Perancangan Semantic Similarity based on Word Thesaurus Menggunakan Pengukuran Omiotis Untuk Pencarian Aplikasi pada I-GRACIAS

Akip Maulana [1], DR. Moch. Arif Bijaksana, Ir., M.Tech [2], M. Syahrul Mubarok, ST., M.T [3]
[123] Prodi S1 Teknik Informatika, Fakultas Informatika ,Universitas Telkom, Bandung
Jalan Telekomunikasi No. 1, Dayeuh Kolot, Bandung 40257
[1] akip.maulana@gmail.com, [2] arifbijaksana@gmail.com, [3] msyahrulmubarok@gmail.com

Abstrak

Proses pencarian dengan cara konvensional akan membuat pengguna I-GRACIAS bingung apabila keyword yang dimasukkan memiliki ejaan kata yang berbeda dengan nama aplikasi yang ada. Semantic similarity adalah suatu pendekatan untuk menangani pencarian dengan mengandalkan nilai keterhubungan antar-term yang dibentuk dari Wordnet. Pendekatan semantic similarity yang digunakan adalah Path-based dengan Wu and Palmer (WUP) sebagai metode perhitungan semantic similarity. Omiotis merupakan metode yang ditujukan untuk mengukur derajat relevansi antar-dokumen. Terdapat dua komponen utama dari perhitungan Omiotis. Komponen tersebut adalah lexical relevance dan semantic similarity. Dengan demikian, proses pencarian yang awalnya menggunakan cara konvensional diubah dengan pendekatan Semantic Textual Similarity (STS). Oleh karena itu, pada tugas akhir ini akan digunakan pengukuran Omiotis untuk menghitung kemiripan antar-dokumen dengan menggunakan pendekatan Path-based sebagai metode semantic similairty, yang mana masih memiliki ketergantungan dengan Wordnet. Sehingga mampu membantu menangani masalah pencarian aplikasi di I-GRACIAS.

Kata Kunci: Semantic Similarity, Lexycal Relevance, Omiotis, PairingWord, Wordnet.

1. Pendahulan

Proses pencarian aplikasi yang ada di I-GRACIAS dilakukan dengan cara pencocokkan keyword dengan nama aplikasi melalui query. Supaya pencarian kita berhasil, maka keyword yang dimasukkan harus sama atau setidaknya berisi suku kata yang sama dengan nama aplikasi. Apabila ada user baru dan user tesebut tidak tahu nama aplikasi yang ingin dicari, dengan kata lain keyword yang dimasukkan berbeda dengan nama aplikasi maka pencarian tidak akan berhasil [1]. Pencarian tersebut juga bisa dilakukan dengan pendekatan Semantic Similarity yang dihitung menggunakan WS4J dengan metode Path-based yaitu Wu and Palmer (WUP). Pengukuran Omiotis adalah perluasan dari pengukuran semantic relatedness. Keutamaan pengukuran Omiotis dibandingkan dengan STASIS dan LSA-based approach adalah kemampuannya untuk menghitung kemiripan antar-text yang pendek seperti kalimat [1]. Selain itu, kelebihan lain dari Omiotis adalah tidak tergantung pada Wordnet. Maka dari itu pada pengukuran ini, sistem akan melakukan perhitungan TF-IDF dengan harmonical mean untuk mendapatkan nilai lexical relevance. Nilai lexical relevance ini digunakan untuk mendapatkan nilai Omiotis.

Dalam penelitian ini akan menggunakan pendekatan semantic similarity dan perhitungan Omiotis untuk mengukur kemiripan keyword dengan beberapa atribut aplikasi (nama aplikasi dan deskripsi aplikasi) yang akan digunakan untuk proses pencarian. Perhitungan Omiotis akan melibatkan perhitungan semantic similarity. Dengan demikian, penggunaan perhitungan Omiotis bisa membantu pencarian aplikasi yang ada di I-GRACIAS.

2. Studi Literatur

Pencarian aplikasi dengan menggunakan pendekatan Semantic Textual Similarity (STS) membutuhkan metode khusus. Omiotis adalah salah satu metode STS yang bisa digunakan untuk membandingkan antar-dokumen. Sebelum menggunakan Omiotis, ada beberapa studi literatur yang harus dipahami yaitu TF_IDF dan semantic similarity. Pada bagian ini akan dijelaskan secara singkat Studi Literatur yang digunakan untuk penelitian Omiotis.

TF_IDF Weighting

Pembobotan TF-IDF digunakan untuk memberikan bobot term terhadap suatu dokumen. Dengan kata lain, TF-IDF juga bisa digunakan untuk mencari keterhubungan suatu term yang ada pada teks. TF-IDF direpresentasikan

dengan pendekatan vector, hal ini dikarenakan setiap dokumen d_i berisi kumpulan kata (t_1, t_2, ..., t_n). Relevansi setiap t_i terhadap di berbeda-beda dengan menggunakan perhitungan TF-IDF [2]. Perhitungan TF-IDF menggabungkan perhitungan Term Frequency (tf) dan Invers Document Frequency (idf) sebagaimana dinotasikan pada Persamaan 1 [3].

$$\mathrm{tf} - \mathrm{idf}_{t,d} = \mathrm{tf}_{t,d} \times \mathrm{idf}_t \quad (1)$$

Term Frequency

Term Frequency adalah pembobotan term terhadap suatu dokumen yang dihitung berdasarkan jumlah kemunculan di dokumen [3]. Notasi term frequency adalah $\mathrm{tf}_{t,d}$ yang memiliki makna bobot term t terhadap dokumen d. Sebelum melakukan proses perhitungan term frequency, hal yang harus dilakukan adalah pembuatan Bag of Words (BOW). Bag of Words berisi kumpulan dari kata-kata yang muncul di dokumen. Sifat Bag of Word adalah unik dan tidak memperhatikan urutan kemunculan kata. Persamaan 2 adalah formula pembobotan term frequency

$$\mathrm{tf}_{t,d} = 1 + \log \mathrm{tf}_{t,d} \quad (2)$$

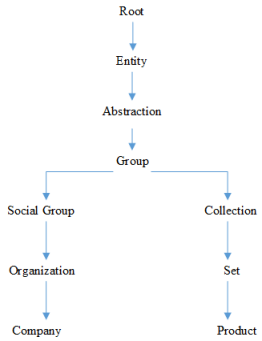
Inverse Document Frequency

Inverse Document Frequency ditujukan untuk melemahkan term yang sering muncul di setiap dokumen. Hal ini dikarenakan, semakin banyak term yang muncul di setiap dokumen maka akan semakin kecil relevansinya [3]. Inverse Document Frequency bergantung pada nilai Document Frequency. Document frequency dinotasikan sebagai df_t yakni jumlah dokumen yang berisi term t. Jika ada term t yang muncul pada dokumen, maka dianggap 1 begitu juga sebaliknya jika tidak berisi term t maka dianggap 0. Pemeriksaan dokumen berisi term t dilakukan di setiap dokumen, setelah itu dijumlahkan. Setelah menghitung df_t maka selanjutnya melakukan perhitungan inverse document frequency seperti Persamaan 3, dimana N adalah keseluruhan jumlah dokumen.

$$\mathrm{idf}_t = \log \frac{N}{\mathrm{df}_t} \quad (3)$$

Wu and Palmer

Wu and Palmer didefinisikan sebagai similarity dari dua concept berdasarkan kedalaman lcs dan jalur terpendek [4]. Proses perhitungan yang dilakukan oleh WUP adalah mencari jalur terpendek dari setiap concept, kemudian setiap jalur yang terbentuk digabungkan untuk mencari lcs-nya. Pencarian LCS (Lowest Common Subsumer) dengan cara mencari sense yang sering dimuncul dari dua jalur yang dihubungkan. Sebagai contoh perhatikan Gambar 1.



Gambar 1: Graf semantik dari concept Company dan Product

Dari Gambar 1, lcs (Company, Product) adalah Group. Hal ini dikarenakan anccestor yang sering dilewati adalah Group, Abstraction dan Entity. Tetapi yang memiliki lowest anccestor dari ketiga itu adalah Group. Persamaan 4 adalah formula yang digunakan untuk menghitung WUP [5] yang sudah dimodifikasi oleh WS4J.

$$\text{sim}_{wp}(c_1, c_2) = \frac{2 \times \text{depth}(\text{lcs}(c_1, c_2))}{\min(\text{depth}(c_1)) + \min(\text{depth}(c_2))} \quad (4)$$

Pada Persamaan 4, terdapat istilah depth. Depth merupakan kedalaman yang diukur dari root sampai ke node yang ditentukan.

Omiotis

Pengukuran Omiotis adalah pengukuran dua segmen semantik pada teks yang mencakup term dan lexycal berdasarkan jaringan semantik yang didapat dari Wordnet [6]. Sebelum kita menghitung Omiotis ada beberapa persamaan yang harus kita hitung terlebih dahulu. Tahap pertama adalah lexical relevance. Lexical relevance adalah persamaan untuk menghitung lexical similarity dari setiap teks yang akan dibandingkan. Pengukuran ini digunakan karena Wordnet tidak selalu menyediakan term khusus, contohnya adalah nama algoritma seperti Algoritma Djikstra

Skema pembobotan TF-IDF digunakan untuk mendapatkan nilai relevance similarity. Pada persamaan relevance similarity yang akan diterapkan untuk mendapatkan nilai Omiotis yakni menggunakan harmonic mean yang dinilai lebih baik dibandingkan menggunakan nilai rata-rata. Hal ini dikarenakan batas atas yang dihasilkan cenderung lebih kuat [1]. Adapun persamaan lexical similarity seperti Persamaan 5.

$$\lambda_{a,b} = \frac{2 \cdot \text{TF\_IDF}(a, A) \cdot \text{TF\_IDF}(b, B)}{\text{TF\_IDF}(a, A) + \text{TF\_IDF}(b, B)} \quad (5)$$

Setelah kita mendapatkan nilai lexical relevance, maka tahap selanjutnya adalah mencari nilai perkalian maksimum dari kombinasi term yang ada di teks A dan B. Berikut ini adalah maximum product dari setiap term $b \in B$ dengan term $a \in A$ dan sebaliknya.

$$b* = \underset{b \in B}{\text{argmax}}(\lambda_{a,b} \cdot \text{Sim}(a, b)) \quad (6)$$

$$a* = \underset{a \in A}{\text{argmax}}(\lambda_{a,b} \cdot \text{Sim}(a, b)) \quad (7)$$

Kemudian tahap selanjutnya adalah menggabungkan nilai perkalian antar semantic similarity dengan lexical relevance. Persamaan 2.9 adalah nilai agregasi untuk semua term di teks A dengan nilai b_*. untuk mencari nilai b_* maka dibutuhkan semua kemungkinan kombinasi antar-term dari dokumen A dan B.

$$\zeta(A, B) = \frac{1}{|A|}\left(\sum_{a \in A} \lambda_{a,b_*} \cdot \text{Sim}(a, b_*)\right) \quad (8)$$

Jika dibalik maka Persamaan 8 seperti Persamaan 9.

$$\zeta(B, A) = \frac{1}{|B|}\left(\sum_{b \in B} \lambda_{b,a_*} \cdot \text{Sim}(b, a_*)\right) \quad (9)$$

Maka didapat nilai Omiotis antara teks A dan B dengan Persamaan 10.

$$O_{A,B} = \frac{\zeta(A, B) + \zeta(B, A)}{2} \quad (10)$$

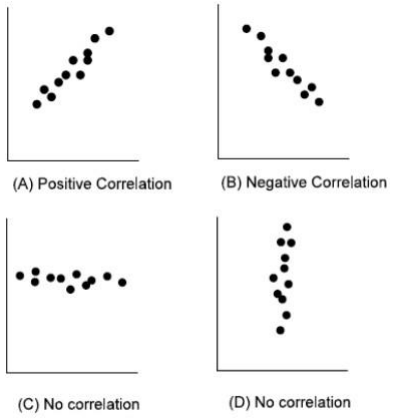
Pengukuran Korelasi

Pengukuran korelasi ditujukan untuk mengevaluasi sistem, pengukuran ini menggunakan Pearson correlation untuk mengukur keterhubungan dua variabel yang memiliki perbedaan nilai secara simultan. Hal ini berlaku jika perubahan dari salah satu variabel mempengaruhi nilai variabel lain. ada tiga tipe korelasi yang biasa digunakan [7].

(a) Positive correlation
Perubahan kedua variabelnya naik atau turun secara simultan.

(b) Negative correlation
Perubahan kedua variabelnya trade-off atau timpang.

(c) Spurious correlation
Perubahan kedua variabel tidak naik dan turun.



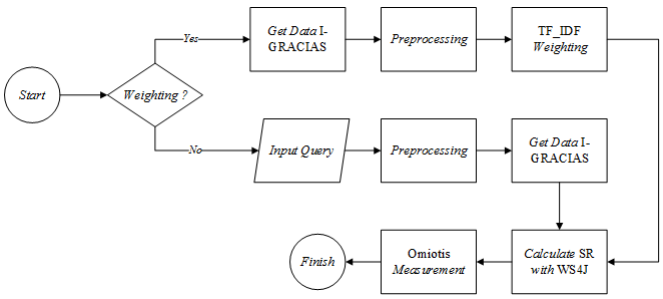
Gambar 2: Plot Diagram dari tipe korelasi [7]

Pearson correlation dinotasikan sebagai **r** adalah pengukuran korelasi atau keterhubungan dua variabel berdasarkan random variate dari masing-masing variabel. Persamaan 11 adalah persamaan untuk menghitung **r**.

$$r_{X,Y} = \frac{\sum (x - \bar{x})(y - \bar{y})}{\sqrt{\sum (x - \bar{x})^2 \sum (y - \bar{y})^2}} \quad (11)$$

3. Perancangan

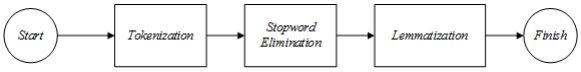
Perancangan sistem Omiotis dilakukan melalui tahap Preprocessing dan Weighting. Gambar 3 merupakan alur sistem secara keseluruhan, berdasarkan flowchart tersebut proses F_IDF Weighting dipisah karena bisa dikataka proses ini adalah fase learning sehingga akan membutuhkan waktu yang cukup lama. Setiap alur tersebut akan dijelaskan dari tahap Preprocessing hingga Omiotis.



Gambar 3: Flowchart sistem

Preprocessing

Pada tahap ini sistem akan melakukan tiga proses preprocessing yakni sebagai berikut.



Gambar 4: Flowchart preprocessing

Tokenization

Proses preprocessing ini melakukan pemecahan kalimat menjadi kata. Setelah Q melalui proses tokenization dengan menggunakan regex[1], maka Q dan A akan berubah menjadi :

$$Q = \{salaries,of,employees\}$$

$$A = \{payroll,management,application\}$$

Stopword Elimination

Proses ini akan mereduksi imbuhan, partikel dan kata lain yang tidak penting berdasarkan daftar stopword yang sudah didefinisikan[2]. Hasil dari tahap ini adalah sebagai berikut.

$$Q = \{salaries,employees\}$$

$$A = \{payroll,management,application\}$$

Lemmatization

Pada proses preprocessing yang terakhir adalah lemmatization. Proses ini menggunakan library Standford CoreNLP dengan annotation yang digunakan adalah tokenize, pos dan lemma. Sehingga hasil dari tahap ini adalah sebagai berikut.

$$Q = \{salary,employee\}$$

$$A = \{payroll,management,application\}$$

TF_IDF Weighting

Pada tahap ini dibutuhkan perhitungan TF_IDF dari kedua teks Q dan A. Sebagai asumsi untuk menghitung TF_IDF dibutuhkan jumlah dokumen, jumlah dokumen diasumsikan sebanyak 50 dokumen. Tabel 1 adalah tabel penilaian TF_IDF weighting setiap term.

Tabel 1: TF_IDF weighting

	TF Raw		TF Weighting		DF	IDF	TF IDF	
	A	Q	A	Q			A	Q
Salary	0	1	0	1	25	0,3	0	0,3
Employee	0	1	0	1	23	0,34	0	0,34
Payroll	1	0	1	0	15	0,52	0,52	0
Management	1	0	1	0	43	0,07	0,07	0
Application	1	0	1	0	45	0,05	0,05	0

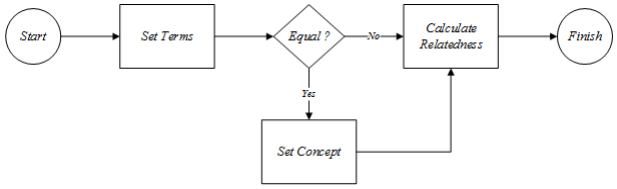
Pada Tabel 1, terdapat beberapa sub header yakni TF_Raw, TF_Weighting, IDF, dan TF_IDF. TF_Raw adalah term frequency dari term tertentu yang muncul pada setiap dokumen, Sedangkan TF_Weighting adalah pembobotan

[1] [A-Za-z][A-Za-z]*
[2] http://www.d.umn.edu/ tpederse/Group01/WordNet/wordnet-stoplist.html

dari TF_Raw berdasarkan Persamaan 2 Setelah mendapatkan nilai TF_Weighting, selanjutnya adalah perhitungan IDF yakni inverse document frequency berdasarkan Persamaan 3. Sehingga perhitungan TF_IDF didapat dari Persamaan 1.

Calculate SR with WS4J

Perhitungan semantic similarity yang digunakan adalah WUP dengan cara memanfaatkan calculator relatedness yang disediakan oleh library WS4J. Pemilihan WUP berlandaskan pada syarat Omiotis yakni Path-based. Adapun Cara menggunakan WUP pada WS4J yaitu dengan mendefinisikan jenis calculator yang digunakan dengan memberikan parameter term yang akan dihitung. Gambar 5 adalah flowchart perhitungan WUP.



Gambar 5: Flowchart Perhitungan WUP di WS4J

(a) Proses Set Terms adalah proses menentukan pasangan antar-term dengan cara membandingkan setiap term yang ada di dokumen A dengan term yang ada di Q seperti persamaan 12.

$$WUP_{t \in d} = \underset{t^0 \in d^0}{\operatorname{argmax}}(sim(t, t^v)) \quad (12)$$

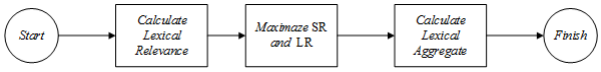
(b) Apabila sama, maka term akan diubah menjadi concept. Jika tidak maka pasangan term langsung dihitung.

(c) Set Concept adalah proses pengubahan term menjadi concept dengan jenis POS-nya adalah noun.

(d) Calculate Relatedness adalah prose perhitungan WUP. Perhitungan tersebut jika di ilusrasikan sama dengan Gambar 1 dan menggunakan Persamaan 4.

Omiotis Measurement

Proses perhitungan Omiotis memiliki tiga perhitungan yang harus dilakukan. Gambar 6 adalah alur proses perhitungan Omiotis.



Gambar 6: Flowchart Pengukuran Omiotis

Calculate Lexycal Relevance

Lexical relevance (LR) adalah proses perhitungan nilai keterkaitan antar-dokumen yang akan dibandingkan berdasarkan lexical dari term $a \in A$ dan $b \in B$. Perhitungan ini melibatkan TF_IDF yang sebelumnya sudah dihitung terlebih dahulu. Persamaan 5 adalah perhitungan lexical relevance dengan menggunakan harmonical mean, sehingga nilai rata-rata keterkaitan antar-dokumen mencapai batas tengah (tight upper bound) [1].

Maximaze SR and LR

Setelah menghitung lexical relevance, perhitungan selanjutnya adalah mencari nilai maksimum perkalian LR dan SR dari perwakilan setiap dokumen. Perhitungan ini menggunakan Persamaan 6 dan 7, sehingga setelah proses ini selesai maka akan didapatkan satu term terbaik dari setiap dokumen yang simbolkan dengan $a \in A$ dan $b \in B$.

Calculate Lexycal Aggregate

Tahap perhitungan terakhir adalah mencari nilai perkalian LR dan SR pada setiap dokumen dengan best term dari dokumen lawan menggunakan Persamaan 8 dan 9. Dua persamaan tersebut kemudian dirata-ratakan menggunakan Persamaan 10 atau yang disebut persamaan Omiotis.

4. Evaluasi

Penelitian ini melibatkan dua dataset yakni dataset Semeval 2014 task 3 dan dataset I-GRACIAS. Dataset Semeval 2014 task 3 digunakan untuk menguji pengukuran Omiotis dan sistem pembandingnya yaitu PairingWord. Dengan menggunakan dataset tersebut dapat dikatakan bahwa Omiotis adalah salah satu metode perhitungan di bidang Semantic Textual Similarity (STS) khususnya untuk kasus Cross-Level Semantic Similarity (CLSS). Sedangkan dataset I-GRACIAS digunakan untuk pembuktian bahwa sistem Omiotis bisa diterapkan pada environment I-GRACIAS.

Gold Standard

Penilaian pasangan teks yang dilakukan berdasarkan penilaian manusia disebut dengan Gold Standard. Setiap dataset memiliki gold standard yang disediakan dalam bentuk file terpisah dan memiliki range dari 0 sampai 4. Setiap range tersebut dijelaskan pada Tabel 2.

Tabel 2: Guidelines for gold standard [8]

4 – Very similar	Apabila secara keseluruhan dua teks yang dibandingkan memiliki makna yang sama atau smaller text merepresentasikan konsep dari larger text dengan kata lain smaller text adalah ringkasan dari larger text
3 – Somewhat similar	Apabila dua teks yang dibandingkan banyak memiliki konsep yang sama, tetapi secara detail sedikit berbeda. Contoh: car vs. vehicle
2 – Somewhat related but not similar	Apabila dua teks yang dibandingkan memiliki perbedaan makna, tetapi ada beberapa konsep yang berelasi. Contoh: windows vs. house.
1 – Slightly related	Apabila dua teks yang dibandingkan memiliki konsep yang berbeda dan beberapa domain yang sama.
0 – Unrelated	Apabila dua teks yang dibandingkan tidak memiliki makna yang sama dan tidak memiliki keterkaitan sama sekali

Hasil Pengujian Semeval 2014 task 3

Pengujian sistem ditujukan untuk memvalidasi seberapa dekat sistem dengan nilai gold standard. Pengujian ini dilakukan dengan menggunakan korelasi Pearson untuk setiap nilai pasangan dokumen di level dataset yang berbeda berdasarkan hasil perhitungan Omiotis dan PairingWord. Sistem PairingWord adalah sistem yang dikenalkan oleh tim Meerkat_Mafia dan meraih peringkat ke - 12. Tetapi, pada penelitian ini penulis menguji sistem tersebut dengan beberapa tahapan yang berbeda. Tahapan tersebut adalah perbedaan proses preprocessing dan perhitungan semantic. Perhitungan semantic similarity yang dikenalkan oleh tim Meerkat_Mafia adalah perhitungan semantic yang menggabungkan Latent Semantic Analysis (LSA) similatiy dan Wordnet. LSA bergantung pada hipotesis kemunculan kata pada konteks yang sama .Jadi, untuk mendapatkan statistik kemunculan kata dibutuhkan beberapa corpus seperti Wikipedia, Project Gutenberg e-Books, ukWac, Reuters News stories dan LDC gigawords [9]. Ketergantungan tersebut melebihi batasan masalah dari penelitian ini yang hanya mengandalkan Wordnet.

Dari hasil pengujian yang dilakukan, penulis dapat menganalisis pengaruh semantic similarity dan lexical relevance terhadap pengukuran Omiotis. Dengan memodifikasi parameter perhitungan semantic similarity akan membuktikan semakin baik nilai semantik maka nilai STS akan semakin baik. Analisis lain yang dapat dilakukan adalah membandingkan hasil Omiotis dan PairingWord dengan menggunakan dataset Semeval 2014 task 3.

Tabel 3: Task Result [8]

No	Tim - Sistem	Para-2-Sent	Sent-2-Phr	Phr-2-Word	Word-2-Sense
1	SimCompas – run1	0,811	0,742	0,415	0,356
2	ECNU – run1	0,834	0,771	0,315	0,269
3	UNAL-NLP – run2	0,837	0,738	0,274	0,256
4	SemantiKLUE – run1	0,817	0,754	0,215	0,314
5	UNAL-NLP – run1	0,817	0,739	0,252	0,249
6	UNIBA – run2	0,784	0,734	0,255	0,180
7	UNIBA – run1	0,769	0,729	0,229	0,165
8	UNIBA – run3	0,769	0,729	0,229	0,165
9	BUAP – run1	0,805	0,714	0,162	0,201
10	BUAP – ruan2	0,805	0,714	0,142	0,194
11	Meerkat Mafia - PairingWord	0,794	0,704	-0,044	0,389
12	HULTECH – run1	0,693	0,665	0,254	0,150
	GST Baseline	0,728	0,662	0,146	0,185
13	HULTECH – run3	0,669	0,671	0,232	0,137
14	RTM-DCU – run3	0,780	0,677	0,208	
15	HULTECH – run2	0,667	0,633	0,180	0,169
16	RTM-DCU – run1	0,786	0,666	0,171	
17	Meerkat Mafia – Super Saiyan	0,834	0,777		
18	Meerkat Mafia – Hulk2	0,826	0,705		
19	RTM-DCU – run2	0,747	0,588	0,164	
20	FBK-TR – run3	0,759	0,702		
21	FBK-TR – run1	0,751	0,685		
22	FBK-TR – run2	0,770	0,648		
23	Duluth – Duluth2	0,501	0,450	0,241	0,2119
24	AI-KU – run1	0,732	0,680		
	LCS Baseline	0,527	0,562	0,165	0,109
25	UNAL-NLP - run3	0,708	0,620		
26	AI-KU - run2	0,698	0,617		
27	TCDSCSS - run2	0,607	0,552		
28	JU-Evora - run1	0.536	0.442	0.090	0.091
29	TCDSCSS - run1	0.575	0.541		
30	Penelitian - Omiotis	0,436	0,398	0,111	0,113
31	Duluth - Duluth1	0,458	0,440	0,075	0,076
32	Duluth - Duluth3	0.455	0.426	0.075	0.079
33	Penelitian - PairingWord	0,377	0,375	0,093	0,116
34	OPI - run1		0,433	0,213	0,152
35	SSMT - run1	0,789			
36	DIT - run1	0,785			
37	DIT - run2	0,784			
38	UMCC DLSI SelSim - run1	0.760			
39	UMCC DLSI SelSim - run2	0.698			
40	UMCC DLSI Prob - run1	0.023			

Tabel 3 berisi hasil submission dari setiap sistem oleh setiap participant. Score ranking berdasarkan Pearson correlation. Pearson correlation dipilih sebagai official rank dengan cara mencari nilai rata-rata dari hasil Pearson di setiap dataset. Nilai rata-rata penelitian Omiotis adalah 0,264 dan menempati posisi 30 dari 38 sistem. Sedangkan, Nilai rata-rata PairingWord adalah 0,239 dan menempati posisi 33 dari 38 sistem. Dua penelitian tersebut masih dibawah baseline, hal ini penulis memiliki hipotesis bahwa perbedaan preprocessing dan semantic similarity akan mempengaruhi hasil STS. Maka dari itu, Dengan hasil pengujian ini, penulis dapat membuktikan bahwa pengukuran Omiotis mampu melebihi salah satu metode dari participant Semeval 2014 task 3.

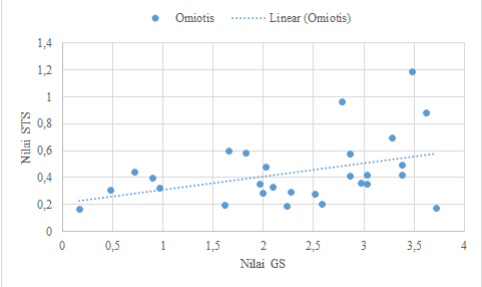
Hasil Pengujian IGRACIAS

Dataset yang bisa dievaluasi adalah dataset yang memiliki gold standard. Sebenarnnya dataset IGRACIAS belum memiliki gold standard karena bentuk datanya masih berdasarkan deskripsi dan nama aplikasi. Maka dari itu, penulis melakukan survei untuk penilaian dataset I-GRACIAS. Aturan penilaian disesuaikan dengan aturan penilaian pada Semeval 2014 task 3 seperti pada tabel 2. Tabel 4 adalah statistik penilaian koreponden.

Tabel 4: Statistik penilaian dataset I-GRACIAS

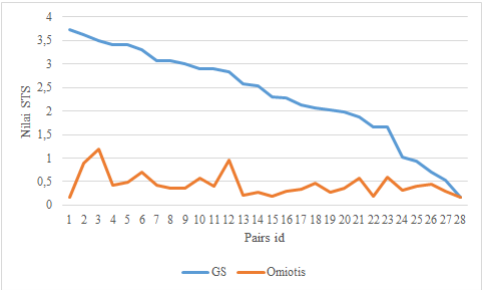
Pairs id	Gold Standard	Pairs id	Gold Standard
1	2.53	15	3.40
2	3.73	16	3.07
3	1.67	17	1.03
4	1.67	18	0.17
5	2.13	19	3.50
6	2.27	20	3.30
7	2.83	21	0.53
8	1.97	22	2.57
9	0.93	23	3.00
10	3.63	24	2.90
11	1.87	25	2.30
12	2.90	26	0.70
13	3.40	27	2.07
14	2.03	28	3.07

Grafik 7 merupakan plot chart dari hasil Omiotis menggunakan dataset I-GRACIAS. Nilai korelasi Pearson pada dataset ini adalah 0,44 yang direpresentasikan dengan garis linear menaik dan nilai tersebut tergolong kategori medium [3]. Nilai tertinggi dan terendah pada dataset ini adalah 1,19 dan 0,16 terjadi pada pairs id Igrace-19 dan Igrace-18.



Gambar 7: Grafik nilai Pearson dataset I-GRACIAS

Sedangkan, Grafik 8 adalah hasil perbandingan Omiotis dengan gold standard. Nilai gold standard yang digunakan berdasarkan Tabel 4. Berdasarkan Grafik 8 tinggi atau rendahnya nilai korelasi Pearson tidak dipengaruhi oleh seberapa dekat hasil Omiotis dengan gold standard per pairs id. Nilai korelasi Pearson akan naik jika hasil Omiotis sesuai dengan naik turunnya gold standard.

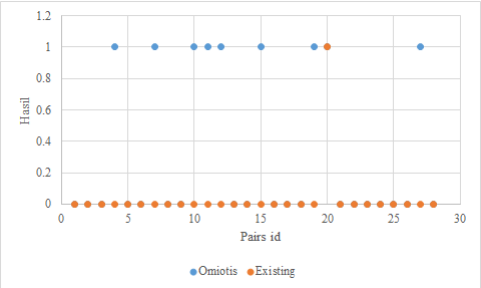


Gambar 8: Grafik selisih gold standard dan hasil Omiotis I-GRACIAS

Pada Grafik 9 menjelaskan bahwa hasil pencarian Omiotis masih bisa dilakukan pada beberapa pairs-id, meskipun secara textual berbeda. Terdapat 9 dari 28 pairs id yang match jika menggunakan Omiotis dan hanya 1 dari 28

[3]https://statistics.laerd.com/statistical-guides/pearson-correlation-coefficient-statistical-guide.php

pairs id jika menggunakan metode konvensional. Berdasarkan hasil tersebut penulis bisa mengatakan performansi dengan metode Omiotis dan konvensional pada data I-GRACIAS adalah 0,32 dan 0,04. Sehingga bisa dibuktikan bahwa Omiotis bisa membantu proses pencarian pada data I-GRACIAS.



Gambar 9: Grafik perbandingan hasil pencarian I-GRACIAS

Pada pengujian ini dilakukan dengan cara membandingkan setiap pairs id menggunakan Omiotis dan metode konvensional dalam hal ini yang masih menggunakan query basis data atau string matching. Skenario pengujian ini adalah membandingkan antar-dokumen setiap pairs-id yang telah dilampirkan pada lampiran bagian dataset I-GRACIAS. Dengan cara metode konvensional proses pencarian berhasil jika terdapat string atau substring yang match, sedangkan jika menggunakan Omiotis pencarian akan berhasil jika nilai Omiotis lebih dari nilai rata - rata dari hasil Omiotis yakni sebesar 0,44.

5. Kesimpulan

Berdasarkan hasil pengujian dan analisis yang telah dilakukan, maka dapat disimpulkan bahwa

(a) Pengukuran Omiotis dipengaruhi oleh metode semantic similarity yakni WUP, Semakin baik metode semantic similarity maka hasil Omiotis akan semakin baik.

(b) Pengaruh lexical relevance memegang peranan penting pada Omiotis, dengan adanya lexical relevance akan menaikkan derajat relevansi antar-dokumen.

(c) Pengukuran perbandingan antar-dokumen dengan menggunakan Omiotis memiliki hasil korelasi Pearson lebih baik, dibandingkan PairingWord. Hal tersebut dikarenakan pada perhitungan Omiotis terdapat komponen lexical relevance, sedangkan PairingWord murni hanya mengandalkan perhitungan semantic similarity.

(d) Pengujian Omiotis dengan menggunakan data I-GRACIAS memiliki nilai korelasi Pearson 0,44. Dengan nilai yang tidak dibawah 0, maka Omiotis masih bisa diterapkan pada I-GRACIAS.

6. Saran

Berikut ini adalah beberapa saran untuk pengembangan penelitian ini lebih lanjut.

(a) Semantic similarity menggunakan pendekatan Information Content.

(b) Sebaiknya menggunakan semantic relatedness karena bisa mencakup cross POS.

(c) Pada metode Omiotis sebaiknya memperhatikan POS-tagging dari setiap document.

Daftar Pustaka

[1] George Tsatsaronis, Iraklis Varlamis, and Michalis Vazirgiannis. Text relatedness based on a word thesaurus. Journal of Artificial Intelligence Research, 37(1):1–40, 2010.

[2] Zhang Yun-tao, Gong Ling, and Wang Yong-cheng. An improved tf-idf approach for text classification. Journal of Zhejiang University Science A, 6(1):49–55, 2005.

[3] Christopher D Manning, Prabhakar Raghavan, Hinrich Schütze, et al. Introduction to information retrieval, volume 1. Cambridge university press Cambridge, 2008.

[4] Ziqi Zhang, Anna Lisa Gentile, and Fabio Ciravegna. Recent advances in methods of lexical semantic relatedness–a survey. Natural Language Engineering, 19(04):411–479, 2013.

[5] Zhibiao Wu and Martha Palmer. Verbs semantics and lexical selection. In Proceedings of the 32nd annual meeting on Association for Computational Linguistics, pages 133–138. Association for Computational Linguistics, 1994.

[6] George Tsatsaronis, Iraklis Varlamis, Michalis Vazirgiannis, and Kjetil Nørvåg. Omiotis: A thesaurus-based measure of text relatedness. In Machine Learning and Knowledge Discovery in Databases, pages 742–745. Springer, 2009.

[7] N. Okendro Singh. Correlation and regression. Indian Agricultural Statistics Research Institute, pages 28–42, 2005.

[8] David Jurgens, Mohammad Taher Pilehvar, and Roberto Navigli. Semeval-2014 task 3: Cross-level semantic similarity. In Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), in conjunction with COLING, pages 17–26, 2014.

[9] Lushan Han, Abhay Kashyap, Tim Finin, James Mayfield, and Jonathan Weese. Umbc ebiquity-core: Semantic textual similarity systems. In Proceedings of the Second Joint Conference on Lexical and Computational Semantics, volume 1, pages 44–52, 2013.