

# Sistem Rekomendasi pada Forum Kesehatan Dengan Pemeringkatan Pertanyaan Serupa Menggunakan Pendekatan Deep Learning

<sup>1</sup>Bryan Khufa Rahmada Aula, <sup>2</sup>Chastine Faticah, dan <sup>3</sup>Diana Purwitasari  
Departemen Teknik Informatika, Fakultas Teknologi Elektro dan Informatika Cerdas,  
Institut Teknologi Sepuluh Nopember, Surabaya, Jawa Timur, Indonesia  
e-mail: <sup>1</sup>bryankhufar@gmail.com, <sup>2</sup>chastine@if.its.ac.id, <sup>3</sup>diana@if.its.ac.id

**Abstrak** – Perkembangan teknologi menyebabkan persebaran informasi yang sangat luas. Untuk membantu pencarian informasi, banyak dikembangkan sistem *information retrieval*. Aspek penting dari *information retrieval system* umumnya terdiri dari pembuatan representasi *query* dan dokumen, metode *retrieval*, dan pemeringkatan ulang dokumen. *Information retrieval system* sudah umum ditemui pada kebanyakan aplikasi, namun masih jarang yang dibangun pada domain kesehatan. Pertanyaan kesehatan sangat mementingkan konteks dan makna semantik kalimatnya. Pada eksperimen ini akan dilakukan pencarian pertanyaan serupa dengan kemiripan semantik kalimat, dan diperingkatan ulang dengan *Cross-encoder*. Awalnya dibuatkan model untuk representasi *embedding* kalimat. *Embedding* kalimat nantinya digunakan dalam proses *retrieval* yang digunakan yaitu *Semantic search*. *Semantic search* bekerja dengan cara menghitung vektor kalimat *query* yang berdekatan pada korpus dokumen menggunakan algoritma perhitungan jarak. Selanjutnya dokumen kandidat yang didapat dari proses sebelumnya diperingkatan ulang dengan *Cross-encoder* yang menerima pasangan kalimat *query* dan dokumen kandidat, kemudian dikeluarkan nilai relevansinya.

**Kata Kunci** – *Cross-Encoder, Deep Learning, Information Retrieval, Semantic search, Sentence embedding, Siamese LSTM*.

## I. PENDAHULUAN

Perkembangan teknologi membuat persebaran informasi menjadi sangat krusial dalam kehidupan. Karena semakin banyaknya informasi yang tersebar, proses penemuan informasi menjadi sesuatu yang tidak mudah jika dilakukan secara manual. Maka dari itu marak dikembangkan sistem yang dapat menemukan sebuah informasi spesifik yang dibutuhkan atau biasa dikenal dengan *information retrieval system*.

Mesin pencarian seperti Google, Duckduckgo, dan Bing merupakan *information retrieval system* yang sudah ramai digunakan masyarakat untuk mencari informasi yang bersifat umum. Untuk menemukan informasi yang jarang ditemukan atau informasi yang sangat spesifik, banyak orang bertanya langsung pada forum-forum yang ada pada internet. Termasuk dalam hal medis, masyarakat biasa menemukan informasi melalui mengajukan pertanyaan pada forum tanya-jawab kesehatan. Forum kesehatan memungkinkan masyarakat bertanya dan berdiskusi mengenai kesehatan secara umum, maupun gejala penyakit yang sedang diderita dan dapat dijawab oleh seorang tenaga medis yang kompeten. Beberapa contoh platform yang menyediakan forum kesehatan contohnya adalah Sehatq, Dokter.id, Alodokter, dan Klikdokter.

Permasalahan yang banyak ditemukan dari forum tanya jawab kesehatan adalah sulitnya masyarakat dalam

menemukan keluhan yang mirip dengan apa yang ingin ditanyakan [1]. Selain itu, seiring dengan bertambah banyaknya pertanyaan dan jawaban yang masuk, semakin banyak pula ditemukan adanya duplikasi pertanyaan [2]. Dampaknya adalah menambah parah permasalahan karena pengguna semakin kesulitan dalam mencari pertanyaan yang mirip dan berakhir pada menanyakan kembali pertanyaan yang sama. Maka dari itu perlu adanya sebuah sistem yang dapat memberi rekomendasi pertanyaan serupa yang sudah pernah terjawab sebelumnya.

Dalam kasus ini, *information retrieval system* dapat dijadikan sebagai sistem rekomendasi pertanyaan yang serupa. Tahapannya sendiri terdiri dari tiga proses utama, yakni pembuatan representasi dokumen agar dapat dilakukan pencarian, proses *retrieval* informasi dari korpus dokumen, dan pemeringkatan ulang hasilnya berdasarkan relevansi terhadap *query*. Apabila ada satu tahapan yang tidak berjalan dengan baik, maka hasil akhirnya tidak akan maksimal.

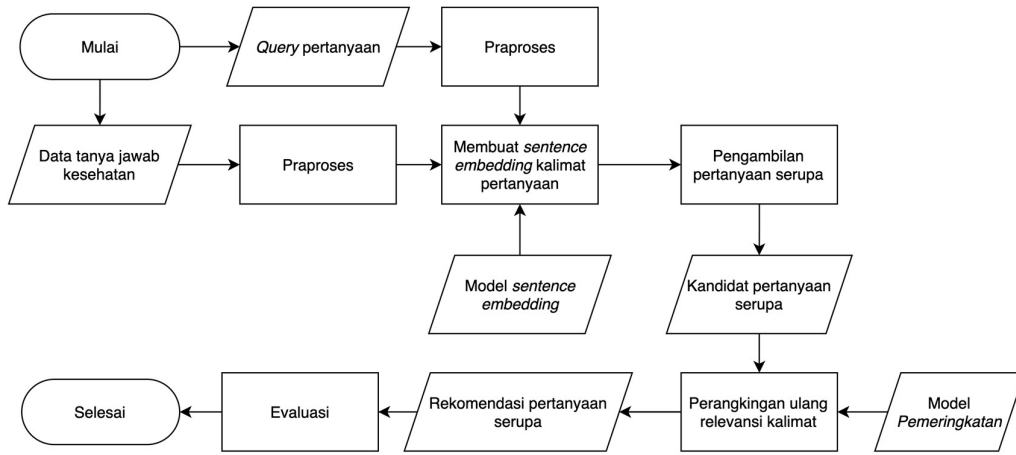
Beberapa penelitian telah dilakukan terhadap sistem *information retrieval* baik pada forum tanya-jawab, hingga pada basis data informasi. Beragam metode telah diuji cobakan, dimulai dari *Hierarchical Compare Aggregate* [3] yang berhasil mendapatkan nilai MAP 0.695. Disusul dengan pemanfaatan *Sentence embedding* dan *Semantic search* untuk melakukan pencarian sekaligus pemeringkatan melalui makna semantik kalimat [5] yang mendapatkan nilai P@5 sebesar 0.628. Pendekatan lainnya adalah dengan pemanfaatan BM25 dan pemeringkatan ulang secara semantik dengan *Siamese LSTM* [4] yang mengeluarkan hasil P@5 0.502 dan MAP 0.573. Hasil yang dicapai dari beberapa penelitian tersebut cukup beragam, namun waktu pencarian sistem jarang diperhatikan.

Pada eksperimen ini, dilakukan implementasi proses penemuan kembali pertanyaan serupa dengan memanfaatkan *Sentence embedding* untuk membentuk representasi kalimat dengan mempertahankan makna semantiknya. *Semantic search* selanjutnya digunakan untuk menemukan kandidat pertanyaan serupa. Kalimat-kalimat pertanyaan kandidat tersebut kemudian diperingkatan ulang menggunakan jaringan *transformer Cross-encoder*.

## II. METODOLOGI

### A. Deskripsi Data

Data merupakan kumpulan kalimat pertanyaan yang diajukan oleh pengguna. Data yang digunakan dalam pembuatan sistem untuk menjadi korpus informasinya adalah



Gambar 1 Diagram Alir Sistem

kalimat-kalimat pertanyaan. *Dataset* diambil dari tiga forum kesehatan, forum-forum tersebut diantaranya adalah Alodokter ([www.alodokter.com](http://www.alodokter.com)), Dokterid ([www.dokter.id](http://www.dokter.id)), dan Klikdokter ([www.klikdokter.com](http://www.klikdokter.com)). Total data yang terkumpul adalah sebanyak 358,331 kalimat tanya jawab. Data merupakan hasil diskusi forum yang dibuat dalam rentang waktu 23 September 2017 pukul 18:50 WIB hingga 9 September 2020 pukul 23:59 WIB. Data diambil dari seluruh kategori penyakit yang ada pada setiap forum kesehatan tersebut. Pertanyaan yang ada pada *dataset* adalah yang pertanyaan sudah berhasil dijawab oleh seorang dokter atau tenaga ahli yang kompeten.

#### B. Desain Umum Sistem

Desain sistem yang akan dibangun meliputi perancangan data dan perancangan proses. Perancangan data berisi penjelasan mengenai deskripsi data yang digunakan dalam pembuatan sistem. Pada perancangan proses dilakukan praproses pada data hingga menjadi representasi *embedding* kalimatnya, proses temu kembali informasi menggunakan *semantic search* dan pembuatan model perangkingan ulang dengan *cross-encoder*. Gambaran umum sistem secara keseluruhan dapat dilihat pada Gambar 1.

#### C. Praproses Data

Tahapan ini umumnya berperan dalam menormalkan data menjadi bentuk yang lebih seragam sebelum diproses lebih lanjut. Beberapa tahapan praproses yang dilakukan pertama-tama adalah penghapusan tag html karena data bersumber dari *crawling* website forum kesehatan. Selanjutnya dilakukan *case folding* untuk mengubah semua huruf kapital menjadi huruf kecil. Lalu simbol-simbol dan tanda baca yang terdapat pada teks dihapus untuk membersihkan dokumen. Kemudian *whitespace* yang berlebih dihapus baik yang ditulis oleh pengguna maupun sisa dari operasi sebelumnya. Pada akhirnya dihasilkan sebuah korpus yang berisi kalimat-kalimat hasil praproses.

#### D. Pembuatan Embedding Kalimat

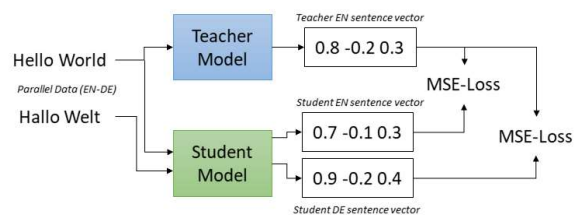
*Sentence embedding* atau *embedding* kalimat dapat digambarkan sebagai metode pemetaan kalimat kedalam bentuk vektor sebagai sarana untuk merepresentasikan teks dengan bilangan real agar dapat diproses untuk *machine learning*.

Metode *embedding* kalimat yang digunakan dalam sistem dibangun dengan kerangka kerja SBERT menggunakan model *pre-trained* yang sudah dilatih dengan *dataset* berbahasa Inggris yang lebih besar. SBERT melakukan modifikasi pada *network* BERT menggunakan arsitektur *Siamese* dan *Triplet Network* yang dapat memperoleh *embedding* kalimat yang bermakna secara semantik [6]. Kalimat yang secara semantis mirip berada pada *vector space* yang berdekatan.

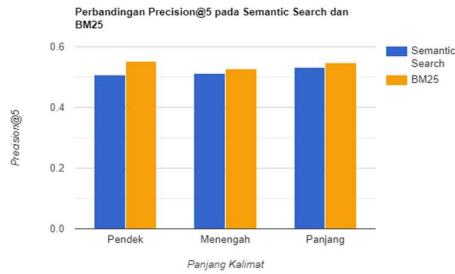
Model *pre-trained* tersebut nantinya akan dilakukan *knowledge distillation* agar dapat memahami *embedding* kalimat berbahasa Indonesia. Proses *knowledge distillation* dijalankan dengan memanfaatkan dua buah model. Model pertama akan menjadi *teacher model* yang berperan sebagai acuan representasi vektor dalam sebuah bahasa. Model kedua adalah *student model* yang nantinya akan dilatih untuk membuat vektor yang menyerupai *teacher model*.

Sepasang kalimat bahasa Inggris dan Indonesia dijadikan sebagai masukan. *Teacher model* akan menghasilkan sebuah representasi vektor berbahasa Inggris. Kemudian *student model* akan menghasilkan dua buah vektor yang merepresentasikan kalimat masukan berbahasa Inggris dan vektor kalimat berbahasa Indonesia. *Loss function* yang digunakan dalam proses *training* adalah Mean Square Error. Arsitektur prosesnya dapat dilihat pada Gambar 2.

*Dataset* yang digunakan sebagai masukan dalam proses *knowledge distillation* adalah *dataset* pasangan kalimat terjemahan TED2020 berbahasa Inggris-Indonesia. Pasangan kalimat berbahasa Inggris-Indonesia total keseluruhannya ada sebanyak 163,319 kalimat [7].



Gambar 2 Arsitektur knowledge distillation Sentence embedding [6]



Gambar 3 Perbandingan  $Precision@5$  antara *Semantic search* dan BM25

Model *pre-train* yang digunakan dalam tahap ini adalah *quora-distilbert-base* yang menjadi *teacher model*. Sedangkan yang menjadi *student model* adalah *quora-distilbert-multilingual*. Proses *knowledge distillation* dilakukan dengan menggunakan *mean aggregation* sebagai *pooling layer*. Setelah vektor kalimat sudah didapatkan dari kedua model, dievaluasi *Mean Squared Error* dengan menghitung nilai *euclidean distance* dari keduanya.

#### E. Semantic Search

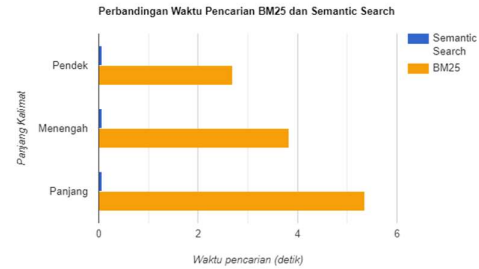
Metode penemuan kembali informasi atau *retrieval* menggunakan *Semantic search*, yang berfokus pada kemiripan makna dari kalimat dibandingkan cara konvensional yang menggunakan pencocokan kata kunci. *Semantic search* juga memahami sinonim dari kata-kata dan kalimat [6].

Tahapannya diawali dengan membuat representasi vektor seluruh data kalimat pertanyaan kesehatan yang telah di praproses. Proses yang sama juga dilakukan untuk kalimat *query* yang baru dimasukkan. Pembuatan vektor dilakukan dengan menjalankan *inference* model *sentence embedding* yang telah dibuat pada tahapan sebelumnya. Korpus kalimat pertanyaan yang sudah dilakukan praproses pada tahapan pertama dimasukkan kedalam *inference* pada model. Setelah mendapatkan bentuk *embedding* dari semua korpus kalimat pertanyaan dan *query*, korpus dibagi menjadi beberapa *chunk* untuk mempercepat proses pencarian. Kemudian semua vektor kalimat pada setiap *chunk* dihitung nilai similaritasnya dengan vektor *query* dengan *cosine similarity*. Hasilnya kemudian diurutkan berdasarkan nilai similaritas yang paling besar dan diambil 10 kalimat teratas.

#### F. Pemeringkatan Ulang

Pada tahap ini, kalimat pertanyaan kandidat serupa yang didapatkan dari proses sebelumnya diurutkan kembali untuk menentukan kalimat mana yang paling relevan dengan *query*. Proses pemeringkatan ulang menggunakan arsitektur *Cross-encoder*. *Cross-encoder* adalah *single transformer network* yang menerapkan mekanisme *full-attention* [8]. Dua buah kalimat dan kandidat objektif masukan di-*encode* bersamaan pada sebuah *transformer* untuk digabungkan menjadi sebuah representasi vektor. Karena kandidat dapat bergabung pada konteks *input* di layer *transformer*, *Cross-encoder* mampu membuat representasi *input* yang lebih sensitif terhadap objektif akhir [9].

Nantinya akan terdapat dua kalimat yang dimasukkan kedalam *Transformer network*. Nantinya diakhir akan



Gambar 4 Perbandingan waktu pencarian antara *Semantic search* dan BM25

dihasilkan sebuah nilai diantara 0 dan 1 yang mengindikasikan kemiripan kedua kalimat tersebut.

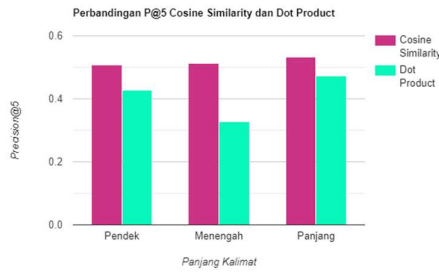
Proses pembuatan model dengan cara *fine-tuning pre-train model cross-encoder quora-roberta-large*. Model tersebut dipilih karena sebelumnya sudah dilatih dengan sebuah *dataset* kalimat duplikat Quora dan menggunakan arsitektur RoBERTa yang merupakan hasil pelatihan ulang BERT dengan 160GB data teks dan menggunakan tambahan metode *dynamic masking* sehingga token yang telah diberi *mask* berubah secara dinamis pada setiap *epoch* pelatihan [10].

*Dataset* yang digunakan untuk melakukan proses ini adalah data terjemahan kalimat parafrase *Paraphrase Adversarial from Word Scrambling* (PAWS). PAWS adalah *dataset* pasangan kalimat paraphrase yang diambil dari Wikipedia. Kalimat dibuat dalam bentuk pasangan parafrase melalui penggantian sebuah kata, dan bentuk *back translation* dari kalimat tersebut. *Dataset* terdiri dari 49,401 kalimat untuk *training*, dan 8,000 data untuk proses *development* dan *testing* [11]. Terdapat label yang menandakan apakah pasangan kalimat tersebut merupakan bentuk parafrasenya atau bukan. Label 0 berarti bukanlah parafrase, dan 1 berarti parafrase.

*Dataset* tersebut diterjemahkan dahulu kedalam bahasa Indonesia sebelum dijadikan masukan dalam proses *fine-tuning* model. Evaluator yang digunakan dalam proses ini adalah *Cross Encoder Binary Classification Evaluator* yang menerima pasangan kalimat dan label biner (0 atau 1) kemudian keluarannya dihitung nilai *average precision* dan nilai skor f1 terbaik. Dalam *training*-nya 10% data *train* digunakan untuk proses *warmup*. *Optimizer* yang digunakan adalah Adam dengan *learning rate*  $2e-5$  dan dilakukan sebanyak 4 *epoch*. Model kemudian disimpan untuk nantinya dilakukan *inference* pada tahap pemeringkatan ulang di keseluruhan proses.

TABEL I HASIL METRIK UJI BM25 DAN SEMANTIC SEARCH

	Semantic Search			BM25		
	P@5	MAP	Waktu	P@5	MAP	Waktu
Kalimat pendek	0.507	0.886	0.063	0.553	0.889	2.690
Kalimat menengah	0.513	0.886	0.061	0.527	0.889	3.830
Kalimat panjang	0.533	0.851	0.066	0.547	0.887	5.350
Rata-rata	0.518	0.875	0.063	0.542	0.888	3.956



Gambar 6 Perbandingan *Precision@5* pencarian antara *Cosine similarity* dan *Dot product*

### III. UJI COBA DAN ANALISIS

Pengujian dilakukan untuk dapat menganalisa dan evaluasi performa dari sistem yang dibuat. Uji coba dilakukan pada metode *retrieval* untuk membandingkan *Semantic search* dengan BM25, metode *distance measure* untuk membandingkan *Cosine similarity* dengan *Dot product*, dan pada model pemeringkatan ulang *Cross-encoder* dengan *Siamese LSTM*.

Data yang diuji adalah 90 kalimat yang diambil dari 10 kategori pertanyaan yang paling sering ditanyakan. Tiap kategori diambil 3 kalimat pendek, 3 kalimat menengah, dan 3 kalimat panjang. Untuk menentukan panjang pendeknya kalimat digunakan nilai kuartil pertama, kedua, dan ketiga pada setiap kategori.

Proses anotasi dilakukan oleh 8 orang pengguna yang merupakan masyarakat awam berumur rata-rata 22 tahun yang tidak memiliki latar belakang tertentu dalam bidang medis. Prosesnya dilakukan dengan memanfaatkan Label Studio sebagai antarmuka anotasinya. Data keluaran sistem dijadikan masukan pada Label Studio, dan data hasil anotasi diekspor lalu dihitung nilai evaluasinya.

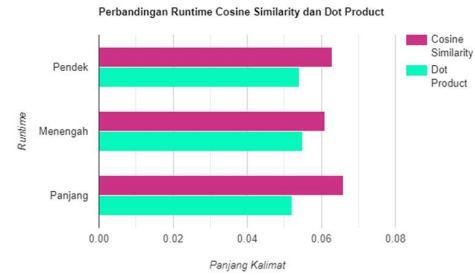
Metrik evaluasi yang digunakan adalah *Precision@5*, *Average Precision*, *Mean Average Precision*, dan waktu eksekusi. *P@5* mampu menilai kemampuan sistem dalam menemukan kalimat-kalimat yang relevan dari 5 kalimat yang ditampilkan. MAP menilai kemampuan sistem dalam memberi peringkat atau urutan kalimat keluaran, karena menghitung nilai presisi dengan bobot pada setiap urutannya.

#### A. Uji Coba metode Retrieval dengan BM25 dan Semantic Search

Uji coba dilakukan pada metode BM25 dan Semantic Search untuk membandingkan performa hasil pencarian berdasarkan semantis kalimat dengan pencarian berdasarkan leksikal kalimatnya. Hasil uji coba dapat dilihat pada TABEL I. Visualisasi perbandingan nilai *Precision@5* dapat dilihat pada Gambar 3. Perbandingan waktu eksekusi antar keduanya dapat dilihat pada Gambar 4.

#### B. Uji Coba metode distance measure dengan Cosine similarity & Dot product

Uji coba pada metode *distance measure* *Cosine similarity* dan *Dot product* dilakukan untuk mengetahui algoritma mana yang dapat memberikan hasil perhitungan vektor *embedding* terbaik yang nantinya memberi hasil pencarian yang relevan dalam proses *Semantic search*. Hasil uji coba dapat dilihat pada TABEL III. Visualisasi Perbandingan nilai *P@k* dapat dilihat pada Gambar 5. Sedangkan perbandingan waktu eksekusi keduanya dapat dilihat pada Gambar 6.



Gambar 5 Perbandingan waktu pencarian antara *Cosine similarity* dan *Dot product*

#### C. Uji Coba pada metode model re-ranking Cross-Encoder & Siamese LSTM

Uji coba juga dilakukan pada model pemeringkatan yang berada di akhir proses. Tujuannya adalah untuk membandingkan hasil akhir keluaran sistem dengan metode lain yang sudah ada. Yang menjadi masukan dalam proses ini adalah pasangan kalimat *query* dengan 10 kalimat hasil *retrieval* menggunakan *Semantic search*. Keluarannya adalah 5 kalimat yang sudah diurutkan berdasarkan relevansinya. Yang akan diuji pada tahap ini adalah pemeringkatan ulang menggunakan model *Cross-encoder* dan *Siamese LSTM*. Model *Siamese LSTM* dibuat dengan mengimplementasikan referensi [12]. Hasil uji coba dapat dilihat pada TABEL II. Perbandingan nilai MAP dapat dilihat pada Gambar 7. Sedangkan perbandingan waktu eksekusi keduanya dapat dilihat pada Gambar 8.

#### D. Analisa Hasil

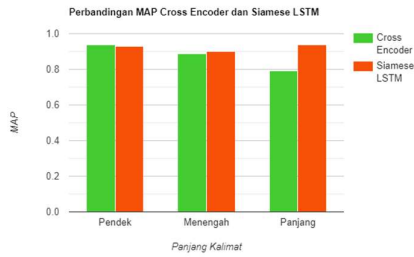
Berdasarkan hasil dari skenario uji coba yang telah dilakukan pada metode *retrieval*, didapatkan hasil sebagai berikut. Pada uji coba algoritma *Semantic search* untuk dataset kalimat pertanyaan kesehatan, diperoleh rata-*Precision@5* sebesar 0.518, dan *Mean Average Precision* sebesar 0.875. Sedangkan hasil dari algoritma BM25 memiliki rata-rata nilai *Precision@5* sebesar 0.542, dan *Mean Average Precision* sebesar 0.888.

Nilai presisi menjadi metrik yang penting untuk menentukan baik buruknya suatu algoritma IR. Dalam proses ini, diperlukan algoritma yang mampu mengeluarkan dokumen relevan terbanyak dan belum terlalu mementingkan urutan pemeringkatannya sehingga nilai MAP tidak terlalu menjadi pertimbangan. Kedua algoritma dinilai memiliki kemampuan yang relatif setara untuk menemukan kalimat yang relevan berdasarkan *query*.

TABEL II HASIL METRIK UJI COBA CROSS-ENCODER DAN SIAMESE LSTM

	Cross-Encoder			Siamese LSTM		
	P@5	MAP	Waktu	P@5	MAP	Waktu
Kalimat pendek	0.500	0.939	0.321	0.493	0.929	16.300
Kalimat menengah	0.507	0.888	0.431	0.413	0.898	16.800
Kalimat panjang	0.520	0.793	0.565	0.526	0.936	17.800
Rata-rata	0.509	0.873	0.439	0.478	0.921	16.967





Gambar 7 Perbandingan MAP pencarian antara *Cross-encoder* dan *Siamese LSTM*

TABEL III HASIL METRIK UJI COBA *COSINE SIMILARITY* DAN *DOT PRODUCT*

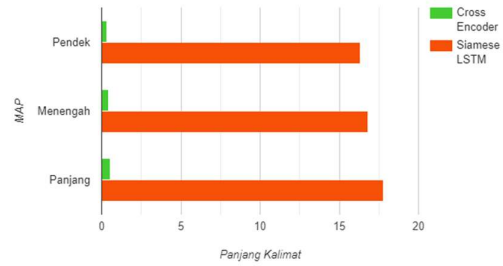
	Cosine Similarity			Dot Product		
	P@5	MAP	Waktu	P@5	MAP	Waktu
Kalimat pendek	0.507	0.886	0.063	0.427	0.889	0.054
Kalimat menengah	0.513	0.886	0.061	0.327	0.889	0.055
Kalimat panjang	0.533	0.851	0.066	0.473	0.887	0.052
Rata-rata	0.518	0.875	0.063	0.409	0.888	0.054

Algoritma *Semantic search* menemukan informasi dengan cara mencari kedekatan vektor *embedding* antara kalimat *query* dan kalimat pada korpus. Kemiripannya dihitung menggunakan fungsi *Cosine similarity*. Vektor *embedding* kalimat-kalimat korpus sudah dibuat sebelum proses pencarian. Pada saat dimasukkan *query* baru yang akan dicari, *query* tersebut hanya perlu dibuatkan vektor *embedding* kalimatnya, kemudian dihitung pada semua vektor kalimat dan diambil yang memiliki jarak terdekat. Sedangkan BM25 mencari berdasarkan istilah-istilah yang muncul pada kalimat *query* yang ada di korpus terlepas dari keterkaitan antar istilah *query* dalam kalimat. Prosesnya diawali dengan proses tokenisasi kalimat dalam korpus yang sudah dipersiapkan sebelumnya. Saat kalimat *query* dimasukkan, dilakukan tokenisasi yang sama, kemudian dicocokkan dengan token-token yang ada pada korpus berdasarkan kemiripan leksikalnya. Proses yang memakan waktu cukup lama adalah pada saat tokenisasinya, karena butuh dilakukan penghapusan

TABEL IV CONTOH HASIL KELUARAN *SEMANTIC SEARCH*

Isi	Kalimat Keluaran
<i>Query</i>	selamat sore dokter saya ingin bertanya hari ini saya dapat haid sedangkan besok jadwal suntik ulang kb suntik 3 bulan saya saya harus suntik kb saat haid atau menunda haid selesai dulu terimakasih
Kalimat 1	selamat sore dokter saya ingin bertanya hari ini saya dapat haid sedangkan besok jadwal suntik ulang kb suntik 3 bulan saya saya harus suntik kb saat haid atau menunda haid selesai dulu terimakasih
Kalimat 2	selamat sore dokter saya mau tanya kalau pasca operasi usus buntu dan tidak kentut selama 3 hari itu bahaya tidak dok tolong di balas ya dok penting terima kasih
Kalimat 3	selamat siang dok saya mau tanya obat agar lekas haid setelah lepas kb suntik 3 bulan apa ya dok terimakasihlgh sbmlnya
Kalimat 4	siang dokter mau tanya kalau obat untuk gatal pada selakangan apa dan obat keputihan selama kurg lbh 1 mnngu blum berhenti apa mohon jawabannya terima kasih dokter
Kalimat 5	hy dok saya ke dokter sya minta suntik kb 1 bulan stlh sya mau suntik lagi saya lihat di kertas ny jangka waktu sampe 3 bulan kemungkinan dokter ny kasih suntikan 3 bulan tapi saya tkut dia salah tulis soalnya saya minta yg 1 bulan klo saya suntik lagi apa kah ngga apa2 dok tlong di jawab y dok trimakasih

In



Gambar 8 Perbandingan waktu pencarian antara *Cross-encoder* dan *Siamese LSTM*

*stop word*. Maka dari itu semakin panjang kalimat yang perlu ditokenisasi semakin banyak pula kata-kata yang harus dihapus *stop word*-nya.

Pada hasil dari skenario uji coba yang telah dilakukan pada algoritma perhitungan *similarity*, didapatkan hasil sebagai berikut. Pada uji coba algoritma *Cosine similarity* untuk *Semantic search*, diperoleh rata-rata nilai *Precision@5* sebesar 0.518, dan *Mean Average Precision* sebesar 0.875. Sedangkan hasil dari algoritma *Dot product* yang memiliki rata-rata nilai *Precision@5* sebesar 0.409, dan *Mean Average Precision* sebesar 0.792.

Pada dasarnya *Cosine similarity* hanya memperhitungkan perbedaan sudut antar dua buah vektor. Sedangkan *Dot product* mempertimbangkan sudut dan magnitudo antar vektornya. Karena panjang kalimat pertanyaan bervariasi, magnitudo vektor kalimat juga bervariasi. Dalam kasus ini magnitudo vektor lebih baik tidak dimasukkan kedalam perhitungan karena malah memberi hasil yang lebih buruk.

Berdasarkan hasil dari skenario uji coba yang telah dilakukan pada model pemeringkatan ulang, didapatkan hasil sebagai berikut. Pada uji coba model *Cross-encoder* untuk pasangan kalimat hasil keluaran *Semantic search*, diperoleh rata-rata nilai *Precision@5* sebesar 0.509, dan *Mean Average Precision* sebesar 0.873. Sedangkan hasil dari model *Siamese LSTM* menghasilkan rata-rata nilai *Precision@5* sebesar 0.478, dan *Mean Average Precision* sebesar 0.921.

Ditemukan bahwa model *Siamese LSTM* memiliki hasil yang baik dalam hal menentukan peringkat kalimat berdasarkan relevansinya, ditunjukkan oleh nilai MAP-nya. Model tersebut juga tampak tidak terlalu terpengaruh oleh panjang kalimat. Karena pada dasarnya model *Siamese LSTM* mempertimbangkan kemunculan leksikal dan juga hubungan semantisnya. Akan tetapi nilai presisi yang dihasilkan yang hanya sebesar 0.478. Hasilnya menunjukkan dokumen yang diambil hanya sedikit yang sesuai namun urutannya sudah benar. Hal ini menunjukkan bahwa model baik dalam kasus pencarian dokumen dengan kandidat jawaban yang sedikit.

Sedangkan pada model *Cross-encoder* nampak performanya sangat baik pada kalimat pendek namun menurun seiring dengan bertambahnya panjang kalimat. Hal ini dikarenakan model *Cross-encoder* di-training dengan data kalimat pendek, dari sini dapat diketahui bahwa model *Cross-encoder* tidak memiliki toleransi yang baik pada perbedaan panjang data saat *inference* dengan data pada saat *training*.

Terdapat perbedaan yang sangat signifikan dari waktu *runtime* yang diperlukan untuk melakukan *inference* setiap model, selisihnya dapat mencapai 17 detik. Model *Cross-encoder* jauh lebih cepat dalam eksekusi *inference*-nya.

Proses yang memakan waktu cukup lama pada model *Siamese LSTM* adalah pada saat membuat *Word2vec embedding* kalimatnya, maka dari itu kalimat yang lebih panjang akan memakan waktu lebih lama.

Hasil yang dimunculkan oleh *Semantic search* kurang terlalu memuaskan karena pencarian berdasarkan *embedding* kalimat yang berdekatan ditentukan oleh makna semantik yang mirip. Dari contoh pada TABEL IV, dapat dilihat bahwa kalimat yang dikeluarkan secara umum berisi tentang keluhan penyakit dan rentang waktunya. Secara semantik hasilnya mirip, akan tetapi penyakitnya berbeda. Model *embedding* kalimat juga belum terlalu pandai dalam memahami domain pertanyaan kesehatan karena keterbatasan *dataset* kalimat serupa pada pertanyaan medis yang sudah dilabeli. Penyebab lainnya adalah karena pertanyaan medis yang sangat spesifik. Semakin spesifik pertanyaan, maka akan semakin sedikit pula pertanyaan yang serupa.

#### IV. KESIMPULAN DAN SARAN

*Sentence Embedding* berhasil membuat representasi kalimat yang dapat mempertahankan makna semantiknya, sehingga dapat ditentukan kalimat yang serupa dari *vector space*-nya. Akan tetapi untuk kalimat-kalimat pada domain medis nampak pencarian berdasarkan semantik saja mengeluarkan hasil yang kurang memuaskan. Hal ini dikarenakan meskipun secara semantik kalimat yang dihasilkan mirip, penyakit yang ditanyakan sedikit berbeda. Permasalahan tersebut dapat diatasi dengan menambahkan pencarian dengan menggunakan kata kunci seperti BM25.

Untuk menemukan kandidat dokumen yang relevan dapat menggunakan algoritma *Semantic Search* yang mampu menghasilkan nilai rata-rata P@5 0.518 dan MAP 0.875, selama 0.064 detik. Hasilnya dinilai konsisten karena tidak terpengaruh oleh panjang kalimat. Pencarian dengan kata kunci dinilai dapat meningkatkan nilai P@5 hingga 0.03. Fungsi *similarity* yang lebih baik adalah *Cosine similarity* jika dibandingkan dengan *Dot Product* yang lebih cepat 0.01 detik namun menghasilkan nilai P@5 yang lebih kecil 0.108.

Dalam pemeringkatan ulang, model *Cross-encoder* berhasil menemukan kalimat relevan lebih banyak dan lebih cepat. Sedangkan model *Siamese LSTM* menghasilkan kalimat relevan lebih sedikit namun hasil peringkatnya lebih baik. Hal ini menunjukkan *Siamese LSTM* adalah model yang baik untuk sistem dengan keluaran jawaban yang lebih sedikit, namun kurang baik untuk sistem dengan keluaran jawaban yang lebih banyak.

Untuk pengembangan penelitian lebih lanjut, sebaiknya dilakukan eksplorasi lebih dalam pada tahap pra-proses teks sebelum dilakukan pembuatan vektor *embedding*, terutama pada pengaruh *stop word*, *stemming*, dan *lemmatization* terhadap makna semantik dari kalimat. Menyesuaikan model *sentence embedding* menggunakan *dataset* yang sesuai dengan domain lebih jauh lagi, agar dapat membuat bentuk *embedding* yang lebih representatif. Menambah variasi data dengan panjang kalimat yang berbeda-beda untuk

meningkatkan toleransi model pemeringkatan terhadap panjang kalimat yang tidak menentu.

#### REFERENSI

- [1] A. Ben Abacha and P. Zweigenbaum, "MEANS: A medical question-answering system combining NLP techniques and semantic Web technologies," *Inf. Process. Manag.*, vol. 51, no. 5, pp. 570–594, 2015, doi: 10.1016/j.ipm.2015.04.006.
- [2] N. Othman, R. Faiz, and K. Smaïli, "Enhancing question retrieval in community question answering using word embeddings," *Procedia Comput. Sci.*, vol. 159, pp. 485–494, 2019, doi: 10.1016/j.procs.2019.09.203.
- [3] M. S. Zahedi, M. Rahgozar, and R. A. Zoroofi, "HCA: Hierarchical Compare Aggregate model for question retrieval in community question answering," *Inf. Process. Manag.*, vol. 57, no. 6, p. 102318, 2020, doi: 10.1016/j.ipm.2020.102318.
- [4] N. Othman, R. Faiz, and K. Smaïli, "Manhattan siamese LSTM for question retrieval in community question answering," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11877 LNCS, pp. 661–677, 2019, doi: 10.1007/978-3-030-33246-4\_41.
- [5] A. A. Deshmukh and U. Sethi, "IR-BERT: Leveraging BERT for Semantic Search in Background Linking for News Articles," 2020, [Online]. Available: <http://arxiv.org/abs/2007.12603>.
- [6] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using siamese BERT-networks," *EMNLP-IJCNLP 2019 - 2019 Conf. Empir. Methods Nat. Lang. Process. 9th Int. Jt. Conf. Nat. Lang. Process. Proc. Conf.*, pp. 3982–3992, 2020, doi: 10.18653/v1/d19-1410.
- [7] N. Reimers and I. Gurevych, "Making Monolingual Sentence Embeddings Multilingual using Knowledge Distillation," pp. 4512–4525, 2020, doi: 10.18653/v1/2020.emnlp-main.365.
- [8] N. Thakur, N. Reimers, J. Daxenberger, and I. Gurevych, "Augmented SBERT: Data Augmentation Method for Improving Bi-Encoders for Pairwise Sentence Scoring Tasks," vol. 2, pp. 296–310, 2021, doi: 10.18653/v1/2021.naacl-main.28.
- [9] Anonymous, "Poly-encoders : Architectures and Pre training Strategies for Fast and Accurate Multi sentence Scoring," *ICLR 2020 Conf.*, pp. 1–13, 2019.
- [10] Y. Liu *et al.*, "RoBERTa: A Robustly Optimized BERT Pretraining Approach," no. 1, 2019, [Online]. Available: <http://arxiv.org/abs/1907.11692>.
- [11] Y. Zhang, J. Baldridge, and L. He, "PAWS: Paraphrase adversaries from word scrambling," *NAACL HLT 2019 - 2019 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. - Proc. Conf.*, vol. 1, no. 2, pp. 1298–1308, 2019.
- [12] J. Mueller and A. Thyagarajan, "Siamese recurrent architectures for learning sentence similarity," *30th AAAI Conf. Artif. Intell. AAAI 2016*, no. November 2015, pp. 2786–2792, 2016.