyaratkan.

(I) Jika (II) terpenuhi, maka (I) mudah, karena $r(x, x) = -r(x, x) \forall x \in F \Rightarrow r(x, x) = 0$.

(II) Dua jaringan nn 1 dan nn 2 identik (karena memiliki parameter yang sama). Karenanya, dapat diterapkan fungsi yang sama $f: F \rightarrow R^n$.

keluaran dari jaringan untuk dua vektor masukan $x, y \in F$ kemudian diberikan oleh:

$$r(x, y) = \tau[w(f(x) - f(y))] = \tau[wf(x) - wf(y)] =: \tau[g(x) - g(y)], \quad (2)$$

dimana w adalah vektor bobot untuk neuron keluaran dan $g: F \rightarrow R$. Ini antisimetris untuk x dan y , sehingga memenuhi kondisi kedua (II).

(III) Selanjutnya $x, y, z \in F$, $r(x, y) \geq 0$, $r(y, z) \geq 0$, dan g didefinisikan sebagai 2. Lalu τ untuk mempertahankan tanda input, yaitu $\tau(x) \geq 0 \Leftrightarrow x \geq 0$, $g(x) \geq g(y)$ dan $g(y) \geq g(z)$, yaitu :

$$r(x, z) = \tau[g(x) - g(z)] = \tau[g(x) - g(y) + g(y) - g(z)] = \tau[g(x) - g(y)] + \tau[g(y) - g(z)] \geq 0 + 0 = 0.$$

Jadi, r bersifat transitif dan (III) terpenuhi.

Hal ini menawarkan beberapa keuntungan selama fase pelatihan jaringan untuk perbedaan kelas relevansi yang berbeda:

- (i) Karena transitivitas, tidak perlu membandingkan korelevansi antar dokumen. Asalkan setiap dokumen dilatih setidaknya dengan dokumen dari kelas relevansi yang sesuai, model secara implisit dapat dilatih untuk semua kombinasi, karena semua kelas terwakili dalam data pelatihan.
- (ii) Meskipun tampaknya masuk akal untuk melatih model sedemikian rupa sehingga mampu untuk memprediksi kesetaraan dua dokumen berbeda dari kelas relevansi yang sama, model sebenarnya dibatasi saat melakukannya.

DirectRanker adalah generalisasi dari model RankNet, yang setara dengan DirectRanker, jika $s(x) = \tanh x$, dan jika biaya cross entropy, dan pengoptimal penurunan gradien. Untuk mempermudah, penulis akan memilih fungsi aktivasi yang akan digunakan $\tau \equiv \text{Id}$ karena fungsi aktivasi tidak mengubah urutan, jika τ adalah tanda konservasi umum. Berikut ini, penulis mencoba untuk menempatkan DirectRanker pada dasar yang lebih baik dengan menganalisis beberapa kasus di mana DirectRanker dapat memperkirakan pesanan (mengingat cukup kompleksitas dan sampel pelatihan [3]).

III. EKSPERIMEN

Pada penelitian ini kami menggunakan Microsoft Learning to Rank Datasets MQ2008 (LETOR) adalah kumpulan data standar yang paling umum digunakan untuk tolok ukur belajar menentukan peringkat model. Setiap contoh dalam kumpulan data sesuai dengan pasangan dokumen kueri yang dicirikan oleh 136 fitur numerik. Selain itu, label relevansi dari 0 (tidak relevan) hingga 4 (sangat relevan) menunjukkan relevansi dokumen yang diberikan sehubungan dengan kueri. Pada bagian ini, kami menyajikan hasil dari percobaan pada algoritma yang kami usulkan yaitu Pertama, kami membandingkan model kami dengan model yang umum digunakan pada bagian. Perbandingan dengan Ranker lainnya Pada dataset MQ2008, perbedaan antara model tidak signifikan ($0,54 \sigma$ perbedaan dalam NDCG @ 10 antara algoritma berperforma terbaik dan terburuk pada dataset MQ2008) membuat eksperimen pada set ini tidak meyakinkan. Penting untuk dicatat bahwa untuk LambdaMart model secara eksplisit didorong pada NDCG @ 10 dan MAP masing-masing untuk dua nilai kinerja sementara DirectRanker menggunakan fungsi biaya yang tidak tergantung pada matrik evaluasi..

Tabel 1. Perbandingan kinerja untuk ranker yang berbeda pada beberapa set data Letor. Nilai untuk ES-Rank, IESR-

Rank dan IESVM-Rank diambil dari Nilai-nilai ini ditandai dengan huruf miring. LambdaMart didorong menggunakan matrik evaluasi yang sesuai selama pelatihan.

Algoritma	NDCG@W	P@BRETAK	MQ2 '008		MQ2 '007	
			NDCG@	P@A	NDCG@	P@A
DirectRanker	0.440 (4)	0.365 (3)	0,720 (12)	0,636 (11)	0,540 (10)	0,534 (9)
RankNet	0,157 (3)	0,195 (2)	0,716 (11)	0,642 (10)	0,525 (11)	0,525 (7)
ListNet	0,157 (3)	0,192 (2)	0,719 (10)	0,647 (6)	0,526 (10)	0,525 (9)
LambdaMart	0.476 (3)	0,366 (3)	0,723 (7)	0,624 (6)	0,531 (12)	0,510 (11)
AdaRank	0,400 (16)	0,322 (10)	0,722 (10)	0,653 (9)	0,526 (10)	0,527 (10)
ES-Rank	0.382	0.570	0.507	0.483	0.451	0.470
Peringkat IESR	0.415	0.603	0.517	0.494	0.455	0.473
Peringkat IESVM	0.224	0.457	0.498	0.473	0.436	0.456

Untuk mendemonstrasikan kesederhanaan DirectRanker, kami menyajikan eksperimen pada runtime untuk pelatihan model pada Tabel 2. Membandingkan rata-rata waktu *running* lima kali lipat dari kumpulan data MQ2008. Nilai-nilai dilatih dengan menggunakan mesin yang disebutkan dalam teks. Nilai-nilai dengan † dilatih menggunakan RankLib. Untuk RankNet dua implementasi digunakan. Satu dengan aliran Tensor dan satu dari RankLib

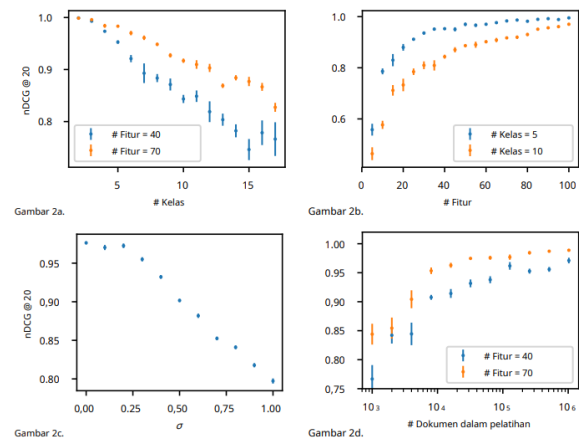
Algorithm	time in seconds
DirectRanker*	151.94(41)
RankNet*	142.27(69)
RankNet*†	2215(351)
AdaRank*†	1261(50)
LambdaMART*†	2664(234)
ListNet*†	3947(481)
ES-Rank‡	1800
IESR-Rank‡	1957
IESVM-Rank‡	34 209

telah dilakukan pada Intel © R Inti TM i7-6850K CPU @ 3.60GHz menggunakan cara di atas menyebutkan dataset MQ2008 rata-rata lima kali lebih cepat. Model kami dilatih menggunakan aliran Tensor , berlawanan dengan implementasi lainnya [4]. Hal ini membuat perbandingan waktu proses menjadi sulit, namun, kami juga menerapkan ulang RankNet menggunakan aliran Tensor dengan cara yang sama seperti model kami. Di sini dapat dilihat bahwa runtime DirectRanker dan RankNet memiliki urutan yang sama. Jadi, kami tidak meningkatkan waktu pelatihan model tetapi hanya kinerja. Di sisi lain, waktu pelatihan mengalahkan semua model lainnya.

(ii) Sensitivitas pada Properti Set Data

Dengan pengujian berikut kami membahas bagaimana DirectRanker bekerja dalam keadaan yang berbeda. Tes dilakukan seperti yang dijelaskan di detik. 4.3. Kinerja DirectRanker telah diuji untuk berbagai jumlah kelas relevansi (Gbr. 2a), fitur (Gbr. 2b), untuk variasi kebisingan pada label kelas (Gbr. 2c), dan set pelatihan ukuran yang berbeda (Gbr. 2d) . Pengujian menunjukkan bahwa, dengan data yang cukup, model kami mampu menangani beragam kumpulan data. Ini terutama menunjukkan bahwa DirectRanker dapat menangani banyak kelas relevansi seperti yang ditunjukkan pada Gambar 2a. Seperti yang diharapkan, kinerja menurun dengan jumlah kelas relevansi. Namun, efek

ini dapat diatasi dengan meningkatkan ukuran set pelatihan (lihat Gambar 2d) atau jumlah fitur (Gbr. 2b). Selain itu, Gambar. 2c menunjukkan ketahanan DirectRanker terhadap *noise* pada kelas relevansi. Hingga beberapa kebisingan kecil (sekitar 5% mislabeling, mis $\sigma = 0,25$), kinerja menurun hanya sedikit, tetapi turun secara signifikan untuk kebisingan yang lebih besar. Namun, bahkan dengan 50% dari dokumen yang salah label (mis $\sigma = 0.75$) NDCG @ 20 tidak turun di bawah



Gambar 2. Plot yang menggambarkan sensitivitas kinerja DirectRanker pada properti data tertentu, dievaluasi pada data sintetis (bagian 4.3). (Gbr. 2a) Ketergantungan pada jumlah kelas relevansi (10 5 dokumen dalam set pelatihan). (Gbr. 2b) Ketergantungan pada jumlah fitur (10 5 dokumen dalam set pelatihan, lima kelas relevansi). (Gbr. 2c) Kinerja DirectRanker dengan tingkat kebisingan yang berbeda pada label kelas dengan 5 kelas dan 70 fitur. (Gbr. 2d) Ketergantungan pada ukuran set pelatihan (lima kelas relevans). Hal ini menunjukkan bahwa temuan teoritis di detik. 3 agar data ideal tetap valid.

IV. Kesimpulan

Skema untuk struktur jaringan yang diusulkan dan dianalisis dalam makalah ini adalah generalisasi RankNet: Kami menunjukkan properti komponen RankNet mana yang penting untuk menghasilkan yang menguntungkan dan dengan demikian dapat melakukan peningkatan kinerja. Ternyata, hanya beberapa asumsi tentang struktur jaringan yang diperlukan untuk dapat mempelajari urutan instance. Persyaratan pada data untuk pelatihan juga minimal: Metode ini dapat diterapkan pada data *diskrit* dan *kontinu*, dan dapat digunakan untuk jadwal pelatihan yang disederhanakan dengan perbandingan kelas yang berdekatan (atau pasangan kelas relevansi lain yang relevan) saja. Hasil teoritis menjelaskan beberapa alasan mengapa ini terjadi. Eksperimen menegaskan hal ini dan menunjukkan bahwa skema tersebut memberikan kinerja yang sangat baik juga pada data dunia nyata, di mana kita dapat berasumsi bahwa contoh diberikan label dengan probabilitas tertentu. Dalam

banyak perbandingan baru-baru ini, RankNet terbukti menunjukkan kinerja yang lebih rendah, yang mengarah pada kesimpulan bahwa pendekatan *listwise* lebih disukai daripada pendekatan *pairwise*. Melihat hasil eksperimen pada dataset LETOR pada penelitian ini, mungkin ada alasan untuk mempertimbangkan kembali pandangan tersebut. Namun, mungkin menarik untuk mengadaptasi ide LambdaRank dan LambdaMART untuk pengoptimalan *listwise* ke model DirectRanker. Selain itu, pendekatan yang sederhana dan transparan dapat menyamai atau mengungguli kinerja model yang lebih baru dan jauh lebih kompleks, seperti ES-Rank dan sejenisnya. Eksperimen dengan data sintetis menunjukkan bagaimana kinerja dapat menurun ketika diberikan lebih banyak kelas relevansi, lebih sedikit fitur atau lebih sedikit contoh pelatihan. Namun, hasil ini juga menunjukkan bagaimana hilangnya kinerja dapat dikompensasi oleh faktor-faktor lain. Selain peringkat standar, kami menyarankan agar DirectRanker dapat digunakan untuk klasifikasi juga. Tes pertama menunjukkan hasil yang menjanjikan. Penyelidikan yang lebih sistematis tentang ini adalah subjek pekerjaan masa depan.

REFERENCES

- [1] S. C. H.-C. K. S. Aich, "Convolutional neural network-based model for web-based text classification," *International Journal of Electrical and Computer Engineering (IJECE)*, " Vol. 9, No. 6, pp. 5185-5186, 2019.
- [2] L. P. M. M. Rigutin, "Pendekatan jaringan saraf untuk mempelajari peringkat objek dalam Konferensi Internasional Tentang Jaringan Saraf Tiruan," pp. 899-908, 2008.
- [3] C. S. R. L. D. Burger, "Belajar menentukan peringkat menggunakan penurunan gradien. Dalam: Prosiding Konferensi internasional 22Nd tentang Pembelajaran Mesin.," vol. V, pp. 89-96, 2005.
- [4] M. Abadi, "TensorFlow: Machine learning skala besar pada sistem heterogen," 2015.