Pengembangan Aplikasi Antar-muka Pemakai untuk Penghitungan Similaritas Semantik Berbasis String dan Wordnet

Lintang Yuniar Banowosari[1], I Wayan Simri Wicaksana[2]

lintang@gunadarma.ac.id , iwayan@staff.gunadarma.ac.id

1,2 Universitas Gunadarma

Jl. Margonda Raya No. 100 Pondok Cina Depok 16424

Abstrak

Dewasa ini, internet mempunyai kontribusi yang sangat besar dalam pertukaran data, dengan kata lain, internet memperkenalkan beberapa isu baru, salah satunya adalah keberagaman atau heterogenitas. Saat ini, sumber informasi semakin banyak, besar, terdistribusi, dinamis dan terbuka. Setiap sumber informasi dapat mempunyai metode yang berbeda dalam menyimpan data. Similaritas semantik mempunyai peran yang sangat penting dalam penyampaian dan pengintegrasian informasi. Pendekatan ke dalam model similaritas semantik dengan menghitung jarak semantik antar definisi di dalam suatu suatu ontologi, antara lain pendekatan berbasis Wordnet dan berbasis string. Untuk membantu menghitung similaritas semantik, dikembangkanlah aplikasi yang mempunyai fitur antar muka pemakai yang berbasis grafis dengan menggunakan bahasa pemrograman Python 2.4.3 dan WxPython. Dengan adanya aplikasi ini pengguna dapat lebih mudah dalam menghitung similaritas semantik.

Keywords: similaritas, semantik, string, Wordnet, aplikasi, Python

1. Pendahuluan

Kemajuan teknologi terutama dalam bidang teknologi informasi pada satu dasawarsa terakhir ini telah berubah dengan sangat cepat seiring dengan kemajuan zaman. Internet dan Web merupakan sumber informasi yang semakin lama semakin besar, hal ini memunculkan masalah dalam beberapa isu tentang sumber informasi yaitu : *massive* (sangat besar), terdistribusi, dinamis, dan open.

Besarnya jumlah sumber informasi tersebut menimbulkan keragaman dari sumber informasi. Keragaman timbul karena adanya perbedaan domain keilmuan, negara, bahasa dan sebagainya. Sumber informasi yang telah tersedia dewasa ini (yang terdapat dalam suatu web) memiliki metode penyajian yang berbeda-beda walaupun tujuan dari pembuatan atau penyajian web tersebut sama, hal ini dikarenakan teknologi web memiliki tingkat autonomi yang tinggi. Hal ini juga yang membutuhkan suatu metode penafsiran yang berbeda - beda untuk masing-masing web tersebut walupun pada akhirnya akan memiliki hasil yang sama.

Keragaman yang terjadi juga ada ditingkat semantik. Contoh sederhana adalah adanya perbedaan pemahaman akan sebuah konsep. Konsep bisa dikatakan seperti pemahaman akan sebuah istilah. Perbedaan istilah yang berarti setiap istilah akan memiliki perbedaan arti tergantung penggunaan

istilah tersebut dan siapa yang menggunakan istilah tersebut. Perbedaan tersebut dapat berupa sebuah istilah yang memiliki arti berbeda, atau istilah yang berbeda tetapi memiliki arti yang sama.

Karena masalah keragaman tersebut, maka untuk pertukaran informasi akan menjadi suatu kendala yang sulit untuk dipertemukan tanpa ada solusi yang efektif. Salah satu pendekatan yang memungkinkan untuk menjembatani masalah ini adalah menggunakan Web Semantik yang memanfaatkan teknologi *ontology* dengan melakukan perhitungan similaritas semantik.

Similaritas semantik mempunyai peran yang sangat penting dalam pemberian informasi dan pengintegrasian informasi. Pendekatan-pendekatan kedalam model similaritas semantik dengan menghitung jarak semantik antar definisi di dalam suatu *ontology*. Salah satunya dengan pendekatan string dan wordnet.

Sebelum ini sudah dikembangkan aplikasi untuk penghitungan similaritas semantik, tetapi pada aplikasi tersebut masih terdapat kekurangan, yaitu sisi tampilan antarmuka dari aplikasi tersebut, dan juga mempermudah koneksi ke internet.

2. Pencocokan Label

Perhitungan similaritas semantik adalah merupakan proses yang memerlukan keterlibatan beberapa disiplin ilmu, seperti bahasa, komputer, matematika logik dan domain yang bersangkutan. Langkah awal perhitungan kesamaan semantik adalah mengacu kepada kesamaan terminological atau kerap kali disebut label, dan disebut sebagai pencocokan label. Terminologi yang dimaksud dapat meliputi *class, property* hingga *instances*.

Pencocokan (matching) adalah pekerjaan yang sangat penting dalam interoperabilitas informasi. Secara umum proses pencocokan membutuhkan dua skema sebagai input, yang berisi kelas, properti, aturan dan lain lain. Hasil dari pencocokan menentukan output dari relasi antara elemen.

Motivasi dari masalah pencocokan adalah banyak sumber informasi mempunyai model yang berbeda untuk merepresentasikan konten informasi. Sebagai contoh, satu sumber mempunyai kelas yang disebut sebagai Server Komputer, sumber lainnya mempunyai kelas yang disebut sebagai Komputer. Ketika sebuah permintaan dikirim ke dua sumber daya tadi, satu pemetaan (mapping) diperlukan untuk merefleksikan relasi diantara Server Komputer dan Komputer.

2.1 Metode Berbasis String

Metode berbasiskan string menggunakan struktur dari string itu sendiri (sebagai satu urutan dari huruf). Metode berbasis string biasanya misalnya akan menemukan dan menganggap Match dan match adalah kelas yang serupa, tapi tidak untuk alignment.

Nama atau label yang serupa (similar) dari kelas dan properti antara skema dapat dikalkulasi berdasarkan pada metode string.

Terdapat banyak cara untuk membandingkan string, beberapa diantaranya adalah Levenshtein dan Euclidian N-Gram Distance (Edit Distance).

2.1.1 Levenshtein

Levenshtein atau *edit* distance [4] adalah suatu pendekatan yang berbasis string yang digunakan untuk menghitung jarak/perbedaan string. *Edit distance* mendefinisikan string dengan sembarang panjangnya, dan dapat menghitung perbedaan antara 2 string, di mana perhitungan perbedaan tidak hanya ketika string mempunyai perbedaan karakter tetapi salah satunya mempunyai karakter tertentu, sedangkan string yang lain tidak. Definisi formal adalah sebagai berikut:

r(a, b) = 0 if a = b. Let r(a, b) = 1,otherwise. (3)

Misalkan diberikan 2 string s dan t dengan panjang d dan m. Kita akan mengisi array d (n+1)x(m+1) dengan bilangan bulat positif (integer) sehingga elemen pojok kanan yang paling rendah d(n+1,m+1) akan memindahkan/menyediakan nilai yang dibutuhkan Levenshtein distance L(s,t). Definisi masukan dari d adalah rekursif. Himpunan pertama d(i,0) =i, i = 0,1,...,n dan d(0,j) = j, j = 0,1,...,m.

2.1.2 Euclidian N-Gram Distance

Jarak Euclidian [4] adalah jarak yang "biasa" antara dua (2) titik yang salah satunya dapat diukur dengan penggaris, yang dapat dibuktikan melalui aplikasi dari teorema Pythagoras. Dengan menggunakan formula tersebut sebagai jarak, ruang Euclidian akan menjadi ruang metrik atau metrik pythagoras.

Jarak Euclidian antara dua (2) titik $P = (p_1, p_2, \ldots, p_n)$ dan $Q = (q_1, q_2, \ldots, q_n)$, dalam n-ruang Euclidian didefinisikan sebagai berikut:

$$\sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \quad (4)$$

Distance dua-dimensi sebagai berikut: untuk dua (2) titik 2D: $P = (p_x, p_y)$ dan $Q = (q_x, q_y)$, jarak dihitung sebagai :

$$\sqrt{(p_x - q_x)^2 + (p_y - q_y)^2} \quad (5)$$

2.2 Wordnet

Wordnet merupakan sebuah leksikal database elektronik. Wordnet dikembangkan untuk referensi bahasa Inggris oleh Universitas Princeton di Amerika. Wordnet adalah system referensi leksikal online yang rancangannya terinspirasi oleh teori psikolinguistik dari memori leksikal manusia. Kata benda, kata kerja, kata sifat dan kata keterangan dalam bahasa Inggris diorganisir menjadi himpunan sinonim, dimana masing-masing merepresentasikan satu konsep leksikal.

Perhitungan similaritas semantik berbasiskan wordnet adalah suatu metode untuk menghitung jarak antar kata berdasarkan pada hierarki dari kata-kata tersebut dalam kamus wordnet.

Metode kesamaan semantik perhitungan pada WordNet dibagi dalam dua kelompok besar pendekatan [9], yaitu *path lenght* dan *information content*. *Path lenght* secara sederhana menghitung jumlah node atau relasi yang menghubungkan antar node dalam taksonomi. Jarak yang lebih pendek antara dua konsep,bararti memiliki kesamaan lebih tinggi.

Pathlenght memberikan keuntungan dengan tidak bergantung pada statistik *corpus* dan tidak terpengaruh dengan penyebaran kata. Tetapi memiliki kelemahan dalam taksonomi yang memiliki jarak yang *uniform*/sama. Beberapa contoh pendekatan dengan *path lenght* adalah Leacock-Chodorow, Resnik, Wu-Palmer.

Pada paper ini yang akan diimplementasikan adalah pendekatan Wu-Palmer (wup) dan Leacock-Codorow (LCh).

2.2.1 Metode Wu-Palmer (wup)

Metode perhitungan ini mempertimbangkan posisi dari konsep C1 dan C2 dalam taksonomi relative dengan posisi pada konsep umum C. Walaupun kemungkinan ada *parent* ganda untuk setiap konsep, dua konsep dapat berbagi *parent* dengan jalur ganda. Rumus untuk mengkalkulasi wup adalah sebagai berikut:

$$sim_{W\&P}(c_1, c_2) = \frac{2H}{N_1 + N_2 + 2H} \quad (5)$$

Dimana $N1$ dan $N2$ adalah nilai dari link IS-A dari C1 dan C2 terhadap konsep umum C dan H adalah nilai dari link IS-A dari C terhadap *root* dari taksonomi. Ini bernilai antara 1 (jika konsep sama) dan 0.

2.2.2 Metode Leacock-Codorow (LCh)

Metode LCh [6], didasarkan pada panjang alur yang paling pendek antara konsep kata benda dalam suatu hirarki *is-a*. Alur yang paling pendek adalah yang meliputi jumlah konsep *intermediate*/antara yang paling sedikit. Nilai ini diskala oleh kedalaman hirarki D, di mana kedalaman digambarkan sebagai panjang alur yang terpanjang dari suatu node daun/*leaf* ke node akar hirarki.

Pengukuran dengan LCh mengasumsikan adanya sebuah top node yang mewakili semua node, dan akan selalu memberikan nilai lebih besar dari nol, sepanjang dua konsep yang akan dibandingkan terdapat di WordNet.

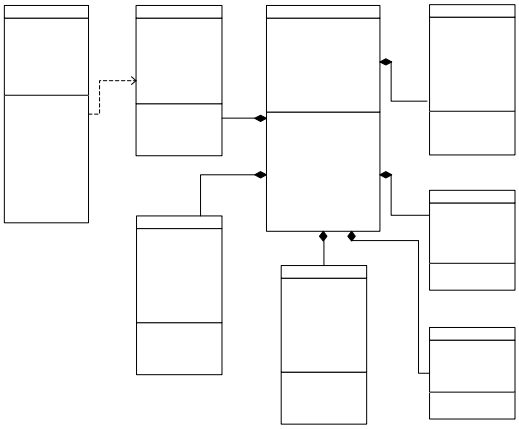
Rumus untuk kalkulasi LCh adalah sebagai berikut:

$$lch = \log(\frac{(2 * D)}{(length(c1,c2))}) \qquad (6)$$

Dimana *length* adalah jarak dari jalur terpendek antara dua *synsets* (menggunakan *node-counting*) dan D adalah kedalaman maksimum dalam taksonomi.

3. Desain Program Aplikasi

Gambar 3 adalah diagram kelas dari aplikasi perhitungan persamaan semantik. Diagram tersebut menggambarkan bahwa dalam aplikasi perhitungan persamaan semantik ini penulis membuat 7 buah kelas utama untuk tampilan dan satu buah kelas tambahan untuk perhitungan persamaan semantik dengan metode ngram. Dalam diagram tersebut dapat dilihat bahwa MenuUtama.py merupakan kelas utama dimana seluruh kelas yang lain berhubungan dengan kelas MenuUtama.py ini.

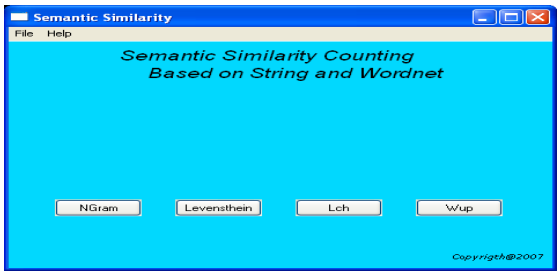


Gambar1 Diagram Kelas

Aplikasi pencocokan label dikembangkan dengan menggunakan Python 2.4.3 sebagai bahasa pemrograman utama dan wxPython 2.6 untuk membuat antar muka yang bersifat grafis.

4. Hasil dan Diskusi

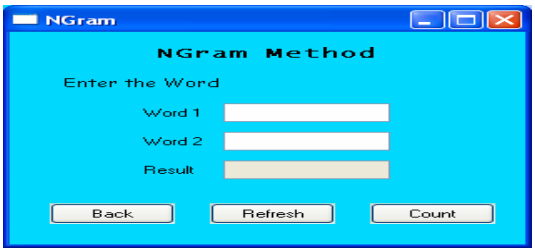
Program yang harus dijalankan adalah pyApp.py, dan akan memanggil MenuUtama.py. Tampilan awal dari program adalah sebagai berikut:



Gambar 2 Menu Utama

Pada tampilan tersebut pada gambar 2, kita dapat memilih metode perhitungan yang akan digunakan yaitu dengan menekan tombol levensthein atau ngram. Atau dengan cara lain, yaitu dengan memilih pada menu File.

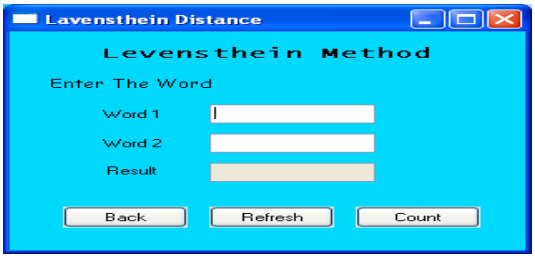
Jika kita memilih menu Ngram, maka aplikasi ini akan memanggil Ngram.py.



Gambar 3 Menu Ngram

Gambar 3 di atas merupakan tampilan dari program Ngram.py yaitu untuk melakukan perhitungan jarak semantik antara dua kata dengan metode ngram. Untuk melakukan perhitungan, pertama kita masukkan dua buah kata yang akan dihitung. Kata tersebut diketikkan pada kolom Word 1 dan Word 2. Setelah itu, tekan tombol hitung dimana tombol tersebut akan memanggil fungsi untuk menghitung menggunakan metode ngram.

Pada menu utama, jika kita memilih menu lavenstein, maka aplikasi ini akan memanggil program Levenshtein.py. Untuk tampilan menu levensthein sama dengan tampilan menu ngram. Yang membedakan hanyalah fungsi perhitungannya.

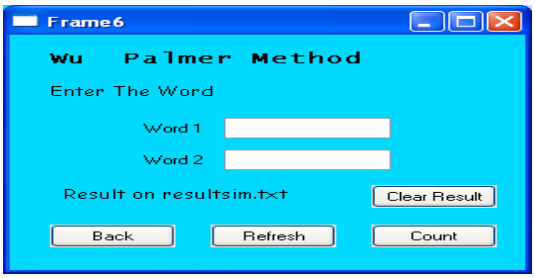


Gambar 4 Menu Levenshtein

Untuk melakukan perhitungan, pertama kita masukkan dua buah kata yang akan dihitung. Kata tersebut diketikkan pada kolom Word 1 dan Word 2. Setelah itu, tekan tombol Count di mana tombol tersebut akan memanggil fungsi untuk menghitung menggunakan metode levensthein.

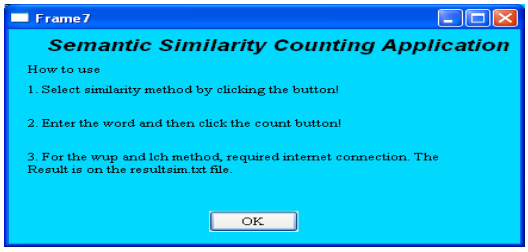
Gambar 5 Menu LCh

Pada menu utama, jika kita memilih menu lch, maka aplikasi ini akan memanggil program lch.py. Gambar 5 di atas merupakan tampilan dari program lch.py yaitu untuk melakukan perhitungan jarak semantik antara dua kata dengan metode lch. Untuk melakukan perhitungan, pertama kita masukkan dua buah kata yang akan dihitung. Kata tersebut diketikkan pada kolom Word 1 dan Word 2. Setelah itu, tekan tombol Count di mana tombol tersebut akan memanggil fungsi untuk menghitung menggunakan metode lch melalui koneksi internet ke situs marimba [10].



Gambar 6 Menu WuP

Pada menu utama, jika kita memilih menu wup, maka aplikasi ini akan memanggil program wup.py. Gambar 6 di atas merupakan tampilan dari program wup.py yaitu untuk melakukan perhitungan jarak semantik antara dua kata dengan metode wu and palmer. Untuk melakukan perhitungan, pertama kita masukkan dua buah kata yang akan dihitung. Kata tersebut diketikkan pada kolom Word 1 dan Word 2. Setelah itu, tekan tombol Count di mana tombol tersebut akan memanggil fungsi untuk menghitung menggunakan metode wup melalui koneksi internet ke situs marimba [10].



Gambar 7 Menu Help

Pilihan dari program yaitu WuP dan LCh untuk menghitung keserupaan label masih menggunakan aplikasi online, oleh karena itu dalam menggunakanya harus tetap terhubung dengan jaringan internet. Dalam proses pemanggilan situs marimba [10], digunakan tool pembantu yang dinamakan Lynx. Lynx adalah web-browser yang hanya untuk text dan merupakan klien Internet Gopher yang menggunakan *cursor-addressable* dan *character cell terminal.*

Hasil dari program tidak dapat ditemukan secara langsung pada jendela menu LCh ataupun WuP, tetapi dapat dilihat pada file resultsim.txt.

Program ini sudah diuji dengan menggunakan beberapa properti dari domain transportasi yang terdapat di beberapa situs di jaringan internet.

3. Kesimpulan

Dengan adanya fitur yang lebih measukan unsur antar muka grafis (GUI), maka apilkasi program untuk menghitung kesamaan atau keserupaan label lebih mudah untuk deigunakan.

Untuk selanjutnya, program aplikasi ini akan ditambahkan fitur lainnya dan akan dibuat sehingga tidak tergantung dengan jaringan internet untuk melakukan perhitungan, khusunya untuk metode WuP dan LCh.

4. Referensi

[1]. Sheth A.P, "Changin Focus On Interoperability in Information System: From System, Syntax, Structure , To Semantics", MITRE, Dec. 3rd 1998.

[2]. Ioannis, Varelas. "Semantic Similarity Methods in WordNet and Their Application to Information Retrieval on the Web", nike.psu.edu/widm05/p/p10-**varelas**.pdf, 2005

[3]. Banerjee, Satanjeev, and Ted Pedersen. "Extended gloss overlaps as a measure of semantic relatedness". In Georg Gottlob and Toby Walsh, editors, *IJCAI*, pages 805–810. Morgan Kaufmann, 2003.

[4]. Rahm,Erhard and Philip A. Bernstein. "A Survey of Approaches to Automatic Schema Matching". *The VLDB Journal*, 10:334–250, 2001.

[5]. Shvaiko,Pavel and Jerome Euzenat. "A Survey of Schema-based Matching Approaches". *J. Data Semantics IV*, pages 146–171, 2005.

[6]. Warin, Martin. "Using WordNet and Semantic Similarity to Disambiguate an Ontology". http://ling16.ling.su.se:8080/ PubDB/ doc repository/ warin2004usingwordnet. pdf, 2004.

[7]. Cohen, W.W., Ravikumar, P., Feinberg, Stephen, A "Comparison of String Distance Metrics for Name-Matching Tasks", JCAI – AAAI Workshop, 2003.

[8]. http://www.cut-the-knot.org/do_you_know/Strings.shtml

[9]. http://www.josef-willenborg.de/java/NGram/NGramApplet.html

[10]. http://marimba.d.umn.edu/cgibin/*similarity*.cgi

[11]. Bekke, J.H. ter, "Semantic Data Modeling", Prentice-Hall, Hemel- Hempstead, 1992.