Ekspansi *Query* Menggunakan Word2Vec pada Pencarian Artikel Ilmiah

1st Bayu Tresna Ariesta
Fakultas Informatika
Universitas Telkom
Bandung, Indonesia
bayutresna@students.telkomuniversity.ac.id

2nd Ade Romadhony
Fakultas Informatika
Universitas Telkom
Bandung, Indonesia
aderomadhony@telkomuniversity.ac.id

3rd Hasmawati
Fakultas Informatika
Universitas Telkom
Bandung, Indonesia
hasmawati@telkomuniversity.ac.id

***Abstrak* — Pencarian informasi di internet sudah menjadi kebutuhan bagi sebagian besar orang dengan keperluannya masing masing, khususnya untuk para pelajar. Untuk pencarian artikel ilmiah sendiri bisa menggunakan mesin pencari google scholar untuk memudahkan pencarian agar lebih spesifik untuk artikel ilmiah. Meski menggunakan mesin pencari google scholar, informasi yang disediakan oleh mesin pencari tersebut masih terbilang banyak dan dibutuhkan kata kunci tertentu agar bisa mencari artikel ilmiah yang sesuai dengan yang diinginkan. Maka penggunaan ekspansi query dirasa sangat tepat untuk membantu pengguna dalam menentukan kata kunci yang tepat untuk melakukan pencarian. Pada penelitian ini dilakukan percobaan untuk menggunakan metode word embedding word2vec untuk melakukan ekspansi query dan melakukan dua skenario pencarian artikel ilmiah dengan menggunakan mesin pencari google scholar. Dataset yang digunakan untuk membuat model word2vec menggunakan data dari repository WING-NUS/scisumm-corpus. Nilai total akurasi yang didapat pada hasil pencarian skenario pertama sebesar 89,9% sedangkan untuk nilai total akurasi untuk hasil pencarian skenario kedua sebesar 76,1%.**

***Kata kunci*— word2vec, ekspansi query, IR, pencarian artikel ilmiah**

I. PENDAHULUAN

A. Latar Belakang

Pencarian informasi di internet sudah menjadi kebutuhan bagi sebagian besar orang dengan berbagai tujuan, baik untuk mencari berita terkini, artikel ilmiah, maupun hiburan. Karena sudah menjadi kebutuhan, manusia dituntut untuk dapat memenuhi kebutuhan tersebut, salah satunya adalah dengan mencarinya di internet [1]. Untuk pencarian artikel ilmiah sendiri bisa menggunakan mesin pencari google scholar untuk memudahkan pencarian agar lebih spesifik untuk artikel ilmiah. Meski menggunakan mesin pencari google scholar, informasi yang disediakan masih terbilang banyak dan membutuhkan kata kunci yang tepat untuk mendapatkan informasi yang sesuai dan relevan.

Ekspansi query adalah salah satu teknik dari query reformulation yang dilakukan dengan menambahkan term kedalamnya [3]. Untuk penelitian ini term yang ditambahkan adalah kata-kata yang memiliki kemiripan arti yang didapat dari hasil word embedding menggunakan word2vec. Ekspansi query berkaitan erat dengan information retrieval atau IR. Karena ekspansi query memiliki peran penting dalam proses melakukan information retrieval [4]. Dengan melakukan pencarian pada mesin pencari google scholar sebagai proses information retrieval, maka penelitian mengenai ekspansi query dengan word2vec ini dapat dilaksanakan.

Proses ekspansi query sendiri dapat menggunakan berbagai metode, salah satunya word embedding. Bahkan pada penelitian yang dilakukan oleh Alfredo Silva dan Marcelo Mendoza dimana mereka meneliti bagaimana metode word embedding dapat meningkatkan performa dari ekspansi query [4]. Pada penelitian tersebut dilakukan percobaan ekspansi query dengan menggunakan pendekatan inverse document frequency dan average word embedding atau IDF-AWE, sebuah representasi vector untuk query berbasis AQE (automated query expansion) lalu ditambahkan dengan berbagai metode word embedding seperti glove, word2vec, FastText, dll [4]. Hasil pada penelitian tersebut menunjukan bahwa metode word embedding memberikan hasil yang lebih baik ketika digunakan bersama AQE dan IDF-AWE. Lalu penelitian yang dilakukan Claudio dan Giovanni memberi penjelasan lebih lanjut mengenai penerapan IR seperti interactive query refinement, relevance feedback, word sense disambiguation in IR, dan search result clustering [6]. Beserta penerapan lainnya pada penggunaan automated query expansion atau AQE seperti question answering, multimedia information retrieval, information filtering, dan cross-language information retrieval [6]. Penelitian tersebut memberikan system ranking untuk menjadi variable pembanding untuk dapat mencari nilai similarity antara query dan dokumen terkait.

Kebanyakan dari penelitian mengenai ekspansi query menggunakan AQE sebagai basis awalnya, selanjutnya menambahkan proses lain seperti word embedding dan sebagainya. Seperti pada penelitian yang dilakukan Saar Kuzi dan rekannya dimana mereka menggunakan word2vec untuk melakukan ekspansi query [7]. Pada penelitian ini, akan

menggunakan word2vec untuk ekspansi query, dan menggunakan mesin pencari google scholar untuk melakukan dua skenario pencarian untuk proses information retrieval. selanjutnya melakukan survey pada lima orang partisipan untuk menilai tingkat relevansi pada hasil skenario pencarian untuk relevance feedback yang digunakan sebagai metode penilaian dari kualitas query. Dataset yang digunakan diambil dari repository WING-NUS/scisumm-corpus yang dimana terdapat 1000 set dokumen artikel ilmiah yang sudah teranotasi dari scisummnet [8].

Tujuan penelitian ini yaitu untuk mengetahui performansi dari penggunaan ekspansi query dengan menggunakan word2vec dalam melakukan pencarian artikel ilmiah di internet.

II. KAJIAN TEORI

A. Studi Terkait

Penelitian mengenai ekspansi query telah banyak dilakukan bersamaan dengan penelitian mengenai information retrieval. Seperti pada penelitian yang dilakukan oleh Claudio dan Giovanni dimana mereka menjelaskan secara detail mengenai perkembangan ekspansi query [6]. Pada penelitian tersebut menjelaskan bahwa dalam sistem ranking pada *information retrieval*, *similarity* antara query dan dokumen dapat direpresentasikan pada persamaan (1)

$$sim(q,d) = \sum_{t \in q \cap d} wt,q \cdot wt,d \quad (1)$$

dimana wt,q dan wt,d adalah weight suatu term t terhadap query q dan dokumen d [6]. Dari perasamaan (1) bisa diasumsikan hasil output dari AQE adalah q' yang terbentuk dari hasil ekspansi dengan weight tertentu. Sehingga bisa didapat persamaan (2).

$$sim(q',d) = \sum_{t \in q' \cap d} w't,q' \cdot wt,d \quad (2)$$

Dari persamaan (2) dapat disimpulkan bahwa nilai *similarity* bergantung pada query yang digunakan. Sehingga query bisa diubah ataupun dilakukan reformulasi seperti ekspansi query.

Pada penelitian [9] dilakukan penelitian mengenai penerapan query expansion menggunakan *tools* berbasis NLP dan model *embedding* word2vec untuk berpartisipasi pada TREC 2018. Penelitian tersebut dilakukan dengan menggunakan dataset dari *washington post corpus* sebanyak 50 topik. Corpus tersebut berisikan berita, artikel, dan blog yang ditulis dan dipublikasikan oleh *washington post* dari bulan januari tahun 2012 sampai agustus 2017.

Pada penelitian [10] dilakukan penelitian mengenai ekspansi query dengan menggunakan *locally trained word embedding*. Penelitian tersebut dilakukan untuk mengetahui perbandingan antara *embedding* lokal dalam menangkap bahasa untuk spesifik topik tertentu [10]

Penelitian tersebut menghasilkan sebuah simpulan dengan menggunakan embedding lokal memberikan hasil yang lebih baik pada beberapa skenario khususnya pada nilai recall jika dibandingkan dengan embedding global. Meski begitu, hasil tersebut belum ada penjelasan yang kuat untuk hasil tersebut. Sehingga masih diperlukan analisis secara empiris dan teori lebih lanjut.

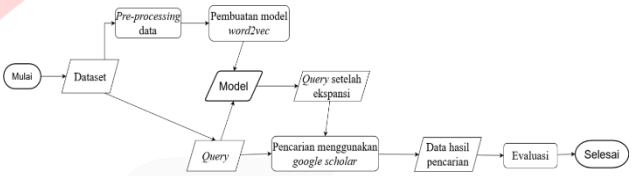
Pada penelitian yang dilakukan oleh Mawloud melakukan penggunaan mekanisme AQE dan pseudo re-ranking MVRA untuk meningkatkan hasil pencarian pada mesin pencari google scholar [11]. Dengan menggunakan MVRA untuk melakukan ekspansi query sehingga didapatkan query terbaru yang lebih sesuai.

Dari beberapa penilitian yang telah digunakan, penggunaan word2vec untuk ekspansi query memberikan hasil yang sangat baik, Dari beberapa penilitian yang telah digunakan, penggunaan word2vec untuk ekspansi query memberikan hasil yang sangat baik, tetapi hanya berfokus pada penilaian yang berbasis sistem. Maka dari itu penelitian ini akan menggunakan pendekatan kepada user untuk memberikan penilaian sebagai bentuk dari *relevance feedback*.

III. METODE

A. Sistem yang Dibangun

1. Alur Penelitian



GAMBAR 1.
Gambaran umum system

Gambar 1 menjelaskan alur penelitian yang dilaksanakan pada penelitian ini. Dimulai dari melakukan pengumpulan dataset, sampai pembuatan model word2vec. Lalu selanjutnya dilakukan dua skenario pencarian dan hasil skenario pencarian digunakan untuk mendapatkan penilaian relevansi dari partisipan.

2. Dataset

Dari 1000 set dokumen artikel yang ada pada *repository* WING-NUS/scisumm-corpus, yang digunakan untuk pembuatan model word2vec sebanyak 520 file dokumen artikel yang terdiri atas 412 judul artikel yang berkaitan dengan NLP.

3. Preprocessing Data

Preprocessing data adalah suatu teknik untuk mengubah data mentah dalam format yang berguna dan efisien [12]. Pada penelitian ini dilakukan beberapa metode data cleaning, stopword removal, tokenization, dan lemmatization. Data cleaning dilakukan untuk menghilangkan missing value atau data = null pada dataset [12]. Stopword removal dilakukan untuk mnghilangkan kata-kata umum yang tidak memiliki arti seperti "is", "a", "are", "the", dll [13]. lemmatization dilakukan untuk mengubah kata-kata yang ada pada dataset. lemmatization sesuai dengan linguistic rule [14].

4. Pengumpulan Query

Pengumpulan query dilakukan dengan menggunakan bagian dari judul-judul artikel yang ada pada dataset. Selanjutnya judul-judul tersebut dilakukan pemotongan untuk diambil inti dari artikel tersebut. Berikut contoh dari

perbandingan antara judul artikel dengan querynya pada table 1.

TABEL 1.
Contoh Judul Artikel yang dijadikan Query

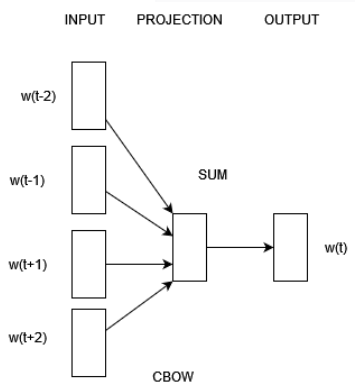
No	Judul Artikel	Query
1	A Cross-lingual Annotation Projection Approach for Relation Detection	annotation projection approach
2	Bilingual Lexicon Extraction from Comparable Corpora Using Label Propagation	bilingual lexicon extraction

Query tersebut digunakan sebagai kata kunci yang digunakan untuk melakukan pencarian pada skema pencarian menggunakan google scholar. Banyak query yang digunakan untuk penelitian ini adalah sebanyak 101 query.

5. Word2vec

Word embedding adalah sebuah representasi kata yang memungkinkan untuk suatu kata yang memiliki kesamaan makna dapat direpresentasikan. Word embedding merupakan pendekatan untuk merepresentasikan kata dan dokumen yang bisa menjadi salah satu kunci dalam menyelesaikan permasalahan natural language processing dan deep learning. [15].

Word2vec adalah salah satu metode dalam word embedding yang dimana dapat mengelompokan beberapa vektor dari kata yang memiliki kemiripan. Word2vec dapat memberikan estimasi kata yang memiliki kesamaan arti dengan baik jika diberikan dataset yang cukup luas [16].



GAMBAR 2.
Ilustrasi Arsitektur Word2vec CBOW

Gambar 2 menjelaskan mengenai arsitektur word2vec CBOW yang dimana arsitektur CBOW memprediksi kata berdasarkan dari input. Maka untuk penelitian ini menggunakan CBOW untuk proses ekspansi query karena untuk mencari kata yang memiliki kesamaan arti dari query yang akan diekspansi.

Pada penelitian ini juga, penulis membuat model word2vec sendiri dengan menggunakan dataset yang sudah dijelaskan pada segmen sebelumnya sebagai input untuk membuat model word2vec.

6. Ekspansi Query

Ekspansi query digunakan untuk meningkatan ketepatan dalam melakukan pencarian, ide dasarnya adalah dengan menggunakan hasil pada query awal untuk memformulasikan ulang query dan melakukan pencarian kedua untuk mendapatkan hasil yang lebih presisi [17]. Untuk dapat melakukan ekspansi query, setiap query harus diekstrak, lalu untuk setiap kata kunci, sinonim dan akronim akan terpilih secara otomatis [18]. Sehingga pada saat melakukan pencarian, hasil yang didapat bisa lebih sesuai dengan apa yang pengguna harapkan.

7. Information Retrieval

Pada penelitian [6] menjelaskan mengenai persamaan (1) dan (2). Dimana nilai *similarity* didapat bergantung pada *weight term* query dan dokumen. Pada penelitian ini, *weight term* dokumen dapat dianggap 1 karena pada penelitian ini penulis menggunakan mesin pencari *google scholar*. Sehingga dapat disimpulkan persamaan yang digunakan adalah sebagai berikut.

$$sim(q,n) = \sum_{t \in q \cap d} wt,q \quad (3)$$

n = artikel ke-n
q = query yang digunakan
wt,q = *weight term* suatu query

Sehingga dapat disimpulkan tingkat *similarity* dalam pencarian bergantung pada query yang digunakan. Dan karena penelitian ini menggunakan metode survey untuk mendapatkan *relevance feedback* pada hasil pencarian maka nilai *similarity* dapat digunakan sebagai alat untuk menentukan kualitas performansi query tersebut dalam pencarian. Maka bisa didapat persamaan sebagai berikut.

$$wt,q = \sum_{n=1}^{10} sim(q,n) \quad (4)$$

n = artikel ke-n
q = query yang digunakan
wt,q = *weight term* suatu query

Dimana pada persamaan (4) n adalah banyaknya dokumen pada hasil pencarian. Dimana pada penelitian ini menggunakan 10 dokumen teratas dari hasil pencarian. Sehingga nilai *weight term* untuk query bisa didapat dari nilai *similarity* dari hasil pencarian. Menurut penelitian [6], rumus persamaan (4) tersebut berlaku juga untuk query yang dilakukan ekspansi, sehingga didapat persamaan sebagai berikut.

$$w't,q' = \sum_{n=1}^{10} sim(q',n) \quad (5)$$

n = artikel ke-n
q' = query yang digunakan
w't,q' = *weight term* suatu query yang sudah diekspansi

Dimana *weight term* untuk query yang dilakukan ekspansi juga nilainya bergantung pada hasil pencariannya. Maka pada untuk menentukan performansi dari penggunaan word2vec untuk ekspansi query pada penelitian ini adalah dengan menilai dari hasil pencariannya yang menggunakan mesin pencari *google scholar*.

IV. HASIL DAN PEMBAHASAN

A. Evaluasi

1. Preprocessing Data

Preprocessing data dilakukan dengan melakukan tokenization, lemmatizing, dan mengubah setiap huruf menjadi lowercase. Hal ini dilakukan untuk mengurangi kata-kata yang tidak diperlukan ataupun yang tidak memiliki arti. Berikut adalah contoh proses preprocessing data pada penelitian ini.

TABEL 2.
Contoh Proses *Preprocessing* Data

Inisialisasi	Tokenization	Lemmatizing	lowercase
Merged\nproceedings of the 2010 conference on empirical methods in natural language processing	['merged', 'proceedings', 'of', 'the', '2010', 'conference', 'on', 'empirical', 'methods', 'in', 'natural', 'language', 'processing',	'merged', 'proceeding', '', '', '2010', 'conference', '', 'empirical', 'method', '', 'natural', 'language', 'processing'	'merged', 'proceeding', '', '', '2010', 'conference', '', 'empirical', 'method', '', 'natural', 'language', 'processing'

2. Model Word2vec

Pembuatan model word2vec dilakukan dengan melakukan training pada artikel-artikel yang ada pada dataset dengan menggunakan parameter yang dijelaskan pada tabel 3.

TABEL 3.
Parameter yang Digunakan dalam Pembuatan Model Word2vec

Parameter Word2vec	Nilai	Fungsi
min_count	1	untuk menentukan berapa kali suatu kata harus muncul agar dimasukan kedalam model
window	1	untuk menentukan jarak berapa N pada input ketika melakukan training atau menentukan berapa banyak tetangga yang digunakan untuk pengecekan
sample	0e-5	untuk batas konfigurasi downsampling pada kata dengan frekuensi tertinggi
alpha	0.03	inisiasi learning rate
min_alpha	0.0007	nilai learning rate akan turun sampai ke nilai pada min_alpha selama proses training
negative	20	untuk menentukan negative sampling untuk mengurangi noise

Setelah membuat model word2vec, selanjutnya dilakukan training pada data yang dilakukan sebanyak dua kali perulangan untuk mendapatkan hasil training datanya.

3. Ekspansi Query

Ekspansi query yang dilakukan adalah dengan memanfaatkan fungsi *wv.most_similar* untuk mendapatkan kata-kata yang memiliki kedekatan secara vector. Berikut adalah contoh ketika query dimasukan kedalam funsgi *wv.most_similar* pada tabel 4 sebagai berikut.

TABEL 4.
Contoh Hasil Penggunaan Fungsi *wv.most_similar*

Keyword	**most similar 1**	**most similar 2**	**most similar 3**	**most similar 4**	**most similar 5**
annotation projection approach	('grammar', 0.9863209724426267),	('robust', 0.9851688146591187),	('disambiguator', 0.9835725426673889),	('accurate', 0.9823119044303894),	('svm', 0.9817441701889038),
bayesian inference	('naive', 0.9100314378738403),	('bayes', 0.8905052542686462),	('nearestneighbor', 0.8856500387191772),	('simple', 0.8853764533996582),	('twostage', 0.8761579990386963),
bayesian model	('class based', 0.8400786519050598),	('mixturemodel', 0.8276405930519104),	('latentvariable', 0.8269985318183899),	('contextfree', 0.8263013958930969),	('unified', 0.8176162838935852),

Selanjutnya dilakukan pemilihan kembali dari kelima kata dari hasil penggunaan fungsi *wv.most_similar* untuk dilakukan ekspansi query. Sehingga menghasilkan query seperti pada tabel 5.

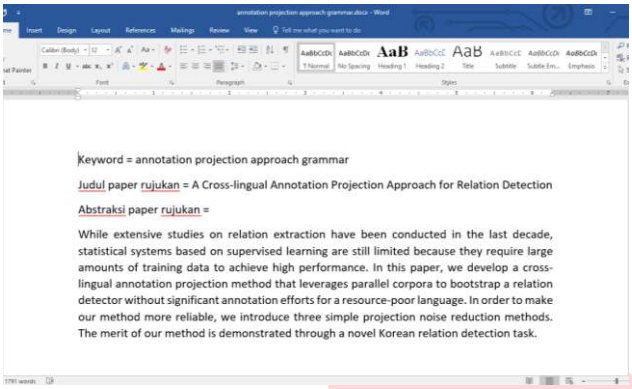
TABEL 5.
Perbandingan Query Sebelum Ekspansi dan Setelah Ekspansi

Query Sebelum Ekspansi	Query Setelah Ekspansi
annotation projection approach	annotation projection approach grammar
bayesian inference	bayesian inference naive
bayesian model	bayesian model class based
bilingual lexicon extraction	bilingual lexicon extraction generator

4. Skenario Pencarian

Skenario pencarian pada penelitian ini dilakukan sebanyak dua kali, dimana pencarian pertama dilakukan menggunakan query sebelum ekspansi dan untuk skenario kedua dilakukan menggunakan query setelah ekspansi. Pencarian dilakukan menggunakan mesin pencari *google scholar* untuk mendapatkan sepuluh artikel teratas dari hasil pencariannya. Lalu sepuluh artikel teratas dari pencarian tersebut diambil judul, abstrak atau ringkasan, dan *screenshoot* dari artikel tersebut beserta artikel rujukannya

kemudian dikumpulkan dalam satu *file Microsoft word*. Berikut adalah contoh hasil dari pencarian yang ditunjukan pada gambar 3 dan gambar 4.



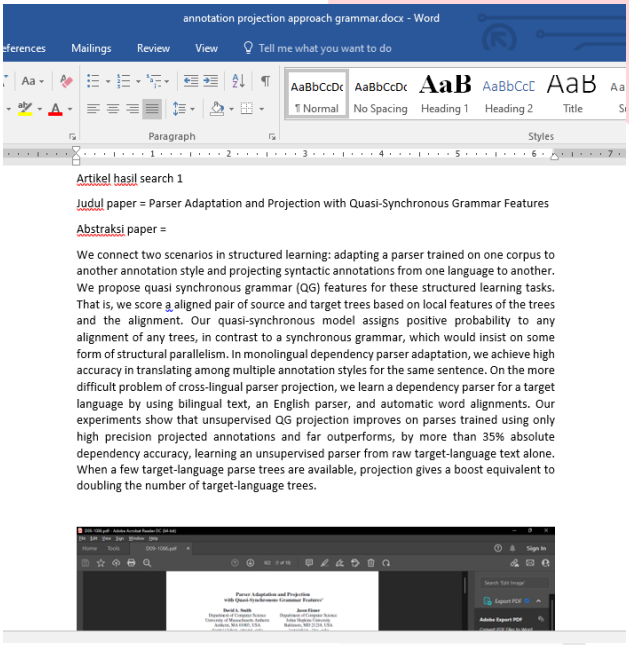
Keyword = annotation projection approach grammar

Judul paper rujukan = A Cross-lingual Annotation Projection Approach for Relation Detection

Abstraksi paper rujukan =

While extensive studies on relation extraction have been conducted in the last decade, statistical systems based on supervised learning are still limited because they require large amounts of training data to achieve high performance. In this paper, we develop a cross-lingual annotation projection method that leverages parallel corpora to bootstrap a relation detector without significant annotation efforts for a resource-poor language. In order to make our method more reliable, we introduce three simple projection noise reduction methods. The merit of our method is demonstrated through a novel Korean relation detection task.

GAMBAR 3.
Contoh Data Hasil Pencarian Bagian Artikel Rujukan



annotation projection approach grammar.docx - Word

Artikel hasil search 1

Judul paper = Parser Adaptation and Projection with Quasi-Synchronous Grammar Features

Abstraksi paper =

We connect two scenarios in structured learning: adapting a parser trained on one corpus to another annotation style and projecting syntactic annotations from one language to another. We propose quasi synchronous grammar (QG) features for these structured learning tasks. That is, we score a aligned pair of source and target trees based on local features of the trees and the alignment. Our quasi-synchronous model assigns positive probability to any alignment of any trees, in contrast to a synchronous grammar, which would insist on some form of structural parallelism. In monolingual dependency parser adaptation, we achieve high accuracy in translating among multiple annotation styles for the same sentence. On the more difficult problem of cross-lingual parser projection, we learn a dependency parser for a target language by using bilingual text, an English parser, and automatic word alignments. Our experiments show that unsupervised QG projection improves on parses trained using only high precision projected annotations and far outperforms, by more than 35% absolute dependency accuracy, learning an unsupervised parser from raw target-language text alone. When a few target-language parse trees are available, projection gives a boost equivalent to doubling the number of target-language trees.

GAMBAR 4.
Contoh Data Hasil Pencarian

Lalu setelah dikumpulkan semua hasil pencarian selanjutnya dilakukan penilaian yang dilakukan oleh lima orang partisipan. Latar belakang setiap partisipan dijelaskan pada tabel 6.

TABEL 6.
Latar Belakang Partisipan

Partisipan ke-n	Keterangan
1	mahasiswa tingkat akhir dari jurusan non-informatika yang sedang mengerjakan skripsi atau tugas akhir
2	mahasiswa tingkat akhir dari jurusan non-informatika yang sedang mengerjakan skripsi atau tugas akhir
3	mahasiswa tingkat akhir dari jurusan informatika yang sedang mengerjakan tugas akhir
4	mahasiswa tingkat akhir dari jurusan informatika yang sedang mengerjakan tugas akhir
5	mahasiswa tingkat akhir dari jurusan informatika yang sedang mengerjakan tugas akhir

Setiap partisipan akan memberi nilai pada setiap artikel dari hasil pencarian dengan nilai 0 jika artikel tersebut tidak relevan baik dari kata kunci maupun artikel rujukan dan memberikan nilai 1 jika artikel tersebut relevan. Sehingga setiap file word akan mendapat maksimal nilai 10 dan nilai minimal 0 tergantung dari berapa banyak artikel yang relevan dari setiap hasil pencarian tersebut. Berikut adalah contoh bentuk penilaian yang diberikan oleh partisipan pada tabel 7.

TABEL 7.
Contoh Hasil Penilaian Oleh Partisipan

No	Keyword (nama file wordnya)	Partisipan 1	Partisipan 2	Partisipan 3	Partisipan 4	Partisipan 5
1	annotation projection approach	10	9	8	7	9
2	bayesian inference	10	9	10	10	10
3	bayesian model	10	3	10	10	10
4	bilingual lexicon extraction	10	9	10	10	10
5	bilingual word embeddings	10	9	8	10	10

Selanjutnya hasil akhir dari setiap penilaian dihitung rata-ratanya baik untuk skenario 1 maupun skenario 2. Lalu nilai rata rata totalnya dibandingkan untuk mengetahui skenario mana yang memberikan nilai yang lebih besar yang berarti memberikan hasil pencarian yang lebih relevan. Untuk mendapatkan nilai totalnya dilakukan perhitungan sebagai berikut.

$$r(q) = \frac{\sum_1^s d}{s} \quad (6)$$

d = banyaknya dokumen yang relevan dalam satu kali pencarian

q = query yang digunakan

s = banyaknya partisipan yang melakukan penilaian

r = nilai relevansi dalam suatu pencarian

persamaan (6) digunakan mencari nilai similarity dalam satu kali pencarian menggunakan satu query atau kata kunci. Karena penelitian ini menggunakan lebih dari satu partisipan maka diambil nilai rata-ratanya.

Lalu untuk menghitung jumlah total dalam satu scenario dapat menggunakan persamaan (7) sebagai berikut.

$$\frac{r(q)}{f} = \left(\frac{\sum_1^f \frac{\sum_1^s d}{s}}{f} \right) * 100\% \quad (7)$$

d = banyaknya dokumen yang relevan dalam satu kali pencarian

q = query yang digunakan

s = banyaknya partisipan yang melakukan penilaian

f = banyaknya query yang digunakan dalam satu scenario

r = nilai relevansi dalam suatu pencarian

persamaa (7) menjelaskan bahwa nilai total relevansi dalam satu scenario didapat dengan menghitung nilai rata-rata dari partisipan pada satu pencarian menggunakan satu query. Lalu menjumlahkan semua nilai rata-rata setiap query yang kemudian dibagi banyaknya query. Sehingga didapat nilai rata-rata total dari keseluruhan query untuk satu skenario.

TABEL 8.
Nilai Total Relevansi Setiap Skenario Pencarian

Keterangan	Nilai
Nilai Total Hasil Pencarian Skenario 1	89%
Nilai Total Hasil Pencarian Skenario 2	76%

Dapat dilihat dari tabel 8, bahwa nilai total pada skenario 1 memiliki nilai yang lebih besar dari pada nilai total pada skenario 2. Hal ini menunjukan bahwa hasil pencarian pada skenario 1 memberikan hasil yang lebih relevan daripada hasil pencarian pada skenario 2. Hal ini bisa terjadi karena beberapa alasan seperti ada terjadi ambiguitas yang terjadi pada saat melakukan pencarian, dataset yang masih perlu banyak *preprocessing* secara manual, dan juga algoritme dan ketersediaan data pada mesin pencari google atau internet itu sendiri.

5. Analisis Hasil Pengujian

Hasil pengujian pada penelitian ini bisa dikatakan tidak memberikan hasil yang tidak lebih baik daripada bahan yang diujikan. Hal tersebut wajar terjadi karena tidak selamanya penelitian bisa menghasilkan hasil yang lebih baik daripada bahan ujinya. Hal itu bisa terjadi karena berapa alasan sebagai berikut.

a. Ambiguitas

Ambiguitas adalah keadaan terdapat dua kata yang memiliki arti yang serupa pada saat melakukan representasi kata khususnya untuk melakukan keperluan yang melibatkan *word recognition* [18]. Pada percobaan yang dilakukan ini terjadi ambiguitas ketika menggunakan kata kunci *stratified seed sampling* dan *stratified seed sampling supply*. Ambiguitas terjadi karena terdapat kata *seed* yang memiliki arti benih atau bibit sedangkan *seed* yang dimaksud adalah untuk keperluan *Clustering-based Stratified Seed Sampling for Semi-Supervised Relation Classification*. Sedangkan pada hasil pencarian dominan artikel berkaitan dengan biologi, perhutanan dan pertanian. Sehingga terdapat penilaian yang bersifat anomali karena hasil pencarian menggunakan kata kunci tersebut dominan tidak relevan sama sekali dengan artikel rujukan yang ditentukan.

b. Algoritme Pencarian Mesin Pencari Google Scholar

Algoritme pada mesin pencari google scholar juga memiliki kontribusi dalam proses pencarian. Karena algoritme pencarian pada mesin pencari *google scholar* menggunakan sistem ranking yang menitik beratkan banyaknya sitasi yang merujuk pada artikel tersebut dan juga tingkat relevansi antara *keyword* yang digunakan dalam pencarian dengan judul artikelnya. Sehingga pengurutannya berdasarkan relevansi judul artikel dengan *keyword* yang digunakan kemudian banyaknya artikel tersebut dijadikan rujukan [19]. Selain itu *keyword* yang digunakan perlu diperhatikan, karena pencarian menggunakan mesin pencari *google scholar* tidak akan memberikan sinonim dari *keyword* yang digunakan. Sehingga diperlukan beberapa *keyword* dalam satu kali pencarian untuk memperluas hasil pencarian [19]. Nama penulis dan jurnal yang diterbitkan juga mempengaruhi ranking hasil pencarian, dan tahun diterbitkannya artikel tersebut juga mempengaruhi hasil pencarian [19]. Terlepas dari semua hal itu, kesediaan data yang dicari pada *database* juga mempengaruhi. Pada percobaan yang dilakukan ini, pada beberapa *keyword* yang memang data yang tersedia sangat terbatas. Hal ini biasa terjadi jika memang tidak banyak artikel yang di*publish* pada keyword tertentu.

c. Dataset

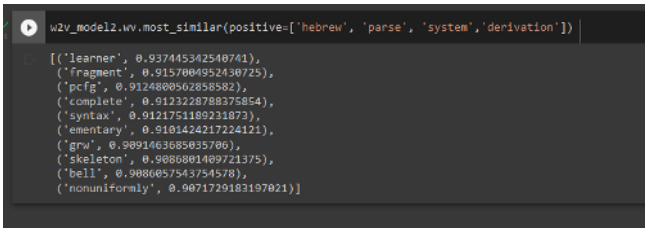
Dataset yang digunakan untuk word embedding menggunakan artikel-artikel ilmiah yang dimana sudah berformat .txt dan .csv dengan isi sebagai berikut.

Tabel 9.
Contoh Isi Artikel yang Memiliki Kata Terpotong

In this paper we present a new multi
lingual data-driven method for coreference
resolution as implemented in the SWIZZLE
system. The results obtained after training
this system on a bilingual corpus of English
and Romanian tagged texts outperformed
coreference resolution in each of the indi
vidual languages.

Pada file txt tersebut, banyak kata-kata yang terpotong garis baru dan dipisah dengan "-" yang menyebabkan satu kata yang terpotong tersebut menjadi dua kata yang berbeda. Selain itu adanya singkatan yang terhitung menjadi satu kata. Seperti pada tabel 9 terdapat kata *indi* dan pada baris dibawahnya terdapat kata *vidual* yang dimana sebenarnya kata tersebut adalah *individual*. Hal ini mempengaruhi

kepada proses *training* word2vec yang menyebabkan terhitungnya menjadi dua kata yang terpisah. Pengaruhnya dapat terlihat saat menggunakan fungsi *wv.most_similar* ketika akan melakukan ekspansi query. Berikut contoh terjadinya kasus terdapat kata yang terpotong saat menggunakan fungsi *wv.most_similar* pada gambar 5.



GAMBAR 5.
Contoh Terdapat Kata yang Terpotong pada Program

Dengan masuknya singkatan dan kata-kata yang terpotong kedalam proses data processing membuat terjadi kesalahan dalam menunjukan hasil word embedding. Seperti pada gambar 5 menunjukan dimana muncul kata "pcfg" dan "grw" dimana keduanya merupakan singkatan. Dan muncul kata "ementary" dimana merupakan kata yang terpotong dari kata "elementary".

V. KESIMPULAN

A. Kesimpulan

Pada penelitian ini dapat disimpulkan bahwa penggunaan word2vec sebagai ekstensi untuk membantu menentukan kata kunci ketika akan melakukan pencarian artikel ilmah di internet masih belum maksimal. Dibuktikan dengan banyaknya hasil pencarian menggunakan kata kunci yang sudah diekspansi menggunakan word2vec memiliki tingkat relevansi yang lebih rendah daripada hasil pencarian menggunakan kata kunci sebelum diekspansi menggunakan word2vec. Hal itu bisa disebabkan oleh beberapa hal seperti pengaruh dari algoritma pencarian google scholar, masih kurang optimal penggunaan algoritma word2vec pada kumpulan kata kunci yang akan digunakan, kurang beragamnya kata kunci yang digunakan. Saran untuk penelitian berikutnya adalah dengan merapihkan terlebih dahulu dataset yang akan digunakan agar data yang akan dilatih menjadi lebih bersih, lalu buat kata kunci yang lebih beragam agar dapat terlihat hasil yang lebih beragam juga.

REFERENSI

[1] SiswadiI. 2016. Mengenal Konsep Penetapan Kata Kunci. Jurnal Pustakawan Indonesia. 12, 2 (Mar. 2016). DOI:https://doi.org/10.29244/jpi.12.2

[2] S. Yang, X. Zheng, X. Yin, H. Mao and D. Zhao, "An Algorithm of Query Expansion for Chinese EMR Retrieval by Improving Expansion Term Weights and Retrieval Scores," in IEEE Access, vol. 8, pp. 200063-200072, 2020, doi: 10.1109/ACCESS.2020.3033017.

[3] Dirga Nugraha, N. P., Maskur, M., & Hayatin, N. (2019). Ekspansi Query Berbasis Semantik Pada Online Public Access Catalog (OPAC). Jurnal Repositor, 1(2), 87-94. https://doi.org/10.22219/repositor.v1i2.24

[4] Azad, H. K., & Deepak, A. (2019). Query expansion techniques for information retrieval: A survey. Information Processing & Management, 56(5), 1698-1735. doi:10.1016/j.ipm.2019.05.009

[5] Silva, Alfredo & Mendoza, Marcelo. (2020). Improving query expansion strategies with word embeddings. 1-4. 10.1145/3395027.3419601.

[6] Claudio Carpineto and Giovanni Romano. 2012. A Survey of Automatic Query Expansion in Information Retrieval. ACM Comput. Surv. 44, 1, Article 1 (January 2012), 50 pages. https://doi.org/10.1145/2071389.2071390

[7] Kuzi, Saar & Shtok, Anna & Kurland, Oren. (2016). Query Expansion Using Word Embeddings. 1929-1932. 10.1145/2983323.2983876.

[8] Chandrasekaran, Yasunaga, M. K., Radev, M. a., Freitag, D. a., Kan, D. a., & Min-Yen. (2019). Overview and Results: CL-SciSumm Shared Task 2019. BIRNDL. Retrieved from Github: https://github.com/WING-NUS/scisumm-corpus

[9] Aklouche, Billel & Bounhas, Ibrahim & Slimani, Yahya. (2018). Query Expansion Based on NLP and Word Embeddings.

[10] Diaz, F.D., Mitra, B., & Craswell, N. (2016). Query Expansion with Locally-Trained Word Embeddings. ArXiv, abs/1605.07891.

[11] Mosbah, Mawloud. (2018). Improving the Results of Google Scholar Engine through Automatic Query Expansion Mechanism and Pseudo Re-ranking using MVRA. Journal of information and organizational sciences. 42. 219-229. 10.31341/jios.42.2.5.

[12] jain, d. (2021, June 29). geeksforgeeks. Retrieved from geeksforgeeks web site: https://www.geeksforgeeks.org/data-preprocessing-in-data-mining/

[13] team, t. p. (n.d.). tutorials point. Retrieved from tutorials point web site: https://www.tutorialspoint.com/python_text_processing/python_remove_stopwords.html

[14] Beri, A. (2020, May 14). Towards Data Science. Retrieved from Towards Data Science web site: https://towardsdatascience.com/stemming-vs-lemmatization-2daddabcb221

[15] Radim Řehůřek, p. (n.d.). Word2vec embeddings. Retrieved from radimrehurek web: https://radimrehurek.com/gensim/models/word2vec.html

[16] Brownlee, J. (2017, Oktober 11). Machine Learning Mastery. Retrieved from Machine Learning Mastery Website: https://machinelearningmastery.com/what-are-word-embeddings/

[17] Vatsal. (2021, juli 29). Toward Data Science. Retrieved from Toward Data Science Website: https://towardsdatascience.com/word2vec-explained-49c52b4ccb71

[18] Piramuthu, Robinson & Bhardwaj, Anurag & di, Wei & Sundaresan, Neel. (2013). Visual Search: A Large-Scale Perspective. 10.1016/B978-0-444-53859-8.00011-4.

[19] Long, B., Chang, Y. (2014). Relevance Ranking for Vertical Search Engines ([edition unavailable]). Elsevier Science. Retrieved from https://www.perlego.com/book/1810021/relevance-ranking-for-vertical-search-engines-pdf (Original work published 2014)

[20] Rodd, Jennifer & Gaskell, Gareth & Marslen-Wilson, William. (2004). Modeling the effect of semantic

ambiguity in word recognition. Cognitive Science. 28. 89-104. 10.1016/j.cogsci.2003.08.002.
[21] Beel, Joeran & Gipp, Bela. (2009). Google Scholar's Ranking Algorithm: An Introductory Overview.