Perbandingan Sistem Temu Balik Metode BM25 dan Metode Neural Network dengan BERT Preprocessing pada Dataset Cranfield

Andreas Aditya[#1], Oviliani Yenti Yuliana[*2], Hery Heryanto[#3]
[#]*Departemen Informatika, Institut Teknologi Harapan Bangsa*
Jalan Dipatiukur No. 80-84, Bandung, Indonesia
[1]andreas.aditya251299@gmail.com
[3]hery_heryanto@ithb.ac.id
[*]*Universitas Kristen Petra*
Jalan Siwalankerto No. 121-131, Surabaya, Indonesia
[2]oviliani@petra.ac.id

Abstract*—The development of information retrieval systems has led to various machine learning methods that are able to learn a document's context to find the relevant information. The BERT model used by Google currently is a great model for information retrieval systems on the internet. This study compares neural network model with BERT preprocessing and the BM25 method on Cranfield dataset. The BM25 method has 2 indicators, namely b and k, each giving weight for the length of the document and the frequency of occurrence of words. The indicators used for the neural network method with BERT preprocessing are the size of the document subsegment, learning rate, neural network layer size, and the method of combining relevance values from subsegments. The dataset used in this research is the Cranfield dataset which contains 1400 documents and 225 queries. Based on the results, the BM25 method gives an nDCG@20 value of 0.5054 and the neural network method with BERT preprocessing produces an nDCG@20 value of 0.5520. Therefore, BERT gives more relevant results early when searching.

***Keywords*—Information Retrieval, Deep Learning,* BM25, BERT, NLP.**

Abstrak*—Perkembangan sistem temu balik informasi memunculkan berbagai metode *machine learning* yang menggunakan konteks dari dokumen untuk menemukan informasi yang dicari. Model BERT yang digunakan oleh Google merupakan model terbaik dari sistem pencarian situs di internet. Penelitian ini membandingkan BERT yang digunakan sebagai komponen preprocessing dokumen dan *query* untuk model *neural network* dengan metode BM25 pada *dataset Cranfield*. Metode BM25 memiliki 2 indikator yaitu b dan k yang masing-masing berfokus pada *weight* untuk panjang dokumen dan frekuensi kemunculan kata. Indikator yang digunakan untuk metode *neural network* dengan BERT *preprocessing* adalah ukuran *subsegment* sebuah dokumen, *learning rate*, ukuran *layer neural network*, dan metode penggabungan nilai relevansi *subsegment*. *Dataset* yang digunakan dalam penelitian ini adalah *dataset Cranfield* yang berisi 1398 dokumen dan 225 *query*. Berdasarkan pengujian semua indikator, metode BM25 memberikan nilai nDCG@20 sebesar 0.5054 dan metode *neural network* dengan BERT *preprocessing* menghasilkan nilai nDCG@20 sebesar 0.5520. Sehingga, BERT memberikan hasil yang lebih relevan lebih awal dalam pencarian.

***Kata Kunci*—Information Retrieval, Deep Learning,* BM25, BERT, NLP.**

I. PENDAHULUAN

Sistem temu balik informasi (STBI) digunakan untuk mencari informasi di *internet* pada era modern. STBI mempelajari konteks dari berbagai situs sehingga menampilkan hasil yang relevan [1]. STBI modern membutuhkan sumber daya yang besar untuk pembuatan dan penggunaan *search engine* [2]. Dampak dari kebutuhan sumber daya yang besar adalah sistem dengan sumber daya yang terbatas kesulitan untuk menggunakan STBI modern. Penelitian ini membandingkan metode STBI statistik modern BM25 [3] dan metode *machine learning* menggunakan model BERT sebagai komponen *preprocessing* [4].

Dalam beberapa waktu terakhir, Google telah menggunakan model BERT sebagai metode utama dalam STBI yang digunakan [4]. Penggunaan model BERT memberikan hasil yang lebih baik dengan *query* bahasa natural dengan kemampuan BERT untuk mempelajari konteks menggunakan *attention* [2]. Seperti model *neural network* lain, model BERT membutuhkan sumber daya yang lebih besar dari metode probabilistik. Untuk menghemat penggunaan sumber daya tersebut, BERT dipakai sebagai komponen *preprocessing* karena BERT menghasilkan *vector encoding* dari teks yang didapat.

Metode BM25 merupakan metode probabilistik dari pengembangan metode TF-IDF. Keunggulan metode BM25 adalah penggunaan sumber daya yang rendah. Kekurangannya, metode BM25 tidak mempelajari konteks sehingga hasil yang diberikan bergantung pada frekuensi kemunculan kata-kata dan panjang dokumen [3].

Kedua metode yang diperlihatkan memiliki karakteristik yang sangat berbeda. Terdapat banyak penelitian yang memperlihatkan seberapa baik hasil metode BERT dalam STBI [1], [5]–[7]. Penelitian-penelitian tersebut menggunakan nDCG@20 sebagai nilai ukur performa metode. Penelitian yang dilaksanakan hendak melihat performa dari kedua

metode terhadap sebuah sistem yang memiliki keterbatasan sumber daya. Penelitian yang umumnya dilakukan terhadap metode STBI berfokus pada hasil dokumen yang didapat oleh metode. Namun, dalam penelitian ini waktu pencarian diperhitungkan agar dampak penggunaan kedua metode pada sistem yang berukuran kecil dapat terlihat.

Penelitian-penelitian yang telah dilaksanakan sebelumnya tidak melihat keterbatasan sebuah sistem komputer. Walau BERT tidak membutuhkan sumber daya sebanyak model berbasis LSTM seperti ELMO, BERT merupakan sebuah model besar yang membutuhkan sumber daya yang tinggi [2]. Untuk menanggulangi penggunaan sumber daya yang tinggi pada saat STBI dijalankan, BERT digunakan hanya untuk *preprocessing* data dokumen. Data yang telah diolah tersebut yang dimasukkan ke dalam model *neural network* saat *runtime*. Selain itu, penelitian sebelumnya yang membandingkan BM25 dan BERT, tidak melakukan *tuning* terhadap parameter BM25 sehingga hasil metode BM25 masih dapat ditingkatkan [5].

Dataset yang digunakan dalam penelitian ini merupakan *dataset cranfield collection*. *Dataset* tersebut merupakan *dataset* yang tergolong kecil untuk STBI. Namun, karena penelitian yang dilakukan membatasi sumber daya dalam penelitian, dataset tersebut tepat untuk penelitian [8].

Penelitian bertujuan untuk melihat waktu pencarian dan hasil dokumen yang didapat untuk melihat viabilitas kedua metode pada sistem yang memiliki keterbatasan sumber daya. Berdasarkan latar belakang masalah dan tujuan penelitian yang telah dipaparkan, dirumuskan beberapa masalah yang akan dibahas dalam penelitian. Rumusan masalah tersebut adalah sebagai berikut:

- Berapa nilai b dan k yang tepat untuk STBI menggunakan metode BM25 pada *dataset Cranfield*?
- Berapa ukuran subsegment, nilai *learning rate*, jumlah *neuron* pada *layer*, dan metode penggabungan nilai relevansi yang optimal untuk STBI berbasis *neural network* dengan BERT *preprocessing* pada *dataset Cranfield*?
- Apakah STBI berbasis *neural network* dengan BERT *preprocessing* memberikan nilai nDCG pada posisi 20 yang lebih baik dari metode BM25 pada *dataset Cranfield*?

Pembahasan penelitian ini diorganisir sebagai berikut. Bab 2 memperlihatkan metodologi dan sistem yang digunakan. Bab 3 membahas hasil dan analisis dari kedua metode. Bab 4 berisi kesimpulan dari penelitian.

II. Metodologi

A. BM25

BM25 merupakan metode statistik pemberian peringkat yang digunakan dalam sistem temu balik. Metode tersebut merupakan pengembangan dari metode TF-IDF. Metode BM25 memiliki parameter yang dapat dapat diubah agar memberikan hasil yang lebih baik. Parameter tersebut mengatur *weight* pada panjang sebuah dokumen dan jumlah kemunculan sebuah kata [3].

Implementasi BM25 dimulai dengan membersihkan dokumen dari *dataset* dengan membuang *stop words*, *lemmatizing*, dan *stemming*. Kemudian, dokumen-dokumen tersebut di-tokenisasi dan dihitung atribut-atributnya [8], [9]. Atribut-atribut tersebut digunakan oleh metode BM25 untuk menghitung nilai relevansi antara dokumen dan *query*. Dokumen-dokumen tersebut akan diurutkan berdasarkan nilai relevansi antara dokumen dan *query* [3]. Hasil urutan tersebut dievaluasi menggunakan nilai nDCG@20.

Variabel-variabel yang diuji dalam penelitian ini merupakan variabel b dan k dari metode BM25. Variabel b dalam metode BM25 *weight* untuk panjang dokumen [8]. Sementara variabel k memberikan *weight* pada kemunculan token. Nilai b yang diuji adalah 0,0; 0,2; 0,4; 0,6; 0,8; dan 1,0. Nilai k yang diuji adalah 0,0; 0,2; 0,4; 0,6; 0,8; 1,0; 1,2; 1,4; 1,6; 1,8; 2,0; 2,2; 2,4; 2,6; 2,8; 3,0; 3,2; dan 3,4. .

B. BERT

BERT merupakan pengembangan dari arsitektur *Transformers* yang berfokus pada *attention* [2]. Secara spesifik, BERT dibuat dari komponen *encoder* dari arsitektur *Transformers*. Pengembangan BERT menghasilkan sebuah arsitektur yang dapat membuat *text encoding* dari dua dokumen sekaligus. Model BERT digunakan sebagai komponen *preprocessing* agar waktu sistem temu balik menjadi lebih cepat [4].

Dokumen dari dataset akan dipecah menjadi beberapa *subsegment*. Setelah itu, semua *subsegment* dokumen dan *query* akan dipasangkan untuk diolah dengan model BERT [1]. Model BERT yang digunakan merupakan model *pre-trained*. Hasil dari model BERT akan disimpan dan dimasukkan ke dalam jaringan syaraf tiruan untuk menemukan nilai relevansi *subsegment* dokumen dan *query* [10]. Kemudian, nilai relevansi *subsegment* dokumen dan *query* digabungkan berdasarkan dokumen untuk mengkalkulasi nilai relevansi antara dokumen dan *query*. Nilai tersebut digunakan untuk mengurutkan dokumen dalam sistem temu balik. Hasil urutan dokumen dievaluasi menggunakan nilai nDCG@20 [1].

Variabel-variabel yang diuji dalam penelitian ini adalah ukuran *subsegment* dokumen, ukuran *layer* model *neural network*, *learning rate*, dan metode penggabungan nilai relevansi. Ukuran *subsegment* yang diuji adalah 50, 100, 150, dan 200. Ukuran *layer neural network* yang diuji adalah 256, 512, dan 768. *Learning rate* yang diuji adalah 0,00001; 0,0001; 0,001; 0,002; dan 0,003. Metode penggabungan yang diuji adalah metode *first*, *max*, *mean*, *geometric mean*, dan *harmonic mean*. Metode *first* melihat nilai relevansi *subsegment* pertama dari dokumen. Metode *max* melihat nilai relevansi terbesar dari *subsegment* dokumen. Metode *mean*, *harmonic mean*, dan *geometric mean* menghitung rata-rata, rata-rata harmonis, dan rata-rata geometris dari *subsegment* dokumen.

C. Perancangan Sistem

1) Kerangka Pemikiran

Gambar 1 memperlihatkan *flowchart* dari proses global penelitian. *Dataset* yang terdiri dari *query*, dokumen, dan nilai relevansi digunakan oleh kedua metode. Untuk metode *neural*

network dengan BERT *preprocessing*, *query* dipecah menjadi *train set* dan *test set*. Metode BM25 akan langsung menggunakan *test set* karena metode tersebut tidak memerlukan *training*. Metode *neural network* akan memecah dokumen menjadi *subsegment* dan memasangkan setiap *subsegment* yang dihasilkan dengan *query*. Pasangan *query* dan *subsegment* akan diolah menggunakan model BERT sebagai *preprocessing* data. Data hasil *preprocessing* dari *query training* akan digunakan untuk melatih model *neural network* dan data dari *test query* digunakan untuk menguji model. Model diuji dengan cara memberi peringkat pada dokumen dan mengurutkan dokumen berdasarkan peringkat yang didapat. Urutan dokumen dibandingkan dengan nilai relevansi dokumen untuk menghitung nilai nDCG@20. Metode BM25 melakukan *preprocessing* dan memberikan peringkat terhadap dokumen. Dari peringkat tersebut dokumen diurutkan dan nilai nDCG@20 dihitung dengan membandingkan hasil urutan dengan nilai relevansi dari *dataset*. Terakhir, nilai nDCG@20 dari kedua metode dibandingkan dan dianalisa.

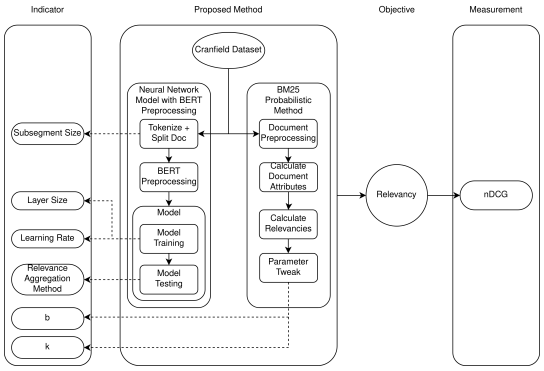

Fig. 1. Kerangka Pemikiran

2) Proses Global

Gambar 2 memperlihatkan kedua metode yang diuji dalam penelitian yaitu metode BM25 dan metode *neural network* dengan BERT *preprocessing*. Kedua metode akan melakukan *preprocessing* terhadap dokumen dari *dataset cranfield*. Metode *neural network* akan memecah dokumen menjadi beberapa *subsegment* dengan indikator ukuran *subsegment*. *Subsegment* yang dihasilkan dipasangkan dengan *query* dan diolah datanya menggunakan model BERT. Hasil *preprocessing* tersebut dimasukkan ke dalam model *neural network* baru yang menghasilkan nilai relevansi antara *subsegment* dan *query*. Model dibuat dengan indikator *layer size* dan *learning rate*. Untuk mendapatkan hasil relevansi antara dokumen dan *query*, nilai relevansi *subsegment* dan query digabungkan dengan indikator metode penggabungan nilai relevansi. Untuk metode BM25, dokumen yang telah melalui tahap *preprocessing* dihitung nilai atributnya dan digunakan untuk menghitung nilai relevansi antara dokumen dan *query*. Indikator b dan k digunakan untuk meningkatkan hasil yang diberikan metode BM25. Objektif dari kedua metode adalah relevansi yang diukur menggunakan nDCG@20.

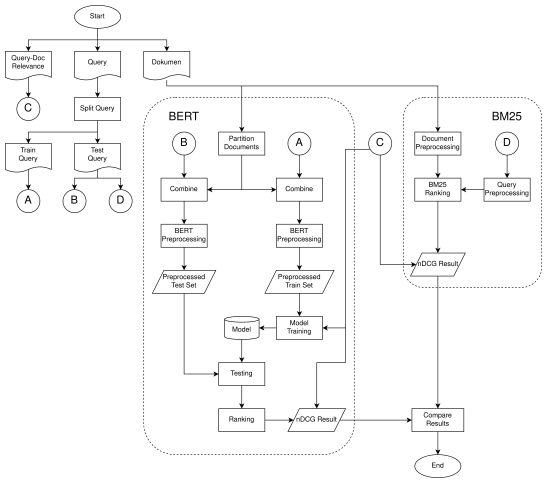

Fig. 2. Urutan Proses Global

3) Dataset Cranfield Collection

Cranfield collection merupakan *dataset* yang terdiri dari 225 *query* dan 1398 dokumen. Dataset memiliki nilai relevansi antara dokumen dan *query* yang bernilai dari 1 sampai 4. Dokumen yang tidak memiliki nilai relevansi terhadap sebuah *query* artinya tidak relevan terhadap *query* tersebut. Dokumen-dokumen dalam *dataset* terdiri dari judul, nama penulis, dan abstrak dari jurnal aerodinamika dalam bahasa inggris. Dalam penelitian ini, informasi yang digunakan adalah judul dan abstrak dari jurnal tersebut.

III. Hasil dan Pembahasan

A. Hasil nDCG@20

1) BERT

Hasil nDCG@20 terbaik dari pengujian metode *neural network* dengan BERT *preprocessing* adalah 0,5520. Hasil terbaik tersebut didapat dengan ukuran *subsegment* 50, ukuran *layer* 256, *learning rate* 0,001, dan metode penggabungan nilai relevansi *mean*. *Boxplot* dari hasil pengujian yang dikelompokkan berdasarkan variabel diperlihatkan pada Gambar 3, 4, 5, dan 6.

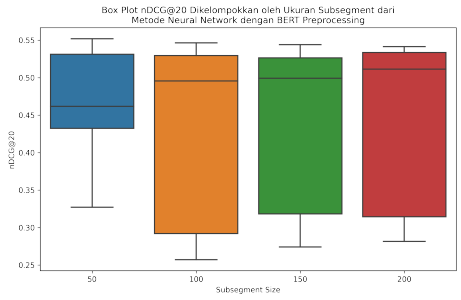

Fig. 3. *Boxplot* Hasil Pengujian Berdasarkan *Subsegment*

Hasil pengujian berdasarkan ukuran subsegment memperlihatkan bahwa *subsegment* dengan ukuran 50 memiliki persebaran data yang lebih kecil sehingga memberikan hasil yang

baik. Kekurangan ukuran *subsegment* 50 adalah median yang lebih rendah sehingga ukuran *subsegment* lain konsisten memberikan hasil yang lebih baik. Setelah analisa lebih lanjut, ukuran *subsegment* kecil memberikan jumlah data point yang lebih banyak sehingga model dapat dilatih dengan lebih baik.

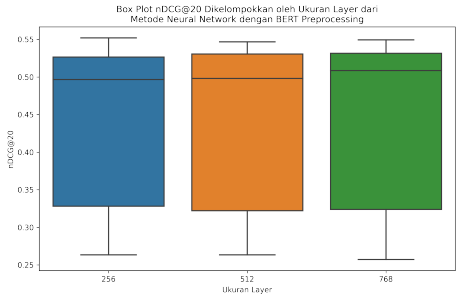

Fig. 4. *Boxplot* Hasil Pengujian Berdasarkan Ukuran *Layer*

Hasil pengujian berdasarkan ukuran *layer* tidak memperlihatkan perbedaan yang signifikan antara ukuran *layer* yang diuji. Satu-satunya perbedaan yang cukup signifikan adalah ukuran layer 768 memiliki median yang lebih tinggi sehingga konsisten memberikan hasil yang lebih baik.

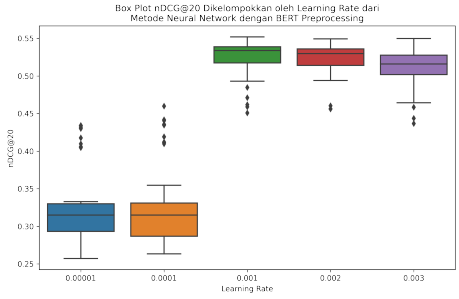

Fig. 5. *Boxplot* Hasil Pengujian Berdasarkan Nilai *Learning Rate*

Hasil pengujian berdasarkan nilai *learning rate* memperlihatkan hasil yang beragam. *Learning rate* 0,00001 dan 0,0001 memberikan hasil yang relatif buruk dengan beberapa *outlier* yang lebih baik. *Learning rate* 0,001, 0,002, dan 0,003 memberikan hasil yang relatif baik dengan *outlier* yang lebih buruk. Setelah analisa lebih lanjut, *learning rate* berukuran kecil memiliki *outlier* baik apabila ukuran *subsegment* kecil yang mengakibatkan jumlah data point pada latihan lebih banyak. Sementara *learning rate* yang lebih besar memiliki *outlier* buruk apabila metode penggabungan relevansi *subsegment* yang digunakan adalah first, memberi nilai relevansi yang buruk apabila komponen relevan dari dokumen tidak di bagian awal.

Hasil pengujian berdasarkan metode penggabungan nilai relevansi memperlihatkan hasil yang beragam. Metode *first* dan *max* memiliki median yang lebih rendah daripada metode lain sehingga metode lain konsisten memberikan nilai yang lebih baik. Metode *mean* memiliki median yang terbesar

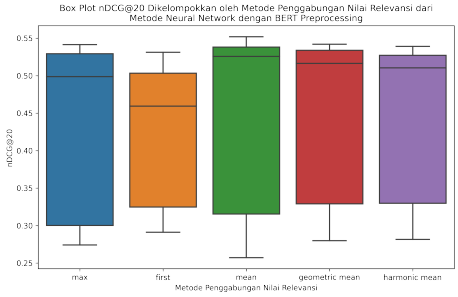

Fig. 6. *Boxplot* Hasil Pengujian Berdasarkan Metode Penggabungan Nilai Relevansi

dan lebih baik dari metode *geometric mean* dan *harmonic mean*. Setelah analisa lebih lanjut, metode *first* dan *max* lebih buruk karena hanya menilai relevansi sebuah dokumen dari satu *subsegment* saja. Sementara, metode *geometric mean* dan *harmonic mean* lebih buruk dari metode *mean* karena memberikan penalti terhadap subsegment yang memiliki nilai yang terlalu rendah. Sehingga, apabila ada *subsegment* yang tidak relevan, *subsegment* tersebut menjatuhkan nilai relevansi.

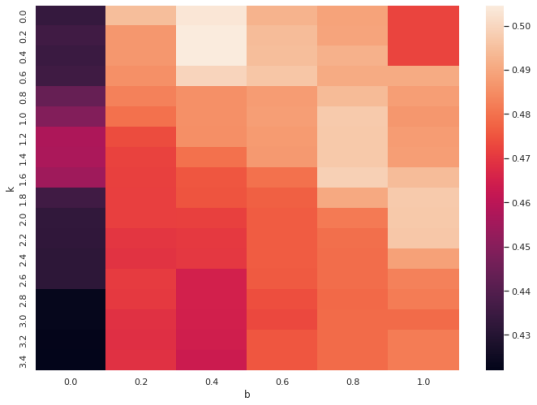

Fig. 7. *Heatmap* Hasil Pengujian BM25

2) BM25

Hasil nDCG@20 terbaik dari pengujian metode BM25 adalah 0,5045. Hasil tersebut didapat dengan parameter k bernilai 0,4 dan b bernilai 0,4. *Heatmap* dari hasil pengujian tertera pada Gambar 7. Dari hasil tersebut, analisa yang dilakukan memperlihatkan bahwa terdapat tarik ulur antara kedua variabel. Tarik ulur yang terpapar adalah sebagai berikut.

- Pada saat variabel b bernilai 0,4, nilai variabel k yang optimal berada di rentang 0,0 $\sim$ 0,6.
- Pada saat variabel b bernilai 0,6, nilai variabel k yang optimal berada di rentang 0,4 $\sim$ 0,6.
- Pada saat variabel b bernilai 0,8, nilai variabel k yang optimal berada di rentang 1,0 $\sim$ 1,4.
- Pada saat variabel b bernilai 1,0, nilai variabel k yang optimal berada di rentang 1,8 $\sim$ 2,2.

TABEL I
WAKTU PENGUJIAN 45 *Query* BERDASARKAN METODE

Waktu	BM25	Metode Neural Network dengan BERT Preprocessing
Min	0,5309 detik	1491,9510 detik
Max	0,5759 detik	1858,1199 detik
Mean	,.5409 detik	1784,2907 detik

B. Waktu

Waktu pengujian minimum, maksimum, dan ratarata metode *neural network* dengan BERT *preprocessing* dipaparkan pada Tabel I. Rata-rata waktu untuk menguji seluruh *test set* adalah 1784,2907 detik. Sehingga, rata-rata waktu yang dibutuhkan untuk satu *query* adalah 39,6509 detik.

Waktu pengujian minimum, maksimum, dan ratarata metode BM25 dapat dipaparkan pada Tabel I Dapat dilihat bahwa waktu yang digunakan tidak memperlihatkan perubahan yang signifikan. Rata-rata waktu untuk menguji seluruh *test set* adalah 0,5409 detik. Sehingga, rata-rata waktu yang dibutuhkan untuk satu *query* adalah 0,0120 detik.

Penggunaan waktu dari kedua metode sangat berbeda secara signifikan. Metode BM25 membutuhkan waktu yang sangat singkat sehingga metode tersebut dapat melakukan pencarian lebih dari 3000 kali sebelum metode *neural network* dengan BERT *preprocessing* selesai melakukan sebuah pencarian.

IV. SIMPULAN

Penelitian ini dilakukan untuk membandingkan metode BM25 dan metode *neural network* dengan BERT *preprocessing* pada *dataset cranfield*. Berdasarkan pengujian yang dilaksanakan, metode BM25 memberikan hasil nDCG@20 0,5045 pada kondisi terbaik, yaitu dengan parameter b bernilai 0,4 dan k bernilai 0,4. Metode *neural network* dengan BERT *preprocessing* memberikan hasil nDCG@20 0,5520 pada kondisi terbaik, yaitu ukuran *subsegment* 50, ukuran layer 256, nilai *learning rate* 0,001, dan metode penggabungan nilai relevansi *mean*. Metode *neural network* menghasilkan nilai nDCG@20 (0,5520) lebih baik dari metode BM25 (0,5045). Hal ini memperlihatkan bahwa metode *neural network* dengan BERT *preprocessing* memberikan hasil yang lebih baik dari metode BM25. Namun, dalam sistem yang terbatas, waktu pencarian sangat lambat. Sehingga, penggunaan metode *neural network* dengan BERT *preprocessing* membutuhkan perangkat keras yang lebih baik.

Dalam penelitian lanjutan, BERT dapat digunakan untuk mengolah dokumen dan *query* secara terpisah. Secara spesifik, dokumen diolah terlebih dahulu dan disimpan dalam *cache*. Kemudian, perbandingan antara kedua metode dapat menggunakan *dataset* yang lebih besar dan memiliki dokumen yang lebih *general*. Selain itu, model BERT yang digunakan dapat dilatih untuk *fine-tuning* atau menggunakan model BERT yang lebih besar.

REFERENCES

[1] Dai, Zhuyun, and Jamie Callan. "Deeper text understanding for IR with contextual neural language modeling." *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval.* 2019.

[2] Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems 30* (2017).

[3] Robertson, Stephen, and Hugo Zaragoza. The probabilistic relevance framework: BM25 and beyond. Now Publishers Inc, 2009.

[4] Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." *arXiv preprint arXiv:1810.04805* (2018).

[5] Ghasemi, Negin, and Djoerd Hiemstra. "BERT meets Cranfield: Uncovering the Properties of Full Ranking on Fully Labeled Data." *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*. 2021.

[6] Han, Shuguang, et al. "Learning-to-rank with bert in tf-ranking." *arXiv preprint arXiv:2004.08476* (2020).

[7] Nogueira, Rodrigo, and Kyunghyun Cho. *"Passage Re-ranking with BERT." arXiv preprint arXiv:1901.04085* (2019).

[8] Schütze, Hinrich, Christopher D. Manning, and Prabhakar Raghavan. Introduction to information retrieval. Vol. 39. Cambridge: Cambridge University Press, 2008.

[9] Hapke, Hannes, Cole Howard, and Hobson Lane. Natural Language Processing in Action: Understanding, analyzing, and generating text with Python. Simon and Schuster, 2019.

[10] Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. Deep learning. MIT press, 2016.