



Tersedia online di www.journal.unipdu.ac.id
Unipdu

Terakreditasi S2 – SK No. 34/E/KPT/2018

Halaman jurnal di www.journal.unipdu.ac.id/index.php/register



Query Expansion menggunakan Word Embedding dan Pseudo Relevance Feedback

Evan Tanuwijaya ^a, Safri Adam ^b, Mohammad Fatoni Anggris ^c, Agus Zainal Arifin ^d

^{a,b,c,d} Teknik Informatika, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia

email: ^aevantanuwijaya.18051@mhs.its.ac.id, ^bsafriadam.18051@mhs.its.ac.id, ^cmohammadanggris.18051@mhs.its.ac.id,
^dagusza@cs.its.ac.id

INFO ARTIKEL

Sejarah artikel:

Menerima 19 Desember 2018

Revisi

17 April 2019

Diterima 11 Mei 2019

Online 29 Mei 2018

Kata kunci:

Pseudo Relevant Feedback

Query Expansion

Word Embedding

Keywords:

Pseudo Relevant Feedback

Query Expansion

Word Embedding

Style APA dalam menyitasi artikel ini:

Tanuwijaya, E., Adam, S., Anggris, M. F., & Arifin, A. Z. (2019). QE menggunakan Word Embedding dan Pseudo Relevance Feedback. *Register: Jurnal Ilmiah Teknologi Sistem Informasi*, 5(1), 47-54.

ABSTRAK

Kata kunci merupakan hal terpenting dalam mencari sebuah informasi. Penggunaan kata kunci yang tepat menghasilkan informasi yang relevan. Saat penggunaannya sebagai *query*, pengguna menggunakan bahasa yang alami, sehingga terdapat kata di luar dokumen jawaban yang telah disiapkan oleh sistem. Sistem tidak dapat memproses bahasa alami secara langsung yang dimasukkan oleh pengguna, sehingga diperlukan proses untuk mengolah kata-kata tersebut dengan mengekspansi setiap kata yang dimasukkan pengguna yang dikenal dengan *Query Expansion* (QE). Metode QE pada penelitian ini menggunakan *Word Embedding* karena hasil dari *Word Embedding* dapat memberikan kata-kata yang sering muncul bersama dengan kata-kata dalam *query*. Hasil dari *word embedding* dipakai sebagai masukan pada *pseudo relevance feedback* untuk diperkaya berdasarkan dokumen jawaban yang telah ada. Metode QE diterapkan dan diuji coba pada aplikasi *chatbot*. Hasil dari uji coba metode QE yang diterapkan pada *chatbot* didapatkan nilai *recall*, *precision*, dan *F-measure* masing-masing 100%; 70% dan 82,35 %. Hasil tersebut meningkat 1,49% daripada *chatbot* tanpa menggunakan QE yang pernah dilakukan sebelumnya yang hanya meraih akurasi sebesar 68,51%. Berdasarkan hasil pengukuran tersebut, QE menggunakan *word embedding* dan *pseudo relevance feedback* pada *chatbot* dapat mengatasi *query* masukan dari pengguna yang ambigu dan alami, sehingga dapat memberikan jawaban yang relevan kepada pengguna.

ABSTRACT

Keywords are the most important words and phrases used to obtain relevant information on content. Although users make use of natural languages, keywords are processed as queries by the system due to its inability to process. The language directly entered by the user is known as query expansion (QE). The proposed QE in this research uses word embedding owing to its ability to provide words that often appear along with those in the query. The results are used as inputs to the pseudo relevance feedback to be enriched based on the existing documents. This method is also applied to the chatbot application and precision, and F-measure values of the results obtained were 100%, 70%, 82.35% respectively. The results are 1.49% better than chatbot without using QE with 68.51% accuracy. Based on the results of these measurements, QE using word embedding and pseudo which gave relevance feedback in chatbots can resolve ambiguous and natural user's input queries thereby enabling the system retrieve relevant answers.

© 2019 Register: Jurnal Ilmiah Teknologi Sistem Informasi. Semua hak cipta dilindungi undang-undang.

1. Pendahuluan

Informasi sangat dibutuhkan oleh semua orang. Dalam mencari suatu informasi, banyak orang menggunakan bahasa yang beragam. Keragaman bahasa tersebut dapat mempengaruhi pencarian informasi seperti penggunaan, pemaknaan, bahkan gabungan kata. Pada sebuah sistem, dibutuhkan

sebuah proses agar dapat memahami keberagaman bahasa yang digunakan oleh orang-orang tersebut, karena bahasa yang digunakan untuk mencari informasi merupakan kata kunci yang penting dalam mencari sebuah informasi. Kata kunci atau *query* yang tepat akan menghasilkan informasi yang sesuai. Namun, bahasa yang digunakan untuk memasukkan *query* tersebut beragam dan bahkan ambigu, sehingga informasi yang didapatkan bisa sesuai dengan yang diinginkan dan juga tidak (Dalpiaz, Ferrari, Franch, & Palomares, 2018). Dalam hal ini, perlu adanya proses untuk mengenali *query* yang dimasukkan agar sistem dapat menarik informasi yang sesuai.

Untuk meningkatkan kinerja penarikan informasi dilakukan beberapa cara, salah satunya dengan melakukan *Query Expansion* (QE) (Nie, Jiang, Ren, Sun, & Li, 2016) yang berguna untuk memperkaya sebuah *query*. Pada QE, *query* dimasukkan oleh pengguna dan diproses oleh sistem kemudian ditambahkan dengan *term* baru yang mirip dengan *query* yang dimasukkan, sehingga menghasilkan *query* baru yang lebih sesuai (Lee & Lee, 2014). Digunakannya QE mempermudah sistem menarik informasi yang lebih sesuai dengan *query* yang dicari oleh pengguna dan diharapkan mampu menyelesaikan masalah *query* yang tidak akurat.

Metode QE yang pernah diterapkan salah satunya menggunakan *relevance feedback* (Ooi, Ma, Qin, & Liew, 2015). Namun, beberapa masalah seperti kesalahan ejaan, lintas bahasa, dan ketidakcocokan antara kosakata pengguna dengan kosakata pada dokumen tidak bisa diselesaikan hanya dengan *relevance feedback*. Oleh karena itu, diperlukan metode yang dapat mengatasi masalah tersebut dengan menggunakan *word embedding* yang dapat menangkap informasi semantik dan sintaksis kata-kata dari korpus besar yang tidak berlabel. Dengan menggunakan metode ini, sistem dapat memproses bahasa alami atau *Natural Language Processing* (NLP) (Dalpiaz, Ferrari, Franch, & Palomares, 2018) dengan mengambil informasi dari bahasa tersebut dan mengetahui hubungan makna antara suatu kata (Şenel, Utlu, Yücesoy, Koç, & Çukur, 2018). Informasi dari kata-kata tersebut direpresentasikan ke dalam masing-masing vektor. Salah satu tipe *relevance feedback* yang sering digunakan adalah *explicit feedback* yaitu memanfaatkan bukti eksplisit yang menunjukkan relevansi dari sebuah dokumen. *Pseudo relevance feedback* adalah salah satu bagian dari jenis *explicit feedback*. *Pseudo relevance feedback* merupakan metode *expansion query* yang digunakan dengan mencocokkan *term* pada *query* dengan *term* yang ada pada dokumen yang ingin ditunjukkan (Yan & Gao, 2017). Nilai dari hasil pencocokan *term* terhadap masing-masing dokumen akan dibandingkan dengan nilai dari dokumen yang lain. Kemudian, dokumen yang memiliki nilai tertinggi tersebut akan dipakai sebagai dokumen yang sesuai dengan *query* yang diinputkan oleh pengguna. Akan tetapi, *term* pada *query* yang dimasukkan oleh pengguna harus semirip mungkin dengan *term* pada dokumen agar metode ini dapat efektif (Xu, Lin, Lin, Yang, & Xu, 2018).

Word embedding sedang menunjukkan kemampuannya dalam bidang pemrosesan bahasa alami dan *Information Retrieval* (IR) (Liu, Huang, Lut, Gao, & Zhang, 2017). Oleh sebab itu, dalam penelitian ini, digunakan metode *word embedding* sebagai metode utama. Namun, *word embedding* memiliki kelemahan yaitu menghasilkan banyak *term* yang memiliki tema sangat luas, bahkan di luar dari *query* awal. Maka, perlu digabungkan dengan *pseudo relevance feedback* sehingga dapat menyederhanakan *query* dari hasil *word embedding* disesuaikan dengan dokumen pertanyaan yang telah disiapkan oleh sistem. Penelitian ini mengusulkan metode QE dengan menggabungkan *word embedding* dengan *pseudo relevance feedback*. Penelitian ini dilakukan untuk mengatasi pertanyaan atau *query* dari pengguna yang tidak tersedia pada dokumen yang telah disiapkan.

2. State of the Art

Dalam Sistem Temu Kembali Informasi (STKI), telah banyak penelitian yang dilakukan terkait *query expansion* (QE), *word embedding*, dan *pseudo-relevance feedback*. Pada bagian ini, akan dibahas beberapa dasar dari penelitian yang telah dilakukan sebelumnya.

2.1. Query Expansion (QE)

Dalam pengambilan data pada sebuah basis data ataupun dokumen, perlu adanya masukan berupa *query* (Reshma & Remya, 2017). *Query* yang dimasukkan oleh pengguna haruslah tepat agar dapat menarik informasi yang tepat. Namun, banyak pengguna memasukkan *query* yang tidak dapat dimengerti oleh sistem. Hal ini menyebabkan informasi yang terambil menjadi kurang sesuai bahkan

tidak terambil (Reshma & Remya, 2017). *Query Expansion* (QE) merupakan sebuah metode untuk mengembangkan *query* yang dimasukkan oleh pengguna agar *query* tersebut dapat menarik informasi yang sesuai (Nie, Jiang, Ren, Sun, & Li, 2016).

Terdapat dua pendekatan QE, yaitu secara global dan lokal (Reshma & Remya, 2017). *Global Query Expansion* yaitu *query* yang dimasukkan dicari kemiripan berupa sinonim dan lain-lain pada kata dari sebuah tesaurus. Kemudian, hasil pencocokan tersebut akan ditambahkan ke dalam *query*, sehingga *query* menjadi lebih kaya. *Local Query Expansion* yaitu *query* yang dimasukkan dibandingkan dengan dokumen yang telah disediakan di awal, bukan dari tesaurus.

Banyak penelitian yang dilakukan untuk mengembangkan metode QE agar lebih akurat. Salah satunya adalah dengan memanfaatkan sinonim dari sebuah kata yang diusulkan oleh Imrandan dan Nohama (Putra, Effendi, & Arifin, 2018). Selanjutnya, dengan pendekatan morfologi, derivasi leksikal, dan kesamaan semantik semacam sinonim diusulkan oleh Pasca (Ludviani, Hayati, Arifin, & Purwitasari, 2015) yang memiliki batasan yang bergantung pada sumber informasi atau data yang digunakan.

2.2. *Word Embedding*

Word embedding mengenali distribusi makna kata yang serupa yang kemudian dikenali pada sebuah model *vector* (Şenel, Utlu, Yücesoy, Koç, & Çukur, 2018). Dengan menangkap karakteristik kata-kata, baik itu kata aslinya maupun kata yang mirip, perlu dihitung kemiripan kata yang satu dengan kata yang lain. Dengan menggunakan rumus *cosine similarity*, sistem dapat mengenali kemiripan antarkata pada sebuah vektor. *Word embedding* biasanya dipakai dalam tahap pertama untuk melakukan proses *deep learning* sebuah informasi (Young, Hazarika, Poria, & Cambria, 2018).

Telah banyak penelitian terdahulu mengenai *word embedding*. Salah satunya penelitian dari Bengio (Young, Hazarika, Poria, & Cambria, 2018) yang membuat sebuah model *Neural Network* yang belajar model dari sebuah bahasa untuk membangun sebuah *word embedding* mirip dengan konsep *Neural Network*. Hanya saja, pada model ini, digunakan *log-bilinear energy* yang menggantikan rumus tanh pada *Neural Network*.

2.3. *Pseudo-Relevance Feedback*

Pseudo relevance feedback merupakan metode untuk analisis *query* lokal otomatis dan tidak melibatkan pengguna dalam perolehan keputusannya (Wang, Fang, & Zhai, 2008). Cara kerja *pseudo relevance feedback* adalah dengan mengambil dokumen sebanyak K teratas secara sederhana dan mengasumsikan dokumen tersebut relevan. Jika pengambilan dokumen K teratas dengan alasan yang cukup, sistem akan mengambil *query* atau *terms* yang baik pula. Namun, bila pengambilan dokumen tidak bagus, terjadi pengambilan *query* atau *terms* yang buruk (Xu, Lin, Lin, Yang, & Xu, 2018) (Mothe, et al., 2015).

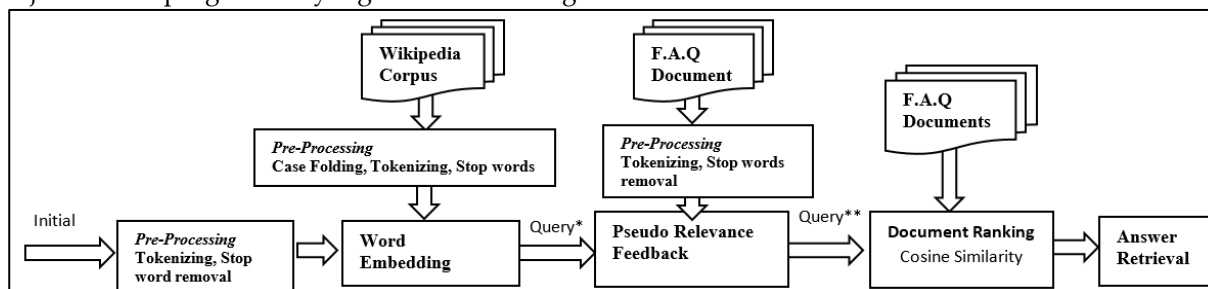
Dalam proses QE yang menggunakan *pseudo relevance feedback*, digunakan algoritma Rocchio sebagai pembentukan *vector space model* untuk menentukan *relevance feedback query*. Dalam proses pembentukan *initial query* menjadi *query* baru, algoritma Rocchio akan menentukan dokumen yang relevan dan dokumen yang tidak relevan. *Initial query* yang dimasukkan oleh pengguna akan terletak pada *centroid* seluruh dokumen, baik dokumen yang relevan maupun tidak (Buckley, Salton, & Allan, 1994) (Dierk, 1972). Berdasarkan nilai *centroid* tersebut, didapatkan *query* baru yang diambil berdasarkan dokumen yang paling relevan (Dierk, 1972).

3. Metode Penelitian

Penelitian ini mengusulkan QE menggunakan *word embedding* dan *pseudo relevance feedback*. Pengembangan metode ini dilakukan dengan bahasa pemrograman Python. Pengguna yang menginginkan informasi seputar kehamilan akan memasukkan pertanyaan ke dalam sistem. Kemudian, sistem akan melakukan *preprocessing* terhadap *query* tersebut. Setelah dilakukan *preprocessing*, dilakukan QE yang diusulkan dalam penelitian ini, yaitu *word embedding* dan *pseudo relevance feedback*. Hasil dari QE kemudian akan dipakai untuk menarik jawaban yang sesuai. Tahapan metode penelitian yang diusulkan dapat dilihat pada Gambar 1.

Dari proses QE yang telah dijelaskan sebelumnya, selanjutnya diimplementasikan ke dalam bahasa pemrograman Python dan *library* yang mendukung. Program dipasang pada aplikasi chat Line

menggunakan Line SDK. Pada aplikasi Line tersebut, dibuat sebuah akun konsultasi ibu hamil yang dapat digunakan untuk konsultasi dengan cara memberi pertanyaan kepada *chatbot*, kemudian akan dijawab oleh program *bot* yang telah dikembangkan.



Gambar 1. Tahapan metode penelitian

3.1. Data dan Dokumen Jawaban

Dalam penelitian ini, QE diterapkan pada aplikasi *chatbot*. *Query* yang dimasukkan ke dalam sistem berupa pertanyaan dari pengguna, sedangkan dokumen yang akan diambil adalah jawaban yang telah disiapkan. Implementasi pada aplikasi *chatbot* difokuskan pada pertanyaan dan jawaban seputar konsumsi makanan dan minuman pada ibu hamil. Hal ini bertujuan agar data yang diproses tidak terlalu besar. Maka, dikumpulkan data berupa dokumen yang terkait dengan konsumsi makanan dan minuman ibu hamil. Dokumen tersebut berisi jawaban-jawaban dari pertanyaan yang biasanya ditanyakan oleh ibu hamil. Dokumen tersebut didapatkan dari berbagai sumber, seperti *blog* pribadi dokter spesialis kandungan (Agung, 2011), Kumpulan Pertanyaan Seputar Ibu Hamil (Indarini, 2018), dan *website* Fakultas Kedokteran (Fitriana, 2016). Data tersebut dikumpulkan dan diringkas menjadi 16 pasang pertanyaan dan jawaban yang kemudian menjadi 16 dokumen. 16 dokumen tersebut ditetapkan label kelas untuk setiap 1 dokumen. Label dalam dokumen pertanyaan jawaban tersebut berupa *term* unik dari setiap dokumen. *Term* tersebut tidak terdapat dalam dokumen lainnya. Penentuan label nantinya digunakan untuk mencari *threshold* sebagai batas minimal kemiripan antara dokumen dan *query* yang akan dikeluarkan sebagai jawaban dari *chatbot*.

3.2. Preprocessing

Teks masukan dari pengguna berupa pertanyaan akan dilakukan *preprocessing* terlebih dahulu. Dokumen yang perlu dilakukan pada *preprocessing* yaitu *query*, *corpus* Wikipedia Bahasa Indonesia, dan dokumen *Frequently Asked Questions* (FAQ). Tahap pertama yang dilakukan yaitu *case folding*, bertujuan mengubah semua huruf dalam dokumen menjadi huruf kecil dan karakter selain huruf akan dihilangkan. Tahap kedua adalah *tokenization*, yaitu pemotongan kalimat menjadi kata-kata yang berdiri sendiri. Pada *tokenization* dilakukan pemisahan tersebut berdasarkan spasi sebagai *delimiter*. Tahapan selanjutnya dilakukan *stopword removal*, yaitu menghilangkan kata-kata tidak penting seperti “di”, “yang”, “ke” dan lain-lain. Dalam *stopword removal* juga ditambahkan beberapa *term* yang harus dieliminasi. Penambahan ini dilakukan secara manual dengan menambahkan *term* tersebut pada dokumen koleksi *term* yang harus dieliminasi.

3.3. Pseudo-Relevance Feedback

Dalam penelitian ini digunakan *word embedding* sebagai *expander query* secara global. Secara khusus, peneliti menggunakan *Word2Vec Continuous-Bag-Of-Word* (C-BOW) yang mewakili ruang vektor tiap kata atau *term* berdasarkan *co-occurrence* di dalam jendela teks. Kemiripan antara vektor istilah ditunjukkan sesuai dengan persamaan semantik (Kuzi, Shtok, & Kurland, 2016). Pada penelitian Kuzi, dkk. ini digunakan *corpus* Wikipedia Bahasa Indonesia sebagai dokumen *train*. *Training* akan dilakukan secara *offline* dan menghasilkan model yang isinya ruang vektor dari seluruh kata-kata yang ada pada *corpus*. Setelah dilakukan *preprocessing*, setelah dilakukan langsung dilakukan pengambilan kata-kata yang terdekat dari masing masing kata pada *query* dari *user*. Setiap kata akan diambil 5 yang terdekat dan disimpan sebagai 1 dokumen *query* baru.

3.4. Pseudo Relevance Feedback

Setelah melakukan QE dari *word embedding*, dihasilkan *Query**, yaitu *query* yang sudah diekspansi dari *query* awal. Pada tahap pencarian bobot TFIDF *Query** dihitung dengan seluruh dokumen jawaban. Hasil dari pembobotan TFIDF digunakan untuk mencari kedekatan antara *Query** dengan seluruh dokumen. Pada tahap *document ranking*, dilakukan penghitungan *cosine similarity* dan diurutkan berdasarkan nilai terkecil. Selanjutnya, pada tahap *Top K Document* dipilih dokumen yang memiliki nilai *cosine similarity* terkecil. Pada tahap terakhir dilakukan proses seleksi *query* dan penambahan *query* berdasarkan *K* dokumen menggunakan *Rocchio algorithm*. Pada *Rocchio algorithm*, proses pencarian akan menghasilkan dokumen yang relevan dan yang tidak. *Query** yang dimasukkan oleh pengguna berada pada *centroid* dari seluruh dokumen, baik dokumen yang relevan maupun tidak. Proses penambahan *Query** menjadi q_m berdasarkan *centroid* dokumen yang paling relevan. q_m dihasilkan dengan Persamaan 1,

$$\vec{q}_m = \alpha \vec{q}_0 + \beta \frac{1}{|D_r|} \sum_{\vec{d}_j \in D_r} \vec{d}_j - \gamma \frac{1}{|D_{nr}|} \sum_{\vec{d}_j \in D_{nr}} \vec{d}_j \quad (1)$$

dengan q_m adalah vektor *query* baru yang kemudian dinotasikan sebagai *Query***, sedangkan q_0 adalah vektor *query* awal. d_j merupakan dokumen *vector*. d_r adalah vektor dokumen relevan. d_{nr} adalah vektor dokumen nonrelevan. α adalah bobot *query* asli. β adalah bobot dokumen relevan dan γ adalah bobot dokumen nonrelevan.

3.5. Pengambilan Jawaban

Query yang telah diekspansi akan dipakai untuk mencari jawaban yang sesuai. *Query* yang berupa token *term* tersebut akan dicari kemiripan antar *term* dari masing-masing dokumen dengan menggunakan rumus *cosine similarity*. Nilai kemiripan tersebut kemudian dibandingkan dengan nilai masing-masing dokumen. Pada pengambilan jawaban ditetapkan *threshold* yang berfungsi membatasi kemiripan jawaban yang akan diambil. Jika dokumen yang memiliki nilai *cosine* terbesar tetapi di bawah *threshold*, dokumen jawaban tidak akan ditarik. Namun, jika dokumen jawaban tersebut memiliki nilai *cosine* berada di atas *threshold*, dokumen tersebut akan diambil, kemudian akan ditampilkan kepada pengguna sebagai jawaban dari pertanyaan yang telah dimasukkan oleh pengguna.

3.6. Precision, Recall, dan Harmonic Mean

Pada penelitian ini, digunakan *recall* dan *precision* untuk mengukur seberapa baik performa QE dalam memperkaya *query* dengan kriteria *confusion matrix*. Kriteria *True Positive* (TP) untuk yang jawaban ada dan relevan dengan pertanyaan, kriteria *False Positive* (FP) untuk yang jawaban ada tetapi tidak relevan dengan pertanyaan, kriteria *False Negative* (FN) untuk yang tidak ada jawaban relevan yang diambil, kriteria *False Negative* (FN) untuk yang tidak ada jawaban yang tidak relevan yang diambil. Namun, *precision* dan *recall* adalah unsur yang saling berlawanan atau *trade-off*, maka diperlukan evaluasi yang dapat menoleransi *trade-off* tersebut dengan perhitungan *F-measure* yang merupakan *weighted harmonic mean* dari *precision* dan *recall*. Titik nilai *precision* dan *recall* mencapai keseimbangan jika perhitungan *F-measure* mencapai 70% (Domarco & Iswari, 2017). Nilai *recall*, *precision* dan *F-measure* (F1) dapat dilihat pada Persamaan 2, Persamaan 3, dan Persamaan 4.

$$\text{Recall} = \frac{TP}{TP+FN} \quad (2)$$

$$\text{Precision} = \frac{TP}{TP+TN} \quad (3)$$

$$F - \text{Measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4)$$

4. Hasil dan Pembahasan

Metode QE yang diusulkan telah diimplementasikan ke dalam aplikasi *chatbot* yang kemudian dilakukan uji coba. Tabel 1 adalah contoh hasil proses dari setiap tahap pada sistem. Pertama, pengguna memasukan pertanyaan sebagai *query*. Selanjutnya, sistem akan melakukan *preprocessing* kata berupa *stopword*, menghilangkan simbol dan mengubah pertanyaan menjadi token kata. Berikutnya, token tersebut dimasukkan ke proses QE. Pada tahap pertama, yaitu *word embedding*, masing-masing dari token dicari kemiripan katanya dan diambil 5 kata paling mirip kemudian ditambahkan sebagai *query* awal.

Tabel 1. Proses menghasilkan QE

Query	Pre-processing	Word Embedding	Pseudo feedback relevance	Expanded query
bolehkah ibu hamil makan tiram?	['ibu', 'hamil', 'makan', 'tiram']	ayah ibunya mertua adik orangtua menyusui kehamilan bayinya haid persalinan makannya sarapan minum memasak tidur udang bandeng ketam cakalang kerang ibu hamil makan tiram	kerang sarapan ibu hamil makan	kerang tiram sarapan ibu hamil makan

Tahap *pseudo relevance feedback* yaitu mencari kemiripan kata pada *query* hasil *word embedding* dengan *term* pada dokumen pertanyaan dan jawaban. Hasil dari *pseudo relevance feedback* dipakai untuk mencari kemiripan dokumen. *Threshold* yang digunakan untuk batas bawah kemiripan antara *query* dan dokumen adalah nilai bobot *cosine similarity*. *Threshold* dihasilkan dari hasil observasi terhadap terhadap nilai *cosine similarity*. Dengan menggunakan label kelas dokumen sebagai *query*, didapat bobot *cosine*, lalu hasilnya disortir. Bobot *cosine* yang terkecil akan menjadi *threshold*. Jika pertanyaan yang diajukan oleh pengguna di bawah *threshold*, akan dibalas dengan pertanyaan balik: "maaf bunda sebenarnya tanya apa? Silakan tanya dengan spesifik". Hasil uji coba *threshold* dari setiap metode dapat dilihat pada Tabel 2 pada kolom *threshold cosine*. Hasil observasi menunjukkan *threshold* yang berbeda-beda disebabkan setiap dokumen pertanyaan dan jawaban memiliki keunikan masing-masing, sehingga tidak bisa disamakan ambang batasnya. Jika semua *threshold* disamakan, sistem akan memberikan dokumen yang tidak relevan dengan *query*.

Untuk melakukan uji coba, disiapkan 10 pertanyaan yang terdiri atas 4 pertanyaan dari responden yang sedang hamil, 4 pertanyaan yang menggunakan bahasa sehari-hari, dan 2 pertanyaan yang tidak terdapat dalam dokumen jawaban *chatbot*. Sepuluh pertanyaan akan ditanyakan kepada *chatbot*. Hasil pengujian sistem *chatbot* dengan QE menggunakan *word embedding* dan *pseudo relevance feedback* dapat dilihat pada Tabel 2. Ditampilkan pula hasil pengujian sistem *chatbot* tanpa menggunakan QE, QE dengan *word embedding*, dan QE dengan *pseudo relevance feedback*.

Tabel 2. Hasil pengujian sistem

Metode Query Expansion (QE)	Threshold cosine	Recall (%)	Precision (%)	F-measure (%)
WE + Pseudo Relevance Feedback	0.21	100	70	82
Word Embedding (Rattinger, Goff, & Guetl, 2018)	0,11	100	20	33
Pseudo Relevance Feedback (Vaidyanathan, Das, & Srivastava, 2015)	0.14	83	71	77
Tanpa QE (Singh, Paste, Shinde, Patel, & Mishra, 2018)	0.15	100	44	62

Berdasarkan hasil pengujian didapatkan bahwa penggunaan *word embedding* dan *pseudo relevance feedback* sukses dalam melakukan QE terbukti dengan nilai F1 mencapai 82,35 %. Jika dihitung akurasi mencapai 70%, hasil tersebut lebih baik daripada *chatbot* tanpa menggunakan QE yang dilakukan Singh, dkk. (2018) yang hanya mencapai akurasi 68,51%. Terdapat beberapa kekurangan dalam sistem, yaitu *query* yang dimasukkan oleh pengguna yang diproses ke dalam *word embedding* tidak dapat mencari kemiripan kata berdasarkan makna katanya. *Word embedding* yang digunakan hanya mencari kemiripan kata berdasarkan kata yang muncul bersama dalam satu dokumen atau *co-occurrence*. Sebagai contoh, jika diinputkan *query* "makan" ke dalam sistem, *word embedding* akan mencari kata yang sering muncul bersama dengan kata "makan" di dalam *corpus*. Maka, yang muncul adalah kata: "minum", "sarapan", "memasak", "menyantap", "tidur" yang menyebabkan berkurangnya makna kata "makan".

5. Kesimpulan

Berdasarkan hasil penelitian penggabungan *word embedding* dan *pseudo relevance feedback* sebagai metode QE pada aplikasi *chatbot* untuk konsumsi makanan dan minuman ibu hamil, disimpulkan bahwa *word embedding* mencari kemiripan antarkata yang memiliki jumlah kemunculan bersama dalam satu dokumen. Namun, *word embedding* tidak dapat mencari kemiripan kata berdasarkan makna kata atau

keterkaitan dengan perbedaan makna kata. Hal ini tentunya berseberangan dengan tujuan QE yang memperkaya *query* dengan kata-kata yang mirip berdasarkan makna kata. Dengan menggunakan *pseudo relevance feedback*, QE hasil *word embedding* dipilih kata yang terdapat dalam dokumen jawaban. Hasil uji coba menunjukkan bahwa QE menggunakan *word embedding* dan *pseudo relevance feedback* pada *chatbot* dapat mengatasi *query* masukan dari pengguna yang ambigu dan alami sehingga dapat memberikan jawaban yang relevan kepada pengguna.

7. Referensi

- Agung, G. (2011, April 20). *17 Pertanyaan Yang Sering Ditanyakan Ibu Hamil*. Retrieved from Dr. Gregorius Agung, SpOG: <http://greg-spog.com/kebidanan-kandungan/17-pertanyaan-yang-sering-ditanyakan-ibu-hamil/>
- Buckley, C., Salton, G., & Allan, J. (1994). The Effect of Adding Relevance Information in a Relevance Feedback Environment. *SIGIR '94* (pp. 292-300). London: Springer.
- Dalpiaz, F., Ferrari, A., Franch, X., & Palomares, C. (2018). Natural Language Processing for Requirements Engineering: The Best Is Yet to Come. *IEEE Software*, 35(5), 115-119.
- Dierk, S. F. (1972). The SMART retrieval system: Experiments in automatic document processing. *IEEE Transactions on Professional Communication*, PC-15(1), 17.
- Domarco, D., & Iswari, N. M. (2017). Rancang Bangun Aplikasi Chatbot Sebagai Media Pencarian Informasi Anime Menggunakan Regular Expression Pattern Matching. *ULTIMATICS: Jurnal Ilmu Teknik Informatika*, 9(1), 19-24.
- Fitriana, D. A. (2016, September 1). *Gizi Seimbang Ibu Hamil*. Retrieved from Jurusan Gizi Fakultas Kedokteran Universitas Brawijaya: <http://gizi.fk.ub.ac.id/gizi-seimbang-ibu-hamil/>
- Indarini, N. (2018, Juli 17). *Kumpulan Pertanyaan Seputar 'Bolehkah Ibu Hamil Makan...'*. Retrieved from HaiBunda.com: <https://www.haibunda.com/kehamilan/20180716143654-49-23095/kumpulan-pertanyaan-seputar-bolehkah-ibu-hamil-makan>
- Kuzi, S., Shtok, A., & Kurland, O. (2016). Query Expansion Using Word Embeddings. *CIKM '16 Proceedings of the 25th ACM International on Conference on Information and Knowledge Management* (pp. 1929-1932). Indianapolis, Indiana, USA: ACM.
- Lee, H.-Y., & Lee, L.-S. (2014). Improved Semantic Retrieval of Spoken Content by Document/Query Expansion with Random Walk Over Acoustic Similarity Graphs. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(1), 80-94.
- Liu, Q., Huang, H., Lut, J., Gao, Y., & Zhang, G. (2017). Enhanced word embedding similarity measures using fuzzy rules for query expansion. *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. Naples, Italy: IEEE.
- Ludviani, R., Hayati, K. F., Arifin, A. Z., & Purwitasari, D. (2015). Optimasi Pembobotan pada Query Expansion dengan Term Relatedness to Query-Entropy based (TRQE). *Jurnal Buana Informatika*, 6(3), 203-212.
- Mothe, J., Savoy, J., Kamps, J., Pinel-Sauvagnat, K., Jones, G., Juan, E. S., . . . Ferro, N. (2015). Experimental IR Meets Multilinguality, Multimodality, and Interaction. *6th International Conference of the CLEF Association (CLEF'15)*. Toulouse, France: Springer.
- Nie, L., Jiang, H., Ren, Z., Sun, Z., & Li, X. (2016). Query Expansion Based on Crowd Knowledge for Code Search. *IEEE Transactions on Services Computing*, 9(5), 771-783.
- Ooi, J., Ma, X., Qin, H., & Liew, S. C. (2015). A survey of query expansion, query suggestion and query refinement techniques. *2015 4th International Conference on Software Engineering and Computer Systems (ICSECS)*. Kuantan, Malaysia: IEEE.
- Putra, F. N., Effendi, A., & Arifin, A. Z. (2018). Pembobotan Kata berdasarkan Kluster untuk Peringkasan Otomatis Multi Dokumen. *Jurnal Linguistik Komputasional*, 1(1), 17-22.
- Rattinger, A., Goff, J.-M. L., & Guetl, C. (2018). Local Word Embeddings for Query Expansion based on Co-Authorship and Citations. *BIR 2018 Workshop on Bibliometric-enhanced Information Retrieval* (pp. 46-53). Grenoble, France: CEUR-WS.
- Reshma, E. U., & Remya, P. C. (2017). A review of different approaches in natural language interfaces to databases. *2017 International Conference on Intelligent Sustainable Systems (ICISS)*. Palladam, India: IEEE.

- Şenel, L. K., Utlü, İ., Yücesoy, V., Koç, A., & Çukur, T. (2018). Semantic Structure and Interpretability of Word Embeddings. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(10), 1769-1779.
- Singh, R., Paste, M., Shinde, N., Patel, H., & Mishra, N. (2018). Chatbot using TensorFlow for small Businesses. *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*. Coimbatore, India: IEEE.
- Vaidyanathan, R., Das, S., & Srivastava, N. (2015, February 18). *Query Expansion Strategy based on Pseudo Relevance Feedback and Term Weight Scheme for Monolingual Retrieval*. Retrieved from arXiv: <https://arxiv.org/abs/1502.05168>
- Wang, X., Fang, H., & Zhai, C. (2008). A Study of Methods for Negative Relevance Feedback. *SIGIR '08 Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 219-226). Singapore: ACM.
- Xu, B., Lin, H., Lin, Y., Yang, L., & Xu, K. (2018). Improving Pseudo-Relevance Feedback With Neural Network-Based Word Representations. *IEEE Access*, 6, 62152-62165.
- Yan, R., & Gao, G. (2017). Pseudo-Based Relevance Analysis for Information Retrieval. *2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI)*. Boston, MA, USA: IEEE.
- Young, T., Hazarika, D., Poria, S., & Cambria, E. (2018). Recent Trends in Deep Learning Based Natural Language Processing [Review Article]. *IEEE Computational Intelligence Magazine*, 13(3), 55-75.