KLASIFIKASI ARGUMEN SEMANTIK MENGGUNAKAN *SUPPORT VECTOR MACHINE* (SVM) TERHADAP KETERGANTUNGAN ARGUMEN SEMANTIK

SEMANTIC ARGUMENT CLASSIFICATION USING *SUPPORT VECTOR MACHINE* (SVM) WITH NEIGHBORING ARGUMENT INTERPENDENCE

Dwi Marlina Sari[1], Moch. Arif Bijaksana, Ir., M.Tech.[2], Siti Sa'adah, S.T., M. T.[3]

[1,2,3]Prodi S1 Teknik Informatika, Fakultas Teknik, Universitas Telkom
[1]dwimarlinasari18@gmail.com, [2]arifbijaksana@gmail.com, [3]tisataz@gmail.com

Abstrak

Argumen Semantik adalah salah satu bidang linguistik dalam mempelajari makna di dalam sebuah kalimat. Argumen sematik merupakan bagian dari teknik mengembangkan solusi *text mining*. Dengan melakukan klasifikasi argumen semantik, akan mengidentifikasikan argumen semantik ke dalam peran semantik yang lebih spesifik, sehingga dapat membantu dalam menggali informasi pada teks, seperti dapat menjawab pertanyaan *Who, Whom, When, Where, Why, and How*.

Tugas akhir ini, berfokus dalam melakukan klasifikasi argumen semantik menggunakan *feature baseline* dan *feature* tambahan yaitu *feature* argumen semantik tetangga yang menggunakan *database* PropBank. *Feature* argumen semantik tetangga dapat dimanfaatkan sebagai *feature* tambahan dalam membantu klasifikasi argumen semantik, dikarenakan pada masing-masing argumen dalam predikat saling ketergantungan. Klasifikasi argumen semantik dilakukan dengan menggunakan *classifier Support Vector Machine* (SVM). Dari uji skenario, hasil rata-rata akurasi klasifikasi argumen semantik mengggunakan *feature baseline* sebesar 63.91%, sedangkan hasil rata-rata akurasi berdasarkan *feature baseline* dan *feature* ketergantungan argumen tetangga dari predikat di dalam sebuah kalimat sebesar 71.21%.

Kata Kunci : klasifikasi argumen semantik, *feature*, ketergantungan argumen semantik

Abstract

Semantic argument is one of the linguistic scope in the study of meaning in a sentences. Semantic argument is part of a technique to develop text mining solutions. By performing semantic argument classification, will identify semantic argument into more specific role label, so it can assist extract information on the text, like can answer question such as Who, Whom, When, Where, Why, and How.

This research proposed to classify the semantic argument the baseline feature and semantic argument neighbor feature using PropBank database. Semantic argument neighbor feature can be used as additional argument to help semantic argument classification, because there is interdependence between all the neighboring argument of the predicate. Semantic argument classification will be use classifier Support Vector Machine (SVM). From the testing scenario, the average accuracy of semantic argument classification using baseline feature is 63.91%, while the average accuracy based on baseline feature and dependence neighboring argument feature of the predicate in a sentence amounting to 71.21%.

Keywords : ***semantic argument classification, feature, dependence semantic argument***

1. Pendahuluan

Informasi atau *knowledge* dapat diambil dari sebuah dokumen, artikel, bahkan dalam sebuah kalimat. Argumen semantik adalah salah satu bidang linguistik dalam mempelajari makna di dalam sebuah kalimat. Argumen semantik merupakan bagian dari teknik dalam mengembangkan solusi dari bidang *text mining* [5]. *Text mining* adalah upaya pencarian dan penambangan data yang berupa teks di dalam sebuah dokumen [6]. Salah satu bidang *text mining* yang memerlukan semantik adalah *task* pada *Natural Language Processing* (NLP) seperti *Information Extraction* (IE), *Question Answering* (QA), dan *Summarization* [8].

Representasi argumen semantik kalimat dapat disampaikan dalam bentuk ekstraksi informasi dan menjawab pertanyaan, seperti *Who*, *What*, *Whom*, *When*, *Where*, *Why*, dan *How* [10]. Representasi argumen semantik dijelaskan dalam bentuk peran semantik dan proses penentuan peran semantik tersebut dikenal sebagai pelabelan peran semantik atau *semantic role labeling*. Pelabelan peran semantik dibagi dalam dua cara, yaitu identifikasi

argumen semantik dan klasifikasi argumen semantik. Identifikasi argumen semantik mengklasifikasi apakah setiap elemen sintaksis merupakan sebuah argumen atau tidak. Sedangkan, klasifikasi argumen semantik mengidentifikasi argumen semantik ke dalam peran semantik yang lebih spesifik, seperti *ARG0*, *ARG1*, *ARG2*, *ARG3*, *ARG4*, dan *ARGM* [7]. Pelabelan peran semantik ini merupakan salah satu tahap sebelum memasuki proses *text mining* [6]. Dengan melakukan klasifikasi argumen semantik, maka akan membantu tahap untuk melakukan *text mining* nantinya, sehingga dapat menjawab pertanyaan *Who*, *What*, *Whom*, *When*, *Where*, *Why*, dan *How*.

Pada penelitian ini, lebih menerapkan klasifikasi argumen semantik pada *database* PropBank. Dalam melakukan klasifikasi argumen semantik digunakan *feature* konteks semantik, yang terdiri dari *feature baseline* dan *feature* argumen semantik tetangganya sebagai *feature* tambahan. *Feature* argumen semantik tetangga dapat dimanfaatkan untuk melihat ketergantungan dari semua argumen-argumen di dalam sebuah kalimat berdasarkan predikat pada masing-masing kalimatnya, sehingga dapat meningkatkan akurasi dalam klasifikasi argumen semantik dengan menggunakan metode *classifier Support Vector Machine* (SVM). Metode ini dipilih karena mampu mengklasifikasikan data berdimensi tinggi yang dalam konteks tugas akhir ini adalah berupa teks. Prinsip dari SVM adalah *Structural Risk Minimization* (SRM) yaitu meminimlakan *error* pada *training-set*.

2. Landasan Teori

2.1 Klasifikasi Argumen Semantik

Klasifikasi semantik adalah kajian analisis makna yang terdapat pada kata. Setiap kata selalu mengandung makna. Makna yang tergantung pada kata tersebut dapat diartikan sebagai argumen. Dan argumen tersebut dikelompokkan ke dalam satu klasifikasi yang memiliki sifat yang sama, sehingga menghasilkan suatu informasi yang baru.Pada PropBank, semantik direpresentasikan ke dalam bentuk peran semantik, seperti *ARG0*, *ARG1*, dan sebagainya di dalam sebuah kalimat dan proses menentukan peran semantik dikenal juga sebagai pelabelan peran semantik [3].

Pelabelan peran semantik merupakan masalah dalam pengklasifikasian yang dibagi dalam 2 subtugas, yaitu identifikasi argumen semantik dan klasifikasi argumen semantik. Identifikasi argumen semantik yaitu mengklasifikasikan masing-masing elemen semantik apakah sebuah argumen atau tidak. Klasifikasi argumen semantik yaitu mengidentifikasi argumen semantik ke dalam peran semantik yang lebih spesifik, seperti *ARG0*, *ARG1*, *ARG2*, *ARGM-ADV*, dan sebagainya [7].

Semantic role labeling atau pelabelan peran semantik merupakan proses pengidentifikasian argumen dari predikat dalam suatu kalimat, dan menentukan peran semantiknya. Mengidentifikasi peran semantik dapat memberikan level analisis semantiknya [6]. Contoh sederhana pelabelan peran semantik sebagai berikut:

"Father bought motorcycle."

Pada kalimat diatas terdapat sebuah kata kerja atau predikat yaitu "bought" dan dua argumennya adalah "Father" dan "motorcycle". Pemberian label berdasarkan peran semantik dilakukan terhadap argumen dimana "Father" mendapat label subyek dan "motorcycle" mendapat label obyek, sehingga struktur *verb-argument* yang terbentuk dari kalimat tersebut adalah sebagai berikut :

"[*ARG0* Father] [*TARGET* bought] [*ARG1* motorcycle]"

2.2 *Feature* Konteks Semantik

2.2.1 *Feature Baseline*

Dalam melakukan klasifikasi argumen semantik, salah satu tugas dalam menyelesaikan permasalahan klasifikasi adalah dengan menggunakan *machine learning*. Satu step yang terpenting dalam membangun akurasi klasifikasi adalah ketepatan dalam pemilihan *feature*. *Feature* yang sering digunakan dan dijadikan dasar *feature* pada riset sebelumnya [7] yang mengkategorikan ke dalam 3 tipe sebagai berikut :

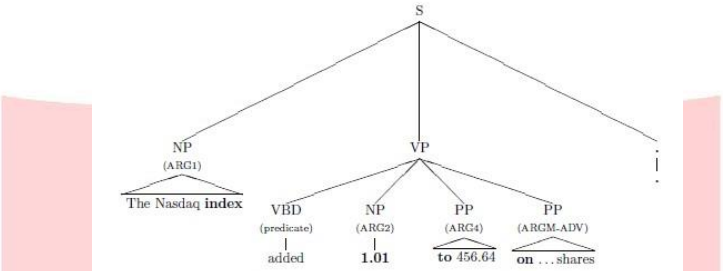
Tabel 2 - 1 Feature Baseline

Feature	Deskripsi
Semantic level features	
Predicate (Pr)	Predikat *lemma* di dalam struktur argumen predikat
Voice (Vo)	*Voice* gramatikal predikat, baik aktif maupun pasif
Subcat (Sc)	Aturan tata bahasa yang memperluas node induk predikat dalam *parse tree*
Argument-specific features	
Phrase type (Pt)	Kategori sintaksis dari unsur argumen
Head word (Hw)	*Head word* dari unsur argumen
Argument-prdicate relational features	
Position (Po)	Posisi relatif dari unsur argumen yang berhubungan dengan node predikat, baik kiri maupun kanan

2.2.2 *Feature* Argumen Semantik Tetangga

Kombinasi dari *feature* argumen semantik *baseline* dan *feature* argumen semantik tetangganya akan menjelaskan hasil saling ketergantungan diantara argumen-argumen semantik.

Berikut adalah contoh *parse tree* pada argumen semantik, dengan predikat "*added*" dan argumen-argumen semantiknya adalah *ARG1*, *ARG2*, *ARG4*, dan *ARGM-ADV*.



Gambar 2 - 1 Contoh Parse Tree [7]

Tabel 2 - 2 Penjelasan Parse Tree Berdasarkan Feature Baseline

Pr	Vo	Sc	Pt	Hw	Po	Ar
add	active	VP:VBD_NP_PP_PP	NP	index	L	ARG1
add	active	VP:VBD_NP_PP_PP	NP	1.01	R	ARG2
add	active	VP:VBD_NP_PP_PP	PP	to	R	ARG4
add	active	VP:VBD_NP_PP_PP	PP	on	R	ARGM-ADV

Pada paper acuan menggunakan *feature* akronim konteks dengan *subscript* untuk menunjukkan jenis tertentu dari *feature* konteks pada lokasi relatif dengan argumen saat diklasifikasikan. Contoh, menggunakan set notasi {-i..i} untuk menunjukkan *feature* konteks dengan *subscript* indeks j € {-i,….,i}, yaitu misalnya Hw{-1..1} yang menunjukkan *feature* Hw-1 dan Hw1. Dengan penggunaan *feature* akronim konteks dengan *subscript* ini, dapat dilihat keterkaitan antar sesama argumen-argumen tetangganya. Berikut pada Tabel 2 -3 menunjukkan feature tambahan sebagai feature argumen tetangga.

Tabel 2 - 3 Tabel Feature Argumen Tetangga

Feature	Deskripsi	Contoh
Pt*i*	Kategori sintaksis konteks i argumen semantik	Pt-1 dan Pt+1
Hw*i*	Headword konteks i argumen semantik	Hw-1 dan Hw+1
Po*i*	Position konteks i argumen semantik	Po-1 dan Po+1
Ar*i*	Role semantik konteks i argumen semantik	Ar-1

Feature argumen semantik tetangga dalam kasifikasi argumen semantik dapat digunakan dengan satu *feature baseline* saja atau dengan menggunakan semua *feature baseline.*

Contoh dengan menggunakan semua *feature baseline.*

Misal, Argumen semantik yang diklasifikasikan adalah 1.01 dengan set notasi {-1..1}.

Tabel 2 - 4 Contoh Menggunakan Semua Feature Baseline

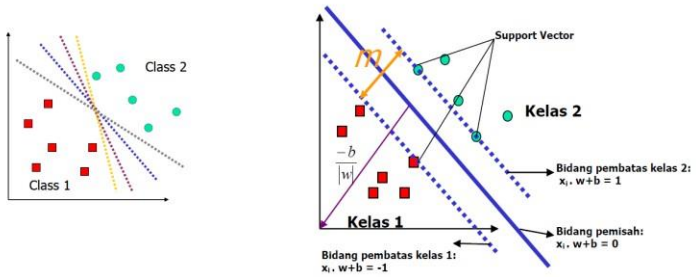
Pr	Vo	Sc	Pt	Hw	Po	Ar
add	active	VP:VBD_NP_PP_PP	NP	index	L	ARG1
add	active	VP:VBD_NP_PP_PP	NP	1.01	R	ARG2
add	active	VP:VBD_NP_PP_PP	PP	to	R	ARG4
add	active	VP:VBD_NP_PP_PP	PP	on	R	ARGM-ADV

2.3 *Support Vector Machine learning* (SVM)

SVM adalah metode *learning machine* yang bekerja atas prinsip *Structural Risk Minimization* (SRM) dengan tujuan menemukan *hyperplane* [8]. *Hyperplane* pemisah terbaik antara kedua kelas pada *input space* yang dapat ditemukan dengan mengukur *margin hyperplane* tersebut dan mencari titik maksimalnya. *Margin* adalah jarak antara *hyperplane* tersebut dengan *pattern* terdekat dari masing-masing kelas. *Pattern* yang paling dekat ini disebut dengan *support vector*.

Gambar 2 – 2 menampilkan beberapa *pattern* yang merupakan anggota dari dua buah *class*. *Pattern* yang terbagung pada *class* 1 disimbolkan dengan kotak warna merah, sedangkan *pattern* pada *class* 2 disimbolkan

dengan lingkaran warna hijau. Problem klasifikasi dapat diterjemahkan dengan usaha menemukan bidang pemisah antara kedua kelompok tersebut (*hyperplane*).



Gambar 2 - 2 Model data dengan berbagai alternatif bidang pemisah (kiri) dan model data dengan bidang pemisah terbaik dengan margin (m) terbesar (kanan) [13]

Pada *Gambar 2 – 2* yang sebelah kiri dapat dilihat alternatif bidang pemisah yang dapat memisahkan semua dataset sesuai dengan kelasnya. Namun, pada *Gambar 2 – 2* yang sebelah kanan, bidang pemisah terbaik tidak hanya dapat memisahkan data tetapi juga memiliki margin paling besar. Dua kelas dapat dipisahkan oleh sepasang pembatas yang sejajar. Bisang pembatas pertama membatasi kelas pertama sedangkan bidang pembatas kedua membatasi kelas kedua. Data yang berada pada bidang pembatas ini disebut *support vector*. Nilai margin (jarak) antara bidang pembatas (berdasarkan rumus jarak garis ke titik pusat) [13]. Dihitung dengan :

Jarak garis wx+b = c ke origin adalah (c-b)/|w|, maka

$$\frac{m}{2} = \frac{1-b-(-1-b)}{} = \qquad (2 - 1)$$

$$|\blacksquare\blacksquare \qquad w$$

Dimana :

m (Margin) = jarak antara dua bidang pembatas
w = normal bidang
b = posisi relatif terhadap origin

Input pada pelatihan SVM terdiri dari poin-poin yang merupakan *vector* dari angka-angka *real*. Data yang tersedia dinotasikan sebagai $\vec{x} \in \Re^d$ sedangkan label masing-masing dinotasikan sebagai $yi \in \{-1, +1\}$ untuk $i = 1,2,\ldots,l$, dimana l adalah banyaknya data. Diasumsikan kedua kelas -1 dan +1 dapat terpisah secara sempurna oleh *hyperplane* berdimensi d, yang didefinisikan [15]:

$$\blacksquare\blacksquare + \vec{b} = 0 \qquad (2 - 2)$$

Sebuah pattern $\vec{x}$ yang termasuk kelas -1 (sampel negatif) dapat dirumuskan sebagai *pattern* yang memenuhi pertidaksamaan :

$$\blacksquare\blacksquare + \vec{b} \leq -1 \qquad (2 - 3)$$

Sedangkan pattern *i* yang termasuk kelas +1 (sampel positif) :

$$\blacksquare\blacksquare + \vec{b} \leq +1 \qquad (2 - 4)$$

Margin terbesar dapat ditemukan dengan memaksimalkan nilai jarak antara *hyperplane* dan titik terdekatnya, yaitu 1/|| w ||. Hal ini dapat dirumuskan sebagai *Quadratic Programming* (QP) *problem*, yaitu mencari titik maksimal persamaan (2 - 6), dengan memperlihatkan *constraint* persamaan (2 - 6).

$$\min_{w} \rightarrow t(w) = \frac{1}{2} \|_{w} \rightarrow \|^2 \qquad (2 - 5)$$

$$y_i (x_i.w + b) - 1 \geq 0 \qquad (2 - 6)$$

Permasalahan ini dapat dipecahkan dengan berbagai teknik komputasi, diantaranya *Lagrange Multiplier* sebagaimana ditunjukkan pada persamaan berikut :

$$L(\vec{w}, b, \alpha) = \frac{1}{2}\|\vec{w}\|^2 - \sum_{i=1}^{l} \alpha_i (y_i((\vec{x}_i.\vec{w} + b) - 1)) \quad (i = 1,2,\ldots,l) \qquad (2 - 7)$$

αi adalah *Lagrange multipliers*, yang bernilai nol atau positif. Nilai optimal dari persamaan (2 - 8) dapat dihitung dengan meminimalkan L terhadap b, dan memaksimalkan L terhadap αi. Dengan memperhatikan sifat bahwa pada titik optial gradient $L = 0$, persamaan (2 - 8) dapat dimodifikasi sebagai maksimalisasi yang hanya mengandung αi saja, yaitu :

$$\sum_{i=1}^{l} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j x_i x_j \qquad (2 - 8)$$

Yang memenuhi,

$$\alpha_i > 0, \ (i = 1,2,\ldots,l) \ \sum_{i=1}^{l} \alpha_i y_i = 0 \qquad (2 - 9)$$

Dari hasil perhitungan di atas didapatlah *αi* yang kebanyakan bernilai positif. Data yang berkorelasi dengan *αi* yang positif inilah yang disebut sebagai *support vector*. Setelah menemukan *support vector*, maka *hyperplane* pun dapat ditentukan.

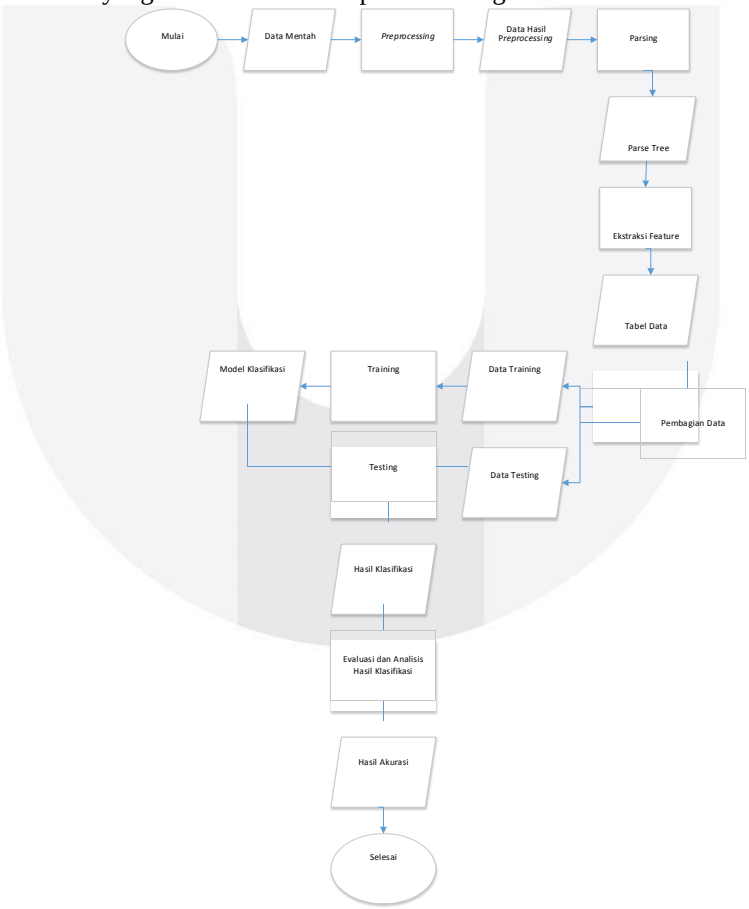
2.4 PropBank

PropBank merupakan salah satu *database* semantik yang digunakan untuk pelabelan *sematic role* kalimat berbahasa Inggris. Dengan melabeli peran semantik untuk setiap kata kerja dalam *corpus*, PropBank menyediakan sumber yang sifatnya *domain-independent*, dengan harapan dapat menghasilkan *Natural Language Processing* (NLP) yang lebih handal dan lebih luas. Fokus dari PropBank adalah pada struktur argumen dari kata kerja dan menyediakan *annotated-corpus* dengan peran semantik, termasuk peran yang ditampilkan sebagai argumen dan sebagai keterangan. PropBank mengizinkan kita pada langkah pertama untuk mencari frekuensi dari variansi sintaksis dalam praktiknya, masalah yang ditangani untuk *natural language understanding*, dan strategi yang sesuai [5].

PropBank seperti kamus yang menyediakan label-label argumen pada kata-kata yang berbahasa Inggris, contoh label argumen semantik seperti: *ARG0*, *ARG1*, *ARG2*, *ARG3*, dan *ARG4*. Selain label argumen nomor yang dianggap inti untuk kata kerja/predikat, terdapat juga label argumen semantik tambahan pada PropBank, yaitu label *ARGM* yang diikuti oleh tag sekunder untuk menunjukkan jenis tambahannya. Misalnya, kata "yesterday" bukan merupakan kata inti untuk kata kerja. Oleh karena itu, dilabeli dengan *ARGM* yang diikuti tag *–TMP*, yang menandakan sebagai waktu. Ada 18 tag sekunder untuk label *ARGM* pada PropBank [16].

3. Perancangan Sistem

Gambaran umum sistem yang akan dibuat dalam penelitian tugas akhir ini adalah sebagai berikut:



Gambar 3 - 1 Gambaran Sistem secara Umum

Berdasarkan *Gambar 3-1*, sistem yang akan dibangun dalam penelitian Tugas Akhir ini adalah sistem yang dapat mengkaji *feature-feature* yang mempengaruhi dalam melakukan klasifikasi argumen semantik. *Input* pada

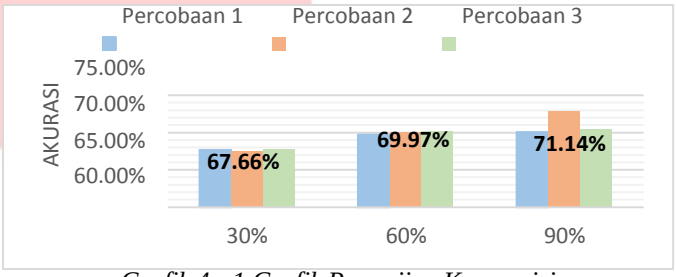
sistem ini adalah berupa kalimat yang berbahas Inggris yang sudah dilabeli argumen semantik dengan menggunakan database PropBank, yang dilakukan *preprocessing* terlebih dahulu, yaitu mengubah format data .XML menjadi kalimat yang mudah diproses, *case folding*, dan penghapusan tanda baca yang tidak diperlukan. Kemudian dibentuk ke dalam *parse tree* yang akan membantu dalam pembuatan ekstraksi *feature-feature* untuk pengklasifikasian argumen semantik. Dibentuk ke dalam tabel yang berisi *feature-feature* yang digunakan dan kelas argumen yang sudah ditentukan pada saat pelabelan argumen semantik. Data tabel digunakan untuk data *training* dan data *testing*, untuk mencocokkan hasil prediksi *classifier*-nya dengan argumen di PropBank. Setelah dianalisa hasil classifiernya, kemudian dianalisa dan dievaluasi perbandingan akurasi yang lebih tinggi dari segi *feature-feature* yang sangat berpengaruh dalam klasifikasi argumen semantik.

4. Hasil Pengujian dan Kesimpulan

4.1 Analisis Pengaruh Komposisi Data *Training*

Tabel 4 - 1 Tabel Pengujian Komposisi Data Training

Jumlah komposisi data training	Percobaan ke-		
	1	2	3
30%	67.68%	67.53%	67.76%
60%	69.73%	69.99%	70.19%
90%	70.07%	72.87%	70.48%



Grafik 4 - 1 Grafik Pengujian Komposisi Data Training

Dari hasil pengujian yang dilakukan, semakin banyak jumlah data yang digunakan sebagai data training maka semakin tinggi akurasi yang didapatkan. Hal ini disebabkan dengan banyaknya data *training*, model yang dibentuk akan lebih banyak menangani keberagaman data sehingga pada saat melakukan testing akan mampu mengklasifikasikan dengan lebih baik. Oleh karena itu, pada skenario yang kedua rata-rata akurasi yang lebih tinggi adalah pembagian data *training* 90% dan *testing* 10% dari dataset yaitu sebesar 71.14%.

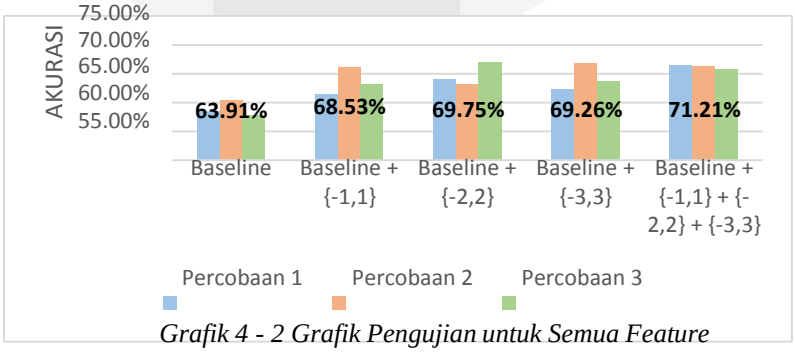
4.2 Analisis Pengaruh *Feature-Feature* yang Digunakan

4.2.1 Semua *Feature*

Perbandingan semua *feature baseline* dengan *window size feature* argumen tetangga, sebagai berikut :

Tabel 4 - 2 Tabel Pengujian untuk Semua Feature

Feature-Feature yang digunakan	Percobaan ke-		
	1	2	3
Baseline	63.50%	65.45%	62.77%
Baseline + {-1,1}	66.42%	71.05%	68.13%
Baseline + {-2,2}	69.10%	68.13%	72.02%
Baseline + {-3,3}	67.40%	71.78%	68.61%
Baseline + {-1,1} + {-2,2} + {-3,3}	71.53%	71.29%	70.80%



Grafik 4 - 2 Grafik Pengujian untuk Semua Feature

Berdasarkan hasil pengujian dilakukan untuk klasifikasi argumen semantik dapat dilihat terjadi peningkatan akurasi pada penggunaan *feature-feature* argumen semantik. Dari data hasil pengujian penggunaan semua *feature*

baseline dengan *feature* argumen tetangga didapatkan rata-raata akurasi tertinggi yaitu *feature baseline* ditambah dengan *feature* argumen tetangga dengan *window size* {-1,1}, {-2,2}, dan {-3,3} sebesar 71.21%.

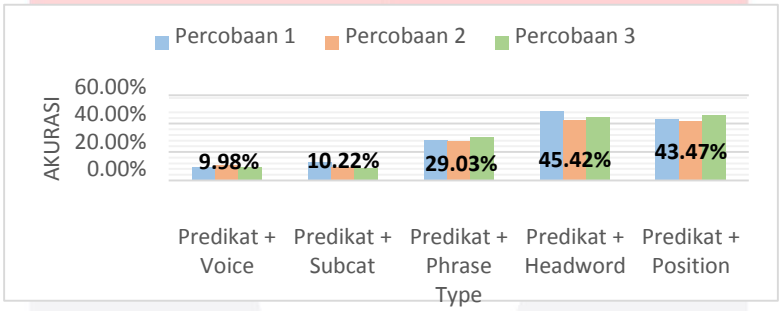
4.2.2 Masing-Masing *Feature*

Perbandingan masing-masing *feature baseline* dengan *window size* masing-masing *feature* argumen tetangga, sebagai berikut :

a. *Baseline*

Tabel 4 - 3 Tabel Pengujian untuk masing-masing Feature Baseline

Feature-Feature yang digunakan	Percobaan ke-		
	1	2	3
Predikat + Voice	9.25%	10.95%	9.73%
Predikat + Subcat	12.65%	9.25%	8.76%
Predikat + Phrase Type	28.22%	27.98%	30.90%
Predikat + Headword	49.15%	42.58%	44.53%
Predikat + Position	43.07%	41.61%	45.74%



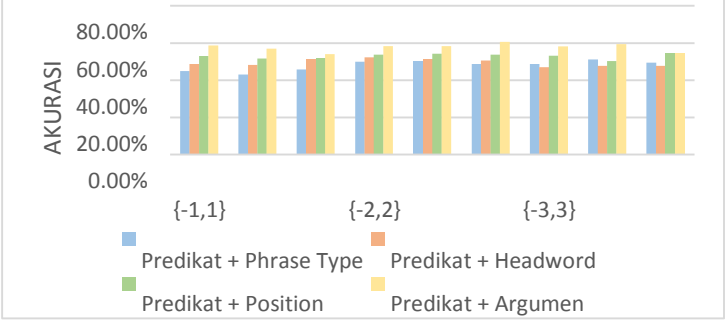
Grafik 4 - 3 Grafik Pengujian untuk masing-masing Feature Baseline

Pada hasil pengujian kombinasi predikat dengan masing-masing *feature baseline* didapatkan hasil rata-rata akurasi tinggi yaitu *feature headword* sebesar 45.42% dan *feature position* dengan akurasi 43.47%. Hal ini dikarenakan *feature headword* dan *feature position* berdasarkan *constituent* dalam kalimat. *Feature phrase type* dan *feature voice* terdapat beberapa nilai 'null' dikarenakan *constituent*-nya tidak berada dalam satu node yang sama dalam *parse tree*. Selain itu, untuk *feature voice* dan *feature subcat* berdasarkan satu kalimatnya bukan *constituent*. Oleh karena itu, *feature headword* dan *feature position* merupakan *feature* yang berpengaruh pada klasifikasi argumen semantik.

b. *Window size Feature* argumen tetangga

Tabel 4 - 4 Tabel Pengujian Window Size Argumen Tetangga

	{-1,1}			{-2,2}			{-3,3}		
Feature	Per.1	Per.2	Per.3	Per.1	Per.2	Per.3	Per.1	Per.2	Per.3
Predikat + Phrase Type	45.01%	43.07%	45.74%	**49.88%**	**50.61%**	**48.66%**	48.66%	51.09%	49.39%
Predikat + Headword	48.66%	48.18%	51.09%	**52.31%**	**51.09%**	**50.61%**	46.96%	47.69%	47.69%
Predikat + Position	53.04%	51.58%	52.07%	**53.77%**	**54.26%**	**53.77%**	53.28%	50.36%	54.50%
	{0,-1}			{0,-2}			{0,-3}		
Feature	Per.1	Per.2	Per.3	Per.1	Per.2	Per.3	Per.1	Per.2	Per.3
Predikat + Argumen	58.64%	56.93%	54.01%	**58.39%**	**58.39%**	**60.58%**	58.15%	59.37%	54.50%



Grafik 4 - 4 Grafik Pengujian Window Size Argumen Tetangga

Berdasarkan hasil pengujian untuk kombinasi *window size feature* argumen tetangga, nilai rata-rata akurasi yang tertinggi adalah argumen tetangga dengan *window size* {-2,2}. Hal ini dikarenakan *window size* {-2,2} lebih bervariasi dibandingkan dengan *window size* {-1,1} dan untuk *window size* {-3,3} lebih banyak memiliki nilai 'null' dikarenakan rata-rata kalimat tidak memiliki lebih dari 3 *constituent* di dalam satu kalimat.

4.3 Analisis Hasil Klasifikasi Argumen Semantik

```
=== Confusion Matrix ===

   a   b   c   d   e   f   g   h   i   j   k   l   m   n   o   p   q   r   s   t   u   v   <-- classified as
  99   0  12   2   0   0   0   0   0   0   0   1   1   0   0   0   0   0   0   0   0   0 |   a = 0.0
   0   0   2   0   0   1   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0 |   b = M
  17   0 120   1   0   0   0   0   0   0   0   0   2   0   0   0   0   0   0   0   0   0 |   c = 1.0
   4   0   7  34   0   0   0   0   0   0   0   1   0   1   0   2   0   0   0   0   0   0 |   d = 2.0
   1   0   1   0   0   2   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0 |   e = M ADV
   1   0   4   4   0  13   0   1   0   0   0   0   1   0   0   1   0   0   0   0   0   0 |   f = M TMP
   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0 |   g = M PRD
   2   0   0   2   0   1   0   2   0   0   0   0   0   0   0   0   0   0   0   0   0   0 |   h = M MNR
   0   0   0   0   0   0   0   1   3   0   0   0   3   0   0   0   0   0   0   0   0   0 |   i = M DIS
   0   0   0   0   0   0   0   0   0   5   1   0   0   0   0   0   0   0   0   0   0   0 |   j = M NEG
   1   0   0   0   0   1   0   0   0   0  10   0   0   0   0   0   0   0   0   0   0   0 |   k = M MOD
   0   0   2   4   0   1   0   1   0   0   0   2   0   0   0   0   0   0   0   0   0   0 |   l = 3.0
   2   0   2   1   0   1   0   0   0   0   0   0   2   0   0   0   0   0   0   0   0   0 |   m = M RCL
   0   0   1   2   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0 |   n = M DIR
```

Gambar 4 - 1 Gambar Hasil Klasifikasi

Pada *Gambar 4 – 1* merupakan contoh beberapa hasil klasifikasi yang tidak mampu diklasifikasi secara benar oleh sistem. Terdapat beberapa kesalahan yang ditandai oleh highlight berwarna kuning. Penyebab kesalahan, karena ada *feature* yang bernilai 'null' yaitu *feature phrase type* dan *feature voice*. *Feature-feature* yang digunakan pada klasifikasi argumen semantik bergantung pada hasil *parse tree*, sehingga jika tidak sesuai dengan *parse tree* maka nilai pada *feature* tersebut bernilai 'null'.

Daftar Putaka:

[1] Babko, O. (2005). PropBank Annotation Guidelines.

[2] Bonial, C., Bonn, J., & Conger, K. (n.d.). PropBank: Semantics of New Predicate Types.

[3] Gildea, & Palmer. (2002). The Necessity of Parsing for Predicate Argument Recognition.

[4] Gildea, D. (n.d.). Automatic Labeling of Semantic Roles. 28-3.

[5] Harlian, M. (2006). *Text Mining.* Austin.

[6] Indrawati, N. (2009). *Semantic Role Labeling Kalimat Bahasa Indonesia sebagai Preprocessing pada Text Mining.* Bandung.

[7] Jiang, Z. P., Li, J., & Ng, H. T. (n.d.). Semantic Argument Classification Exploiting Argument Interpendence.

[8] Nugroho, A. S., Witarto, & Arif Budi, H. D. (2003). Support Vector Machine.

[9] Palmer, Kingsbury, & Gildea. (n.d.). The Proposition Bank: An Annotated Corpus of Semantic Roles.

[10] Pradhan, S., & Hacioglu, K. (2004). Support Vector Learning for Semantic Argument Classification.

[11] Pradhan, S., Ward, W., & Martin, J. (2005). Towards Robust Semantic Role Labeling. *Association for Computational Linguistics*.

[12] Punyakanok, V., Roth, D., & Yih, W.-t. (n.d.). Generalized Inference with Multiple Semantic Role Labeling System.

[13] Sembiring, K. (2007). *Penerapan Teknik Support Vector Machine untuk Pendeteksian Intrusi pada Jaringan.* Bandung: Teknik Elektro dan Informatika, ITB.

[14] Sulianta, F., & Juju, D. (2010). *Data Mining.* Jakarta: PT Elex Media Komputindo.

[15] Widodo, Handayanto, & Herlawati. (2013, June 16). *Penerapan Data Mining dengan Matlab*. Bandung: Rekayasa Sains. Retrieved October 2013, from http://dataq.wordpress.com/2013/06/16/perbedaan-precision-recall-accuracy/

[16] Xue, N., & Palmer, M. (n.d.). Calibrating Features for Semantic Role Labeling.