



UNIVERSITAS INDONESIA

CROSS LANGUAGE WORD SENSE DISAMBIGUATION

SKRIPSI

ADITYA RAMA

1306397854

**FAKULTAS ILMU KOMPUTER
PROGRAM STUDI ILMU KOMPUTER
DEPOK
JUNI 2017**



UNIVERSITAS INDONESIA

CROSS LANGUAGE WORD SENSE DISAMBIGUATION

SKRIPSI

**Diajukan sebagai salah satu syarat untuk memperoleh gelar
Sarjana Ilmu Komputer**

ADITYA RAMA

1306397854

**FAKULTAS ILMU KOMPUTER
PROGRAM STUDI ILMU KOMPUTER
DEPOK
JUNI 2017**

HALAMAN PERNYATAAN ORISINALITAS

**Skripsi ini adalah hasil karya saya sendiri,
dan semua sumber baik yang dikutip maupun dirujuk
telah saya nyatakan dengan benar.**

Nama : Aditya Rama
NPM : 1306397854
Tanda Tangan :

Tanggal : 5 Juni 2017

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :

Nama : Aditya Rama

NPM : 1306397854

Program Studi : Ilmu Komputer

Judul Skripsi : Cross Language Word Sense Disambiguation

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Ilmu Komputer pada Program Studi Ilmu Komputer, Fakultas Ilmu Komputer, Universitas Indonesia.

DEWAN PENGUJI

Pembimbing : Mirna Adriani, Dra, Ph.D. ()

Pembimbing : Rahmad Mahendra, S.Kom., M.Sc. ()

Penguji 1 : ()

Penguji 2 : ()

Ditetapkan di : Depok

Tanggal : Juni 2017

KATA PENGANTAR

Segala puji bagi Allah SWT atas segala rahmat dan karunia yang telah diberikan, sehingga laporan penelitian ini dapat diselesaikan untuk memenuhi salah satu syarat menyelesaikan pendidikan program Sarjana Ilmu Komputer Universitas Indonesia. Penulis ingin menyampaikan terima kasih untuk pihak-pihak yang sudah secara langsung maupun tidak langsung membantu jalannya penelitian ini, diantaranya:

1. Keluarga penulis baik itu Ibu, Ayah, dan juga Kakak yang telah memberikan doa dan segala dukungan yang dibutuhkan selama jalannya penelitian.
2. Dra. Mirna Adriani, Ph.D dan Rahmad Mahendra, S.Kom., M.Sc. sebagai dosen pembimbing yang telah memberikan arahan, bimbingan, dan motivasi untuk menyelesaikan skripsi ini.
3. Nadiarani yang selalu membantu dalam memberikan semangat dan juga menuntut hasil penulisan secara berkala agar mendorong keinginan untuk menulis.
4. Jodi Prayogo, Hartico, Ilham Kurniawan, dan Firza Pratama sebagai pemberi masukannya pada pembuatan panduan anotasi dan juga proses evaluasi *word alignment*.
5. Teman-teman angkatan 2013 (Angklung) yang saling memberi dukungan dan doa untuk kelancaran setiap orang di dalamnya.

Depok, Mei 2017

Aditya Rama

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

Nama : Aditya Rama
NPM : 1306397854
Program Studi : Ilmu Komputer
Fakultas : Ilmu Komputer
Jenis Karya : Skripsi

demikian pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif (Non-exclusive Royalty Free Right)** atas karya ilmiah saya yang berjudul:

Cross Language Word Sense Disambiguation

berserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia-/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok
Pada tanggal : 5 Juni 2017
Yang menyatakan

(Aditya Rama)

ABSTRAK

Nama : Aditya Rama
Program Studi : Ilmu Komputer
Judul : Cross Language Word Sense Disambiguation

Cross Language Word Sense Disambiguation (CLWSD) merupakan salah satu pendekatan untuk menyelesaikan permasalahan disambiguasi makna kata di bidang NLP. Pendekatan ini memanfaatkan sebuah konsep dimana suatu kata dapat diterjemahkan menjadi beberapa kata yang berbeda tergantung dengan konteks dimana kata tersebut muncul. Keterbatasan data berupa *sense tagged corpus* menjadi salah satu permasalahan yang menghambat penelitian WSD di bahasa Indonesia ini. Pada penelitian kali ini, penulis akan menggunakan pendekatan CLWSD untuk *transfer sense* dari *sense tagged corpus* bahasa Inggris ke bahasa Indonesia dengan memanfaatkan korpus paralel dwibahasa. Hasil dari penelitian ini merupakan *sense tagged corpus* dalam bahasa Indonesia yang kemudian juga akan dicoba dalam sistem WSD yang dibuat sendiri. Berdasarkan hasil percobaan, dapat terlihat bahwa korpus hasil dari *sense transferring* menghasilkan data yang banyak dengan kualitas cukup baik. Selain *sense tagged corpus* tersebut, sistem WSD yang dibangun juga memiliki performa diatas baseline pembanding pada *target word* tertentu yang diberikan.

Kata Kunci:

Cross Language, Word Sense Disambiguation

ABSTRACT

Name : Aditya Rama
Program : Computer Science
Title : Cross Language Word Sense Disambiguation

Cross Language Word Sense Disambiguation (*CLSWD*) is one among the methods in solving word sense disambiguation problem in NLP field. This approach utilize a concept that a word can translated to many words depend on where that word appear. Limitation in data (*sense tagged corpus* in Indonesian language) become one problem that hold the development of research in Indonesian WSD. In this research, we will try using CLWSD to transfer sense from english sense tagged corpus into Indonesian by using parallel corpora. Result of this research is a sense tagged corpus in Indonesian language that will be tested by our implemented WSD system. Based on the result of the experiment, we could see that the corpus from the sense transferring process produce many data with sufficient quality. Beside the sense tagged corpora, WSD system that will be built also has a performance above the baseline on some given target words.

Keywords:

Cross Language, Word Sense Disambiguation

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PERNYATAAN ORISINALITAS	ii
LEMBAR PENGESAHAN	iii
KATA PENGANTAR	iv
LEMBAR PERSETUJUAN PUBLIKASI ILMIAH	v
ABSTRAK	vi
Daftar Isi	viii
Daftar Gambar	x
Daftar Tabel	xi
Daftar Kode	xii
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	2
1.3 Tujuan dan Manfaat Penelitian	2
1.4 Ruang Lingkup Penelitian	2
1.5 Metodologi Penelitian	2
1.6 Sistematika Penulisan	3
2 TINJAUAN PUSTAKA	5
2.1 Word Sense Disambiguation	5
2.1.1 WSD Bahasa Inggris	6
2.1.2 WSD <i>Cross Language</i>	7
2.2 Word Sense Induction	8
2.2.1 Pendekatan <i>Clustering</i>	8
2.2.2 Pendekatan <i>Cross Language</i>	9
2.3 Korpus Paralel dan <i>Comparable</i>	9
2.4 <i>Word Alignment</i>	10
2.4.1 <i>Task Word Alignment</i>	10
2.4.2 Tokenisasi dan Representasi Kalimat	11
2.4.3 Pengukuran Evaluasi	11
2.5 Support Vector Machine	12
2.6 Word Embedding Language Model	13

3	RANCANGAN PENELITIAN	15
3.1	Korpus Identik	15
3.2	Pembuatan <i>Sense Tagged Corpus</i> Bahasa Inggris	16
3.3	<i>Word alignment</i> pada Korpus Paralel	16
3.4	Evaluasi <i>Word Alignment</i>	17
3.5	Peningkatan Kualitas Hasil <i>Alignment</i>	17
3.6	<i>Sense Transferring</i>	18
3.7	Sistem WSD	18
4	IMPLEMENTASI	19
4.1	Pre-Processing	19
4.2	Pembuatan <i>Sense Tagged Corpus</i> Bahasa Inggris	19
4.3	<i>Word Alignment</i>	20
4.3.1	Pemrosesan <i>Word Alignment</i>	20
4.3.2	Post-Processing	21
4.4	Evaluasi <i>Word Alignment</i>	22
4.4.1	Pre-Processing	22
4.4.2	Proses Anotasi Data	23
4.4.3	Evaluasi	23
4.5	Peningkatan Kualitas <i>Alignment</i>	24
4.5.1	Pendekatan <i>Crawling</i>	24
4.5.2	Pendekatan <i>Bi-directional</i>	25
4.6	<i>Sense Transferring</i>	27
4.7	Sistem WSD	28
4.7.1	Pemilihan Fitur dan <i>Classifier</i>	28
4.7.2	Ekstraksi Fitur	28
4.7.3	Evaluasi Sistem	31
5	HASIL DAN ANALISIS	32
5.1	Korpus Identik	32
5.2	Pembuatan <i>Sense Tagged Corpus</i> Bahasa Inggris	32
5.3	Evaluasi <i>Word Alignment</i>	33
5.4	<i>Sense Transferring</i>	34
5.5	Sistem WSD	37
6	PENUTUP	39
6.1	Kesimpulan	39
6.2	Saran	39
	Daftar Referensi	41

DAFTAR GAMBAR

2.1	Arsitektur IMS	6
2.2	Performa IMS (Zhong dan Ng, 2010)	7
2.3	Akurasi Klasifikasi dari penelitian (Rudnick, 2011) dengan cross validation	8
2.4	Pengukuran Word Alignment	12
2.5	Hyperplane SVM pada (Fradkin dan Muchnik, 2006)	13
2.6	Word2Vec	14
3.1	Rancangan Sistem	15
4.1	Pre-Processing Giza	19
4.2	Bi-directional <i>Enhancement</i>	25
4.3	Ilustrasi Fitur Word Embedding	30

DAFTAR TABEL

5.1	Jumlah <i>instance</i> korpus bahasa Inggris	32
5.2	Evaluasi <i>Sense Transferring</i> Berdasarkan Jumlah Kelas	35
5.3	Evaluasi <i>Sense Transferring</i> Berdasarkan Sebaran Kelas	36
5.4	Evaluasi Sistem WSD	38

DAFTAR KODE

4.1	IMS	19
4.2	Word Alignment	20
4.3	A3-File	21
4.4	Word Alignment Evaluation	23
4.5	Word Alignment Enhancement	26
4.6	Fitur Bag of Words	29
4.7	Stanford POS Tagger	29

BAB 1

PENDAHULUAN

Bab ini membahas mengenai latar belakang penelitian, perumusan masalah, tujuan dan manfaat penelitian, ruang lingkup penelitian, metodologi penelitian, serta sistematika penulisan.

1.1 Latar Belakang

Word Sense Disambiguation (WSD) merupakan salah satu tugas untuk menentukan makna terbaik dari sebuah kata. Sebuah kata sendiri dapat memiliki beberapa makna dan bergantung pada konteks dimana kata tersebut muncul. Penentuan makna kata yang paling tepat ini secara tidak langsung dapat membantu beberapa *task Natural Language Processing* ataupun *Information Retrieval* lainnya seperti misalnya *machine translation*. Pendekatan yang biasa digunakan untuk menyelesaikan permasalahan WSD ini pada umumnya adalah pendekatan *machine learning* baik itu *supervised*, *semi-supervised*, ataupun *unsupervised*.

Pendekatan *supervised* yang digunakan untuk membangun sistem WSD membutuhkan data yang tidak sedikit. Data yang dibutuhkan dapat berupa *sense-tagged corpus* dimana isinya adalah kata-kata yang sudah mempunyai kelas makna kata yang tepat. Kebutuhan akan data yang relatif besar tersebut merupakan kendala yang ada pada bahasa-bahasa tertentu. Bahasa Inggris sebagai salah satu bahasa internasional mempunyai data yang cukup banyak untuk membangun sistem dengan *supervised learning*. Namun demikian, bahasa Indonesia sendiri termasuk dalam *under resource language* dimana data yang dapat dimanfaatkan untuk sistem WSD masih terbatas. Belum adanya data seperti *sense-tagged corpus* untuk membangun sistem WSD bahasa Indonesia, merupakan salah satu permasalahan yang dihadapi jika dibandingkan dengan bahasa Inggris.

Membangun Wordnet secara manual untuk memenuhi kebutuhannya sebagai inventaris makna kata membutuhkan waktu dan dana yang relatif tidak sedikit. Berdasarkan isu tersebut, metode lain dibutuhkan untuk membangun *supervised WSD system* yang akan mengatasi permasalahan *resource* yang belum memadai. Salah satu metode yang akan dicoba pada penelitian ini adalah pemanfaatan korpus dwibahasa dengan pendekatan *cross language*.

Pendekatan *cross language* dengan bahasa Inggris sebagai pasangan korpus

diharapkan dapat memperkaya *resource* yang masih kurang pada bahasa Indonesia.

1.2 Perumusan Masalah

Beberapa pertanyaan yang menjadi rumusan masalah dalam penelitian ini yaitu:

1. Bagaimana cara menerapkan pemindahan makna (*sense transfer*) dari korpus paralel untuk menghasilkan *sense tagged corpus* bahasa Indonesia?
2. Seberapa baik performa *WSD* yang dibangun untuk bahasa Indonesia tersebut?

1.3 Tujuan dan Manfaat Penelitian

Tujuan dari penelitian yang dilakukan adalah menghasilkan *sense tagged corpus* dalam bahasa Indonesia sebagai data makna kata, dan menghitung seberapa baik performa *wsd system* bahasa Indonesia yang dibuat.

1.4 Ruang Lingkup Penelitian

Penelitian berfokus pada pemindahan *sense* dari korpus paralel berbahasa Inggris ke Indonesia, dan melakukan *WSD task* pada hasil pemindahan makna kata tersebut.

Word sense disambiguation yang dilakukan pada penelitian ini hanya pada tingkatan *coarse-grained wsd*.

1.5 Metodologi Penelitian

Ada lima tahapan yang dilakukan pada penelitian ini. Penjelasan dari tiap tahapan adalah sebagai berikut.

1. Studi Literatur

Tahap ini berfokus pada pencarian informasi mengenai *WSD system* baik secara umum maupun teknik yang digunakan, dan juga *task* lain yang berkaitan dengan *WSD* seperti *word sense induction (WSI)*.

2. Perumusan Masalah

Masalah-masalah yang ada dalam penelitian nantinya dianalisis penyelesaiannya pada tahap ini.

3. Rancangan Penelitian

Proses yang melibatkan seluruh penelitian untuk menyelesaikan permasalahan yang ada.

4. Implementasi

Tahap ini merupakan implementasi dari rancangan yang sudah dibuat untuk memecahkan permasalahan yang ada.

5. Analisis dan Kesimpulan

Hasil percobaan dianalisis untuk mendapatkan gambaran seberapa baik performa dari sistem yang dibuat.

1.6 Sistematika Penulisan

Sistematika penulisan yang ada dalam laporan penelitian ini sebagai berikut:

- Bab 1 PENDAHULUAN

Bab ini akan menjelaskan mengenai latar belakang, perumusan masalah, tujuan penelitian, tahapan penelitian, ruang lingkup, metodologi, dan sistematika penulisan dari penelitian ini.

- Bab 2 TINJAUAN PUSTAKA

Bab ini akan menjelaskan mengenai konsep dan teori yang relevan dari hasil studi literatur yang telah dilakukan. Teori-teori yang dijelaskan meliputi *Word sense disambiguation*, *Word sense induction*, dan beberapa hal lain yang dibutuhkan pada penelitian.

- Bab 3 RANCANGAN PENELITIAN

Bab ini akan membahas perihal rancangan dari proses penelitian yang meliputi *sense tagging* korpus Inggris, *word alignment* Indonesia-Inggris, *sense transferring*, dan sistem WSD bahasa Indonesia.

- Bab 4 IMPLEMENTASI

Pada bab ini akan menjelaskan mengenai implementasi dari rancangan sistem yang dibuat.

- Bab 5 HASIL DAN ANALISIS

Pada bab ini, dijelaskan mengenai hasil penelitian beserta evaluasi dan analisisnya.

- Bab 6 PENUTUP

Kesimpulan dan saran dari hasil dan pelaksanaan penelitian akan dijelaskan pada bab ini.

BAB 2

TINJAUAN PUSTAKA

Bab ini membahas mengenai studi literatur yang digunakan selama penelitian. Studi literatur ini menjelaskan tentang hal-hal mendasar yang dibutuhkan dalam penelitian.

2.1 Word Sense Disambiguation

Word Sense Disambiguation merupakan salah satu penelitian di bidang NLP yang bertujuan untuk menentukan makna yang paling tepat dari suatu kata berdasarkan konteks kata tersebut ditemukan. Sebagaimana kata dalam suatu bahasa bisa memiliki makna lebih dari satu (polisemi), *task* ini akan menentukan makna kata mana yang paling tepat.

Penentuan makna kalimat dilakukan dengan pemberian informasi berupa kata yang menjadi *target* dan konteks berupa kalimat. Contoh proses disambiguasi yang dilakukan untuk kata **cokelat**:

K1: Roni memakan **cokelat** yang diberikan ibunya.

K2: Walaupun mobil **cokelat** itu mahal, dia sangat ingin membelinya.

Pada kalimat pertama (K1), **cokelat** yang dimaksud memiliki makna sebagai makanan yang terbuat dari buah *cokelat*. Sementara itu, Kata **cokelat** pada kalimat kedua (K2) memiliki makna yang berbeda, dimana kata tersebut merupakan satu keterangan warna. Penentuan makna yang tepat dapat dilakukan dengan bantuan informasi konteks dari kalimat dimana kata tersebut muncul. Pada K1, kata **memakan** memberikan informasi bahwa *cokelat* yang dimaksud adalah objek yang bisa dimakan. Kata yang memberikan informasi pada kalimat kedua adalah kata **berwarna** yang secara eksplisit menerangkan bahwa **cokelat** yang dimaksud adalah warna. Namun demikian, konteks maupun informasi yang bisa diambil dari kalimat tidak selalu eksplisit. Pada contoh kalimat seperti "Pohon cokelat tua di belakang rumahku sangat besar", cokelat yang dimaksud bisa bermakna "buah cokelat yang sudah tua" atau "berwarna cokelat tua".

Penentuan makna kata yang tepat oleh sistem WSD ditentukan berdasarkan konteks dari kata tersebut berada. Walaupun satu kata dapat memiliki beberapa makna, terdapat kecil kemungkinan bahwa kata yang sama digunakan dalam satu *discourse* untuk menyatakan makna yang berbeda sebagaimana "*one sense per*

discourse" (Gale et al., 1992).

2.1.1 WSD Bahasa Inggris

Salah satu sistem WSD untuk bahasa Inggris yang ada adalah "It Makes Sense" (IMS) yang dibuat oleh Zhi Zhong dan Hwee Tou Ng (Zhong dan Ng, 2010). Sistem dibangun menggunakan pendekatan *supervised learning* yang dapat digunakan untuk semua kata bahasa Inggris. Pada dasarnya, *classifier* yang dipilih untuk *task* ini adalah *support vector machine* (SVM). Arsitektur yang dibangun pada IMS dapat dilihat pada gambar berikut:



Gambar 2.1: Arsitektur IMS

Proses *pre-processing* pada IMS dilakukan dengan empat tahapan:

1. Mendeteksi batasan kalimat dengan *sentence splitter*
2. Tokenisasi dengan *tokenizer*
3. POS Tagging untuk semua token
4. Mengubah token menjadi lemma dengan *lemmatizer*

Ekstraksi fitur dilakukan dengan mengombinasikan:

1. POS Tag dari tiga buah kata di kiri dan kanan *target word*, serta kata itu sendiri.
2. Kata-kata sekitar pada konteks kalimat ataupun kalimat tetangganya. Kata-kata yang terkandung di dalam *stopwords* dan memiliki simbol atau angka dibuang dari kalimat tersebut. Kata-kata yang tersisa tersebut kemudian diubah menjadi bentuk kata dasarnya dalam huruf kecil.
3. *Local Collocation* dengan 11 buah *collocation* baik itu sebelum *target word* maupun setelahnya.

Pengujian seberapa baik performa IMS dalam melakukan WSD *task* mendapatkan hasil:

	SensEval-2 Fine-grained	SensEval-3 Fine-grained	SemEval-2007	
			Fine-grained	Coarse-grained
IMS	68.2%	67.6%	58.3%	82.6%
Rank 1 System	69.0%	65.2%	59.1%	82.5%
Rank 2 System	63.6%	64.6%	58.7%	81.6%
WNs1	61.9%	62.4%	51.4%	78.9%

Gambar 2.2: Performa IMS (Zhong dan Ng, 2010)

2.1.2 WSD *Cross Language*

Salah satu penelitian yang ada terkait dengan *cross language* WSD dilakukan untuk antara bahasa Quechua dan Spanyol (Rudnick, 2011). Penelitian tersebut dilakukan untuk menghasilkan sebuah sistem WSD yang nantinya dapat diintegrasikan dengan *Machine Translation System* untuk bahasa tersebut. Kata yang menjadi fokus pada penelitian tersebut hanyalah *adjective word* saja, karena sedikitnya kata infleksional dalam bahasa Quechua. Terdapat dua buah kamus digital yang digunakan untuk melakukan penerjemahan kata antara kedua bahasa. Wordnet bahasa Spanyol yang digunakan bukan sepenuhnya Wordnet *full version* yang membutuhkan biaya dan *lisence* untuk dapat diakses, namun merupakan subset dari Wordnet *full version* tersebut. Korpus yang digunakan sebagai *parallel resource text* merupakan Bitext (bibel Katolik) dengan total 31 ribu jumlah ayat.

Proses ekstraksi kata-kata yang ambigu dari kamus kedua bahasa dilakukan dengan cara menetapkan kata-kata dalam bahasa Spanyol yang dapat diterjemahkan menjadi lebih dari satu kata dalam bahasa Quechua yang berbeda (dengan tipe kelas kata adalah *adjective*).

Setelah mendapatkan kata-kata ambigu pada bahasa Spanyol tersebut, proses berikutnya adalah mencari dan mengumpulkan ayat yang mengandung kata ambigu tersebut untuk data *training*.

Proses disambiguasi yang dicoba menggunakan beberapa cara diantaranya KNN yang memperhitungkan jarak terdekat kedua node kata pada Wordnet, Simplified lesk Algorithm, dan penggunaan fitur untuk *classifier* tertentu. Pendekatan fitur yang digunakan adalah kata samping dari *target word*, synsets dari Wordnet Spanyol untuk kata tetangganya, dan hasil *parse* dengan menggunakan *dependancy parser*. Semua fitur tersebut dicoba pada *classifier* KNN, decision tree, dan naive bayes.

classifier	features	disagree	correctly disagree	accuracy
baseline	MFS in training instances			76.1
	MFS, corpus			69.1
	MFS, other stories			61.7
	uniform guess			38.9
Simplified Lesk		21.3	19.9	65.9
naïve bayes	words	17.0	44.9	75.8
	words, wn	19.2	40.1	74.0
	words, parse	16.2	42.8	75.1
decision trees	words	6.4	52.7	76.6
	words, wn	6.5	44.0	76.0
	words, parse	7.2	56.6	77.2
KNN	words	4.8	60.0	77.6
	wn	15.8	44.3	75.3
	words, wn	6.2	52.8	77.2
	words, parse	4.0	65.2	77.6

Gambar 2.3: Akurasi Klasifikasi dari penelitian (Rudnick, 2011) dengan cross validation

Simplified Lesk dalam percobaan tersebut menghadapi kasus dimana tidak terdapat kata yang ditemukan dari konteks word pada entri kamus. Fitur yang menghasilkan akurasi terbaik dari percobaan tersebut merupakan KNN dengan fitur konteks kata dengan fitur *parser*.

2.2 Word Sense Induction

Word Sense Induction (WSI) adalah sebuah *task* yang mempunyai fungsi utama untuk mendapatkan makna kata dari sebuah korpus atau teks yang belum dianotasi secara otomatis. WSI dapat dilakukan jika penelitian WSD yang ingin dilakukan tidak mempunyai cukup *resource* seperti misalnya Wordnet yang memadai. Terdapat berbagai macam pendekatan dalam melakukan WSI, diantaranya adalah dengan melakukan *clustering* kata (Denkowski, 2009), ataupun menggunakan pendekatan *cross language*.

2.2.1 Pendekatan *Clustering*

Dua kata dianggap dekat secara semantik jika memiliki *co-occurrence* dengan kata-kata tetangganya yang sama (Nasiruddin, 2013). Konsep tersebut mendasari cara WSI mendapatkan *sense* kata secara implisit berdasarkan hasil *cluster* yang terbentuk dari data atau teks mentah (teks yang tidak dianotasi).

Penarikan makna secara implisit dapat dicontohkan pada beberapa kalimat rujukan berikut (Denkowski, 2009):

1. A bottle of tezgüno is on the table.
2. Everyone likes tezgüno.
3. Tezgüno makes you drunk.
4. We make tezgüno out of corn.

Walaupun belum terdapat informasi eksplisit makna dari tezgüno, dapat disimpulkan bahwa tezgüno mengacu pada minuman beralkohol yang memabukkan. Penarikan kesimpulan ini didapatkan dari kemunculan kata tersebut dengan kata lain pada konteks yang sama.

Pada pendekatan *clustering* ini, makna kata bisa didapatkan secara implisit dari hasil *cluster* yang terbentuk, namun demikian pelabelan yang dilakukan untuk menentukan apa yang direpresentasikan *cluster* tersebut merupakan sebuah *task* tersendiri.

2.2.2 Pendekatan *Cross Language*

Selain pendekatan *clustering*, WSI juga dapat memanfaatkan fitur dimana satu kata dari suatu bahasa, dapat diterjemahkan menjadi beberapa kata di bahasa lain. Contoh kasus tersebut dapat dilihat pada kata "halaman" berikut:

(K1-Indonesia): Aku membaca 10 **halaman** buku Harry Potter

(K1-English): I read 10 **pages** of Harry Potter book

(K2-Indonesia): Ani tinggal di rumah dengan **halaman** yang sangat luas

(K2-English): Ani lives in a house with very large **yard**

Berdasarkan kedua pasangan kalimat tersebut, kata **halaman** dalam bahasa Indonesia dapat diterjemahkan menjadi dua buah kata dalam bahasa Inggris, yaitu *page* ataupun *yard*. Hal ini menunjukkan bahwa terjemahan dari suatu kata bergantung pada konteks dimana kata tersebut muncul.

2.3 Korpus Paralel dan *Comparable*

Terdapat dua macam korpus bilingual yang dapat dimanfaatkan untuk pemanfaatan *cross language* WSD yaitu korpus paralel dan *comparable*. Perbedaan utama terhadap kedua buah korpus berada pada seberapa identik kedua buah konteks yang dimilikinya. Korpus paralel memiliki kalimat dan kata-kata yang serupa antara dua buah pasangan *instance* di masing-masing korpus. Hal ini dapat dicontohkan misalnya dengan kalimat satu pada korpus bahasa Indonesia "Aku

makan" dengan "I eat" pada korpus bahasa Inggris. Berbeda dengan korpus paralel, *comparable* berarti kedua kalimat atau *instance* yang berpasangan hanya sebatas mirip/sama dalam suatu kategori kriteria tertentu. Dengan adanya korpus paralel dan *comparable* tersebut, dibutuhkan juga alat untuk menyelaraskan (*aligning*) konten pada kedua korpus tersebut. *Alignment* yang dapat dilakukan memiliki beberapa tingkatan mulai dari *scope* yang besar sampai kecil. *Scope* besar tersebut meliputi *alignment* dokumen yang mana fungsinya adalah menyelaraskan antar dokumen yang konten atau kriterianya sama. Tingkatan yang lebih kecil berikutnya yaitu kalimat dimana *alignment* dilakukan pada *level* kalimat (pasangan kalimat yang makna atau kriterianya sama). *Alignment* dengan tingkatan yang lebih spesifik lagi adalah kata (*word alignment*), dimana hasil yang didapat dari proses ini adalah pasangan kata pada kedua korpus dwibahasa yang selaras.

2.4 Word Alignment

2.4.1 Task Word Alignment

Tugas dari *word alignment* adalah menemukan korespondensi antara kata dan frasa pada teks paralel (Mihalcea dan Pedersen, 2003). Evaluasi ini akan membandingkan antara hasil *alignment* dari sebuah tool *word alignment* dengan hasil *alignment* manusia sebagai *gold standard*. Kasus yang dapat terjadi pada proses *alignment* ini adalah ketika terdapat kata yang tidak memiliki pasangan. Contoh dari kasus tersebut dapat dilihat pada pasangan kalimat berikut:

K1(en) : *He would do it regardless what people say*

K1(id) : Dia akan melakukannya segalanya

Bila melihat bahasa Indonesia sebagai sumber bahasa, maka kata "segalanya" pada kalimat tersebut tidak memiliki pasangan. Pada kasus seperti contoh diatas, kata yang tidak memiliki pasangan akan dipasangkan dengan *token NULL*.

Selain kata yang tidak memiliki pasangan, terdapat juga kasus dimana pasangan adalah berupa frasa. Hal ini dapat dilihat pada contoh berikut:

K2(en) : *The victim must be taken to the hospital*

K2(id) : Korban tersebut harus di bawa ke rumah sakit

Berangkat dari bahasa asal yaitu Indonesia, kata "rumah sakit" dipasangkan kepada kata "hospital". Hal ini dapat berlaku berkebalikan jika bahasa asal yang digunakan adalah bahasa Inggris seperti kata "untuknya" berpasangan dengan kata "for him".

2.4.2 Tokenisasi dan Representasi Kalimat

Pada penelitian (Mihalcea dan Pedersen, 2003), pemisah kata yang umum digunakan untuk tokenisasi adalah karakter spasi. setiap token dari hasil tokenisasi tersebut kemudian dianggap sebagai satu unit kata. Kata ini akan diindeks dengan angka untuk mempermudah proses *alignment* dan komputasi evaluasi. Contoh tokenisasi dan pemberian indeks pada kalimat "Aku ingin membeli mainan" adalah:

K3(id) : Aku ingin membeli mainan

K3(indeks) : 1 2 3 4

Angka yang menjadi indeks tersebut akan merepresentasikan kata yang bersesuaian, indeks 1 mewakili kata "Aku", indeks 2 mewakili "ingin", dan lain-lain. Beberapa *tool* merepresentasikan kalimat dan kata sebagai indeks angka tersebut untuk mempermudah pemrosesan tahap-tahap selanjutnya. Suatu *file* dapat berisi indeks dari kalimat dan kata yang ada pada kalimat tersebut seperti:

```
1 4 5 7
2 4 9 2
3 1 8 4
...
```

Dimana angka pertama merepresentasikan kalimat ke n pada korpus, dan angka-angka selanjutnya adalah indeks dari kata pada kalimat tersebut.

2.4.3 Pengukuran Evaluasi

Terdapat empat buah pengukuran berbeda, yaitu *precision*, *recall*, *f-measure*, dan *alignment error rate (AER)* (Mihalcea dan Pedersen, 2003). Diberikan hasil *alignment* dari program berupa A, dan *gold standard alignment* dari *evaluator* (manusia) sebagai G, masing-masing mengandung dua buah *set* yaitu *probable alignment* dan *sure alignment*. Pengukuran evaluasi dapat dilakukan dengan cara berikut:

$$P_T = \frac{|A_T \cap G_T|}{|A_T|} \quad (1)$$

$$R_T = \frac{|A_T \cap G_T|}{|G_T|} \quad (2)$$

$$F_T = \frac{2P_T R_T}{P_T + R_T} \quad (3)$$

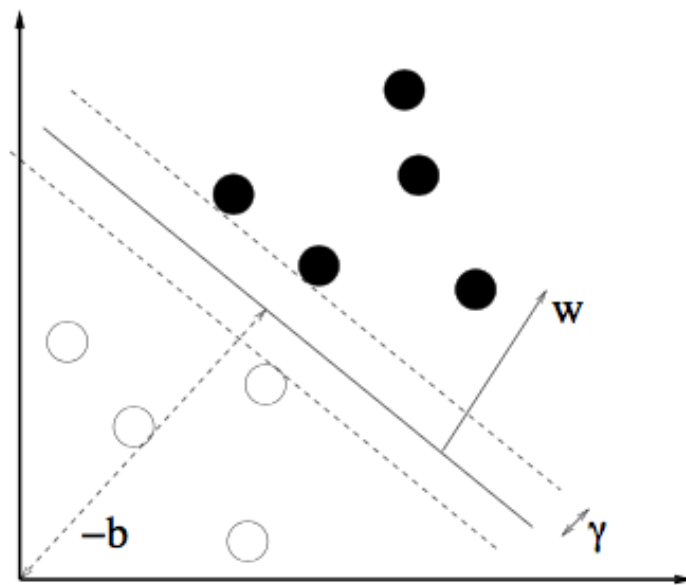
$$AER = 1 - \frac{|A_P \cap G_S| + |A_P \cap G_P|}{|A_P| + |G_S|} \quad (4)$$

Gambar 2.4: Pengukuran Word Alignment

Dikarenakan proses *word alignment* pada penelitian ini tidak membedakan antara *probable* dan *sure alignment*, maka semua dianggap merupakan set dari *sure alignment*.

2.5 Support Vector Machine

SVM merupakan salah satu *classifier* yang dapat digunakan untuk permasalahan klasifikasi. SVM termasuk sebagai metode klasifikasi yang populer dan telah digunakan untuk berbagai permasalahan seperti klasifikasi teks, *facial expression recognition*, analisis gen, *word sense disambiguation*, dan lain-lain. SVM dapat dikatakan sebagai salah satu metode yang membangun aturan yang dinamakan sebagai *linear classifier* yang secara teori akan menghasilkan kualitas prediksi dari *unseen data* yang baik (Fradkin dan Muchnik, 2006).



Gambar 2.5: Hyperplane SVM pada (Fradkin dan Muchnik, 2006)

Konsep dari cara SVM bekerja adalah dengan menemukan sebuah *hyperplane* dengan *margin* (jarak dari *hyperplane* dengan titik kelas terdekat) yang terbesar. Pemilihan *margin* dengan nilai terbesar ini ditujukan agar *classifier* lebih optimal dalam memisahkan objek dengan kelas yang berbeda.

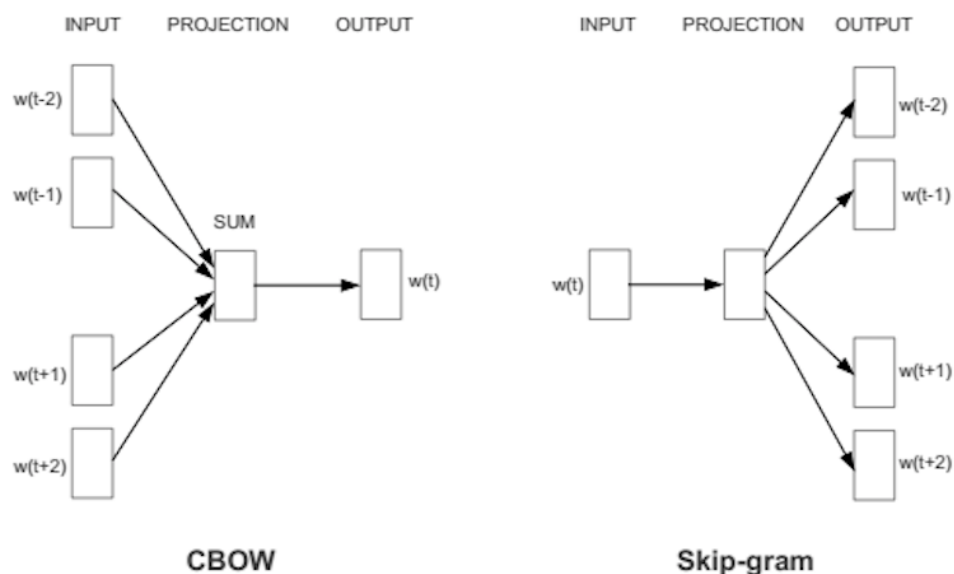
2.6 Word Embedding Language Model

Terdapat beberapa cara untuk merepresentasikan sebuah kata sebagai *input model*. Salah satu cara yang sederhana adalah dengan merepresentasikan kata sebagai *one hot vector*. Pada model ini, setiap kata di dalam sebuah korpus diberikan nomor indeks untuk membangun vektor yang mewakili keberadaan kata tersebut. Jika terdapat sebuah kata yang muncul pada konteks yang ingin direpresentasikan, indeks vektor yang sama dengan indeks kata tersebut akan bernilai 1. Bila terdapat sebuah korpus dengan jumlah kata unik berjumlah 4 dengan kata-kata "Ani", "marah", "kemarin", dan "malam" (sebuah vektor dengan panjang empat). Representasi *one hot vector* untuk kalimat "Ani marah" dapat ditulis dengan vektor $[1, 1, 0, 0]$.

Representasi *one hot vector* akan mempunyai panjang vektor yang besar jika korpus mempunyai jumlah kata unik yang besar. Terdapat bentuk representasi lain untuk membentuk vektor dari kata, salah satunya adalah dengan *word embedding*. *Word Embedding* menggunakan representasi bilangan *real* pada vektor untuk merepresentasikan sebuah kata berdasarkan hasil *training* dengan

suatu korpus. Contoh representasi dari *word embedding* pada suatu kata "makan" adalah vektor $[0.6, -0.3, \dots, 0.5]$ (misalnya). Vektor dari hasil *word embedding* mempunyai karakteristik dimana jarak antara dua buah vektor dari kata yang mirip secara semantik bernilai kecil(dekat). Bila misalkan pada data *training* untuk *word embedding* terdapat banyak kalimat-kalimat berbentuk "... makan Y ..." dimana Y adalah sebuah objek berupa makanan. Maka kata-kata yang mewakili Y seperti misalnya "burger", "apel", "steak", dan lain-lain, akan memiliki vektor yang mirip dan secara implisit dapat saling menggantikan untuk menempati posisi Y tersebut. Berdasarkan keterdekatan vektor tersebut, *word embedding* mampu untuk menangkap semantik dari kata-kata yang ada pada korpus.

Salah satu model *word embedding* yang dapat digunakan adalah Word2Vec (Mikolov et al., 2013). Terdapat dua buah arsitektur Word2Vec tersebut, yaitu skip-Gram dan *Continuous bag-of-words* (CBOW). Arsitektur pada CBOW memiliki pendekatan untuk memprediksi setiap kata berdasarkan kata-kata disekelilingnya. Lain halnya dengan arsitektur tersebut, Skip-Gram akan memprediksi kata-kata di sekeliling berdasarkan kata yang diberikan. Gambaran dari skip-gram dan CBOW dapat dilihat pada gambar berikut:

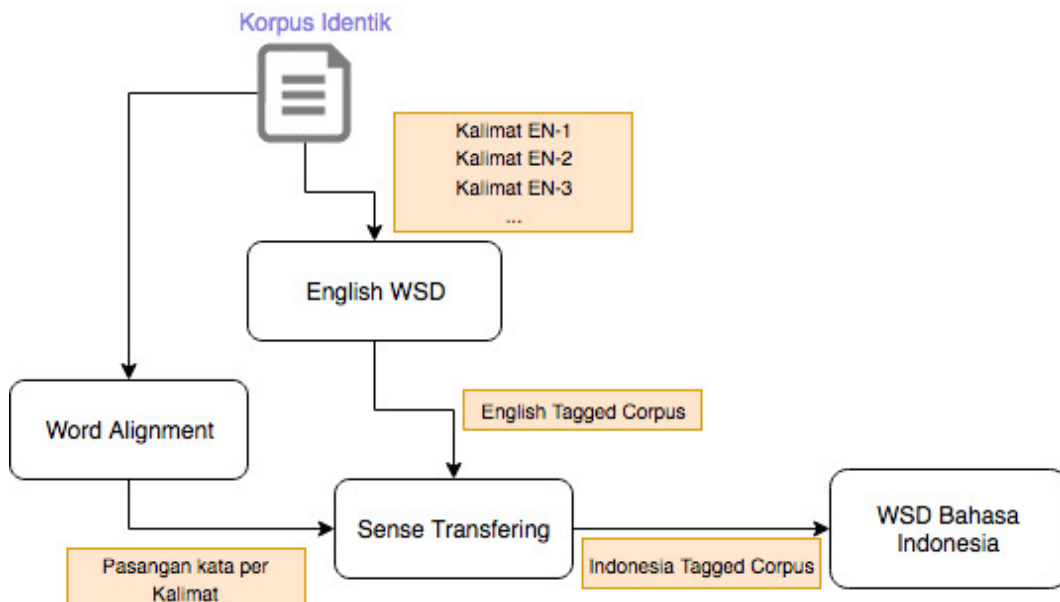


Gambar 2.6: Word2Vec

BAB 3

RANCANGAN PENELITIAN

Bab ini akan menjelaskan gambaran proses penelitian secara keseluruhan yang terdiri dari pembuatan *sense tagged corpus* bahasa Inggris, *word alignment* korpus paralel, peningkatan kualitas dan evaluasi *word alignment*, pemindahan *sense* dari korpus bahasa Inggris, dan sistem WSD yang diimplementasikan. Diagram dari rancangan sistem yang akan dibuat pada penelitian ini dapat dilihat pada gambar berikut:



Gambar 3.1: Rancangan Sistem

3.1 Korpus Identik

Korpus utama yang digunakan sebagai sumber data penelitian ini adalah korpus identik. Korpus identik berisi pasangan kalimat-kalimat dalam bahasa Indonesia dan Inggris. Kalimat yang berpasangan di dalamnya sebagian besar mempunyai makna konten yang paralel walaupun terdapat juga yang *comparable*. Korpus identik ini mempunyai 88.918 buah pasangan kalimat di dalamnya.

3.2 Pembuatan *Sense Tagged Corpus* Bahasa Inggris

Pembuatan *sense tagged corpus* bahasa Inggris dilakukan dengan menggunakan *tool* IMS untuk mendapatkan makna terbaik yang dapat ditag oleh *tool* tersebut. Makna kata hasil dari proses ini akan dipindahkan ke kata yang bersesuaian pada kalimat yang sama pada bagian *sense transferring*. *File* yang diberikan sebagai masukan dari IMS adalah kalimat-kalimat pada bahasa Inggris yang berasal dari korpus identik.

3.3 *Word alignment* pada Korpus Paralel

Word alignment pada korpus berbahasa Inggris dan Indonesia menggunakan *tools word alignment* bernama Giza++. *Tool* ini merupakan salah satu *word alignment tools* pada *statistical machine translation* (SMT) yang dapat digunakan untuk memasang kata-kata pada dua buah korpus atau lebih. Terdapat beberapa *word alignment tools* lain seperti Berkeley aligner, anymalalign, dan lain-lain. Penye-larasan kata ini digunakan untuk kebutuhan pemindahan *sense* dari kata bahasa Inggris ke kata dalam bahasa Indonesia.

Proses *alignment* yang dilakukan dengan Giza++ meliputi tahap-tahap berikut:

1. Mempersiapkan kedua buah *file* yaitu korpus bahasa asal (*source*) dan korpus bahasa tujuan (*target*). Kedua *file* ini berpasangan dalam setiap barisnya. Baris pertama dalam *file* pertama berpasangan dengan baris pertama pada *file* kedua sampai akhir baris pada kedua *file*.
2. Menghasilkan *file* perbendaharaan kata dari kedua bahasa dan *list* indeks perbendaharaan kata pada tiap kalimat yang sudah diselaraskan
3. Menghasilkan *cooccurence file* dari kosa kata dan pasangan kalimat tersebut
4. Proses *alignment* yang menghasilkan beberapa macam *output file*

Terdapat satu buah *output file* Giza++ yang berisi pasangan-pasangan kalimat dengan kata-kata yang sudah diselaraskan dengan translasinya dalam bahasa tujuan. Hasil ini merupakan *best viterbi alignment* menurut Giza++.

Pada skenario *alignment* dengan bahasa Indonesia sebagai *source* dan bahasa Inggris sebagai *target*, satu kata dalam bahasa Indonesia akan dipasangkan dengan tepat satu kata dalam bahasa Inggris.

3.4 Evaluasi *Word Alignment*

Word alignment hasil dari *tool* Giza++ dievaluasi dengan menggunakan *anotator* hasil *alignment* dari *anotator* yang akan ditujukan sebagai *gold standard*. Nilai-nilai yang akan dihitung meliputi *precision* (P), *recall* (R), dan *F-score*. Metode evaluasi keseluruhan meliputi:

1. Pemilihan *random sampling* sebanyak dua ratus buah pasangan kalimat
2. Masing-masing *anotator* memasang-masangkan kata yang tepat pada masing-masing pasangan kalimat, dengan asumsi bahwa anotasi manusia sebagai *gold standard*
3. Hasil anotasi manusia dan keluaran dari *tool* Giza dibandingkan untuk mendapatkan ketiga nilai P, R, F-Score, dan *agreement* kedua *anotator*.

3.5 Peningkatan Kualitas Hasil *Alignment*

Proses peningkatan kualitas hasil *alignment* diperlukan untuk meminimalisir kesalahan pemasangan kata-kata pada proses sebelumnya. Permasalahan yang terjadi adalah adanya pasangan-pasangan kata yang tidak benar seperti pada halnya kata "lapangan" misalnya yang dipasangkan dengan kata dalam bahasa Inggris *field*, *ground*, *involved*, *job*, *program*, dan beberapa kata lainnya. Peningkatan kualitas *alignment* ini dilakukan dengan dua buah pendekatan, yaitu dengan bantuan *online dictionary* bahasa Indonesia-Inggris dan *bi-directional alignment*.

Pendekatan dengan bantuan kamus diterapkan dengan mencari kata terjemahan pada bahasa Inggris untuk menentukan apakah *alignment* benar atau salah. Pada pendekatan kedua, dilakukan *inverse alignment* antara bahasa Indonesia ke Inggris. Jika pada proses awal *alignment* dilakukan dengan menerapkan bahasa Indonesia sebagai *source* dan Inggris sebagai *target*, kali ini dilakukan proses yang berkebalikan. Pemanfaatan hasil *alignment* korpus bahasa Inggris ke Indonesia akan menghasilkan pasangan-pasangan kata dengan tingkat kesalahan *alignment* lebih kecil dari *alignment* satu arah saja. Metode yang akan dilakukan adalah dengan memeriksa setiap pasangan kata dari bahasa Indonesia yang mana merupakan kata dalam bahasa Inggris, apakah kata tersebut memiliki pasangan dalam *inverse alignment* Giza.

3.6 *Sense Transferring*

Pemindahan makna kata dilakukan dengan tiga buah *sub-process* yang terdiri dari pemasangan antar kalimat, pemeriksaan kata, dan *sense transferring*.

1. Pemasangan antar kalimat yang bersesuaian dengan kata-kata yang berpasangan. Pada contoh kata "halaman" yang berpasangan dengan "courtyard", maka pasangan kalimat "Aku bermain di halaman" akan dipasangkan dengan kalimat "I play at the courtyard".
2. Pemeriksaan untuk kata yang saling berpasangan dari hasil *alignment* dan kamus hasil *alignment enhancement*.
3. *Sense* dari kata yang menjadi *target* tersebut kemudian diperiksa dengan *sense* kata yang sama yang sudah pernah dipindahkan dari pasangan kalimat lain.
4. Jika *sense* yang ingin dipindahkan "mirip" dan memiliki kedekatan makna dibatas *threshold*, maka *sense* yang akan dipindahkan hanya salah satu saja. Proses ini diperlukan untuk meminimalisir adanya satu kata bahasa Indonesia yang mempunyai lebih dari satu *sense* yang mirip dari definisi makna tersebut.
5. Bila "courtyard" memiliki *sense* yang artinya adalah "halaman rumah", maka "halaman" pada kalimat "Aku bermain di halaman" memiliki *sense* "halaman rumah".

3.7 *Sistem WSD*

Sistem WSD yang dibangun adalah dengan menggunakan pendekatan *supervised learning*. Hasil dari pemindahan makna kata akan digunakan sebagai *training* dan *testing* data untuk menguji performa dari sistem yang dibangun. *Classifier* yang digunakan dalam sistem WSD ini adalah SVM. Pengujian dilakukan dengan menggunakan beberapa fitur seperti *bag of words*, *POS Tag*, dan *word embedding*. Fitur *bag of words* menggunakan *window* sebanyak dua buah kata kanan dan kiri kata tujuan sebagai kata konteks. *POS Tag* dan vektor *word embedding* juga akan diimplementasikan pada penelitian ini. Performa dari sistem WSD akan dilihat berdasarkan perhitungan F1-score *micro* dari hasil klasifikasi.

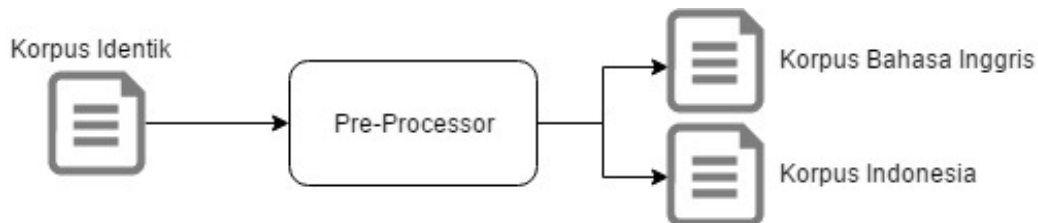
BAB 4

IMPLEMENTASI

Bab ini akan menjelaskan perihal implementasi dari rancangan yang sudah dibuat pada bab sebelumnya.

4.1 Pre-Processing

Perlu dilakukan *pre-processing* untuk memisahkan kalimat bahasa Inggris dan bahasa Indonesia dari korpus identik menjadi dua buah *file* paralel untuk dapat diproses pada tahap-tahap berikutnya.



Gambar 4.1: Pre-Processing Giza

4.2 Pembuatan *Sense Tagged Corpus* Bahasa Inggris

Sense Tagged Corpus dibuat dengan menggunakan bantuan IMS untuk memberikan tag pada korpus bahasa Inggris. *Pre-processing* dilakukan terlebih dahulu terhadap korpus bahasa Inggris seperti menghilangkan tanda baca dan mengubah semua token menjadi huruf kecil. Proses *tagging* dilakukan dengan menjalankan perintah:

Kode 4.1: IMS

```
./testPlain.bash <model> <file_input> <file_output> <  
file_index_sense>
```

model yang digunakan IMS pada penelitian ini adalah model yang tersedia pada *website software NUS* berdasarkan versi Wordnet 3.0. Model yang digunakan tersebut meliputi hasil *training* kata-kata dalam bahasa Inggris yang sudah dilakukan di penelitian IMS. Proses yang dilakukan IMS dalam melakukan *tagging sense* adalah dengan mengiterasikan melakukan *sentence splitter*, *tokenizing*, *POS Tagging*, dan *lemmatizing*. Setelah proses itu dilakukan, ekstraksi fitur dilakukan

sebelum hasilnya masuk ke dalam *classifier* berdasarkan model kata yang sudah ada.

Hasil *output* dari *tool* tersebut adalah korpus dengan kata-kata sudah ditag dengan makna kata yang mungkin (dalam bentuk *sense key*). Makna kata yang diambil untuk kata tersebut merupakan *sense key* dengan nilai kemungkinan terbesar. *Sense key* merupakan *identifier* unik yang menyimpan arti dari suatu kata pada Wordnet Princeton dengan format "lemma_suatu_kata%key". Untuk mempermudah pemakaian *sense tagged corpus* ini pada proses selanjutnya, dilakukan *post-processing* untuk mengubah hasil keluaran ke dalam format berikut:

```
<sentence>kata-1||sensekey-1 kata-2||sensekey-2 ...</sentence>
<sentence>kata-n||sensekey-n kata-m||sensekey-m ...</sentence>
...
```

Contoh dari kalimat yang sudah diberikan *tag* sampai keluar dari *post-processing* adalah:

```
<sentence>years||year%1:28:01:: animals||animal%1:03:00:: have||
have%2:40:04:: caused||cause%2:36:00:: havoc||havoc
%1:04:00::</sentence>
```

Pada contoh tersebut, kata *years* memiliki *sense key* berupa "year%1:28:01::", dimana berdasarkan Wordnet mempunyai arti sebagai 'a period of time containing 365 (or 366) days'. Tidak semua kata dalam korpus bahasa Inggris ditag oleh IMS, kata-kata sapaan seperti "I", "you", "a", "the", dan beberapa kata lainnya tidak diberikan *sense key*.

4.3 Word Alignment

4.3.1 Pemrosesan Word Alignment

Proses *word alignment* menggunakan dua buah *file* yaitu korpus berbahasa Indonesia dan Inggris yang sudah dipisah dari *pre-processing*. Perintah berikut digunakan untuk melakukan *word alignment* dengan Giza pada penelitian ini:

Kode 4.2: Word Alignment

```
# Lakukan pada direktori Giza
./plain2snt.out [source_language] [target_language]

# proses diatas menghasilkan tiga buah file yaitu dua buah file
vocabulary yang berisi indeks dengan kata (bahasa asal, dan
bahasa tujuan), dan satu buah file snt yang berisi \textit{
alignment} dari kalimat.
```

```

./snt2cooc.out [source_language_vcb_file] [
    target_language_vcb_file] [snt_file] > [cooccurrence_file]

# proses snt2cooc akan menghasilkan \textit{cooccurrence file}

./GIZA++ -S [source_language_vcb] -T [target_language_vcb] -C [
    snt_file] -CooccurrenceFile [cooc_file]

```

Giza mengeluarkan beberapa *file* hasil dari proses tersebut. *Output* yang akan digunakan diantaranya adalah *file* bernama A3.final yang merupakan pasangan kalimat dengan kata-kata yang sudah dipasangkan sesuai dengan prediksi terbaik hasil pemrosesan Giza.

4.3.2 Post-Processing

Setelah mendapatkan *file* A3 dari Giza, dilakukan *post-processing* untuk menghasilkan *file* yang dengan mudah dapat diproses untuk melakukan *sense transferring*.

Berikut ini merupakan salah satu contoh pasangan kalimat pada *file* A3 keluaran Giza:

Kode 4.3: A3-File

```

# Sentence pair (47183) source length 9 target length 9 alignment
  score : 6.85298e-13
Undang-Undang No 14 tahun 2008 tentang Kebebasan Memperoleh
  Informasi
NULL ({ }) Law ({ 1 }) No ({ 2 }) 14 ({ 3 }) of ({ }) 2008 ({ 4 5
  }) on ({ 6 }) Freedom ({ 7 8 }) of ({ }) Information ({ 9 })

```

Pembacaan pasangan kata berdasarkan hasil keluaran dilakukan dengan indeks nomor kata yang berada pada dalam kurung kata di bahasa Inggris. Karena pemisah token *by default* adalah spasi, maka kata "Undang-Undang" adalah kata dengan indeks nomor 1, kata "No" adalah kata dengan indeks nomor 2, dan berlaku hal yang sama sampai kata "Informasi". Pada kata bahasa Inggris, "Law" dipasangkan dengan indeks satu yang mana adalah "Undang-Undang", kata "No" dipasangkan dengan indeks dua yang mana adalah "No".

Terdapat dua buah *post-processing* yang dilakukan dengan tujuan masing-masing untuk:

1. Penyimpan pasangan kata-kata yang bersesuaian untuk sistem WSD.

2. Sebagai *resource* untuk proses *enhancement word alignment*.

Untuk keperluan nomor satu, bentuk *output* diproses menjadi bentuk lain dengan format:

```
<pair>kata_en_1||kata_id_1 kata_id_2</pair?##<pair>kata_en_2||
kata_id\_3</pair>...</pair>
```

Contoh dari hasil *post-processing* pada pasangan kalimat sebelumnya adalah:

```
<pair>law||undang-undang</pair>##<pair>no||no</pair>##<pair>
>14||14</pair>##<pair>2008||tahun 2008</pair>##<pair>on||
tentang</pair>##<pair>freedom||kebebasan memperoleh</pair>##<
pair>information||informasi</pair>
```

Hasil ini kemudian disimpan sebagai sebuah *file* sendiri yang akan digunakan kembali pada sistem WSD nantinya. Sementara itu, keperluan nomor dua difokuskan untuk membuat *dictionary* yang akan ditingkatkan kualitasnya pada tahap berikutnya. Untuk menghasilkan *file* yang dibutuhkan pada nomor kedua, dilakukan pengumpulan pasangan kata bahasa Indonesia dengan bahasa Inggris. Bila misalkan pada kalimat ke-*n* terdapat kata "undang-undang" yang dipasangkan dengan "law", dan pasangan kata "undang-undang" dengan "regulation" pada kalimat lain (kalimat ke-*m*, dimana $m \neq n$). Berdasarkan kedua kalimat tersebut, maka kata "undang-undang" akan berpasangan dengan dua kata yaitu "law", dan "regulation".

4.4 Evaluasi *Word Alignment*

4.4.1 Pre-Processing

Pertama, dilakukan pemilihan acak terhadap 200 pasang kalimat yang merupakan hasil keluaran Giza(*file* A3). Setiap pasang tersebut meliputi dua buah isi yaitu kalimat bahasa Indonesia, dan kalimat bahasa Inggris dengan tanda *alignment* Giza. Proses selanjutnya adalah menyiapkan 200 pasang kalimat tersebut untuk dievaluasi oleh anotator. Terdapat beberapa proses dalam mempersiapkan data untuk evaluasi oleh anotator.

1. Pertama, pada setiap kata dalam bahasa Indonesia diberikan sebuah tanda indeks berupa angka untuk mempermudah proses evaluasi anotator nantinya. Pada proses tersebut, kalimat "Aku ingin makan" sebagai perumpamaan, diubah menjadi "Aku(1) ingin(2) makan(3)". Angka tersebut diperuntukan untuk mempercepat dan mempermudah kerja anotator nanti untuk melihat pasangan kata dari kalimat bahasa Inggris.

2. Kedua, kosongkan nomor indeks hasil *alignment* Giza pada kalimat bahasa Inggris. Perumpamaan pada kalimat "NULL ({ }) i ({ 1 }) want ({ 2 }) to ({ }) eat ({ 3 })" akan berubah menjadi "NULL ({ }) i ({ }) want ({ }) to ({ }) eat ({ })" yang nantinya akan diisi secara manual oleh anotator.

4.4.2 Proses Anotasi Data

Setelah 200 buah pasangan kalimat(data) tersebut selesai dipersiapkan, anotator akan melakukan anotasi data (*alignment*) secara manual dengan panduan yang diberikan oleh penulis. Panduan tersebut meliputi keterangan dari *task word alignment*, format data yang diberikan dan cara pembacaannya, dan yang paling utama adalah cara melakukan proses *alignment*. Panduan anotasi yang diberikan dapat dilihat di lampiran pada laporan ini.

4.4.3 Evaluasi

Evaluasi perhitungan *precision*, *recall*, *f-score*, dan *agreement* dilakukan secara otomatis dengan program yang dibuat dengan algoritma berikut.

Kode 4.4: Word Alignment Evaluation

```
def evaluate_bracket(list_giza, list_anotator):
    # evaluate per character
    numbers_anotator = len(list_anotator)
    numbers_giza = len(list_giza)
    match = 0
    if numbers_giza < numbers_anotator:
        # iterate through the numbers giza
        for n in list_giza:
            if n in list_anotator:
                match += 1
    else:
        # iterate through the numbers anotator
        for n in list_anotator:
            if n in list_giza:
                match += 1
    return (numbers_anotator, numbers_giza, match)

# given sentence which already been aligned from the first
# anotator (an1), second anotator(an2), and giza(giza)
matches, total_giza, total_anotator, total_giza_1,
total_anotator_1 = 0, 0, 0, 0, 0
```

```

precision, recall = [[], []], [[], []]

for each sentence in sentences:
    for each token in sentence:
        (numbers_annotator, numbers_giza, match) = evaluate_bracket(
            an1(token), giza(token))
        (numbers_annotator_1, numbers_giza_1, match_1) =
            evaluate_bracket(an2(token), giza(token))
        agreement = count_agreement(an1(token), an2(token))
        matches += match
        matches_1 += match_1
        total_giza += numbers_giza
        total_giza_1 += numbers_giza_1
        total_annotator += numbers_annotator
        total_annotator_1 += numbers_annotator_1
        agreement.append(agreement)
    precision[0].append(matches/total_giza)
    recall[0].append(matches/total_annotator)
    precision[1].append(matches_1/total_giza_1)
    recall[1].append(matches_1/total_annotator_1)

precision = average(precision[0])
recall = average(recall[0])
precision_1 = average(precision[1])
recall_1 = average(recall[1])
agreement = average(agreement)

```

Berdasarkan cara perhitungan evaluasi tersebut, terdapat dua buah *precision* dan *recall* yang mana masing-masing menunjukkan indikator penilaian untuk anotator pertama dan kedua.

4.5 Peningkatan Kualitas *Alignment*

Hasil dari *alignment* kata yang dilakukan Giza masih menghasilkan pasangan-pasangan kata yang tidak tepat. Untuk mengurangi jumlah pasangan kata yang salah tersebut, dilakukan *enhancement* terhadap hasil pasangan kata dari Giza. Terdapat dua macam peningkatan kualitas *alignment* yang digunakan pada penelitian ini, yaitu *crawling based* dan *bi-directional based*.

4.5.1 Pendekatan *Crawling*

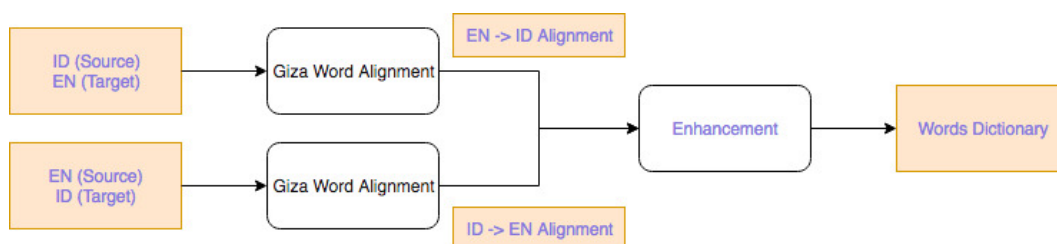
Konsep *crawling* pada penelitian ini mengacu pada kebutuhan untuk *filtering* hasil pasangan kata yang salah berdasarkan kamus Indonesia-Inggris. Karena keter-

batasan *resource digital* kamus tersebut, maka dibutuhkan pendekatan *crawling* dari *online dictionary* untuk mendapatkan pasangan kata yang benar. Salah satu *online dictionary* yang dapat diakses dan memiliki hasil terjemahan yang cukup baik adalah *website* sederet.com. *Crawling* dilakukan dengan memeriksa setiap pasangan bahasa Inggris dari kata Indonesia hasil Giza, apakah pasangan kata tersebut berada pada hasil penerjemahan yang sesuai. Ilustrasi dari proses ini dapat dimodelkan dalam contoh berikut:

1. Kata "halaman" memiliki pasangan bahasa Inggris hasil Giza yaitu "courtyard", "yard", "page", dan "lawn".
2. Gunakan *crawler* untuk menerima hasil terjemahan dari kata "halaman" dalam bahasa Inggris
3. *Crawler* mendapatkan hasil kata "page", dan "courtyard".
4. Berdasarkan kedua hasil tersebut, maka pasangan kata "halaman" yang dianggap benar adalah irisan dari kedua *set* tersebut yaitu "page" dan "courtyard"

4.5.2 Pendekatan *Bi-directional*

Metode lain yang digunakan untuk meningkatkan kualitas *alignment* yaitu dengan memanfaatkan *bi-directional alignment*. Proses yang dilakukan adalah melakukan validasi terhadap kata-kata yang berpasangan dari kedua korpus. Bagan dari *enhancement* pada *bi-directional* ini dapat dilihat pada gambar 4.2.



Gambar 4.2: Bi-directional *Enhancement*

Pertama, setiap kata dalam bahasa Indonesia dikumpulkan terlebih dahulu dengan setiap pasangan kata bahasa Inggrisnya (satu kata bisa memiliki lebih dari satu pasangan). Proses selanjutnya adalah melakukan pengumpulan yang serupa terhadap kata dalam bahasa Inggris dengan pasangan kata dalam bahasa Indonesianya. Berbagai kata dalam bahasa Indonesia beserta pasangannya disimpan

sebagai "kamus-1". Kata dalam bahasa Inggris, beserta pasangan kata Indonesianya disimpan sebagai "kamus-2". Proses validasi dilakukan dengan cara:

1. Untuk setiap kata di bahasa Indonesia dalam "kamus-1" semisal kata "kali".
2. Lakukan pengecekan terhadap setiap pasangan kata di bahasa Inggris pada "kamus-1" dari kata "kali", misalkan pasangan kata bahasa inggrisnya adalah "time", "river", dan "fire".
3. Jika pada "kamus-2" kata "time" dipasangkan dengan "kali", dan "waktu" maka kata "time" merupakan pasangan yang dianggap benar (karena kata "time" berpasangan dengan "kali"). Pada kasus kata "fire", bila pasangan bahasa Indonesianya adalah "api" dan "tungku", maka kata "fire" dianggap bukan pasangan yang tepat dengan "kali" karena tidak terdapat pasangan "fire -> kali".

Kode 4.5: Word Alignment Enhancement

```
dict_id = {}
dict_en = {}

# masukan setiap kosa kata bahasa Indonesia ke dalam dict_id
# sebagai key dan kumpulan pasangan kata bahasa inggrisnya
# sebagai value
# proses yang sama dilakukan untuk dict_en dengan kosa kata
# bahasa Inggris sebagai key dan kumpulan pasangan kata bahasa
# Indonesia sebagai value
# stop adalah list stopword yang didapat dari korpus nltk

# this section is for filtering which english word that has
# corresponding indo translation (bidirectional) from Giza
# output
for indo_word in dict_id.keys():
    for en_word in dict_id[indo_word].keys():
        if en_word != dict_en:
            # filtering so no same translation is entered, answer ->
            # answer, jawaban -> jawaban
            if en_word in dict_en and indo_word in dict_en[en_word] and
                en_word not in stop:
                if indo_word not in final_dictionary:
                    final_dictionary[indo_word] = { en_word: dict_en[
                        word_en][word_id] }
                else:
```

```

if en_word not in final_dictionary[indo_word]:
    final_dictionary[indo_word][en_word] = dict_en[
        word_en][word_id]

```

4.6 Sense Transferring

Hasil dari proses peningkatan kualitas *alignment* adalah sebuah "kamus" bahasa yang akan digunakan sebagai referensi untuk memindahkan makna kata dari bahasa Inggris ke kata yang benar pada bahasa Indonesia. Proses ini dilakukan dengan tahap-tahap sebagai berikut:

1. Iterasi untuk setiap kata dalam bahasa Indonesia pada kamus
2. Iterasi pada setiap pasangan kalimat
3. Periksa apakah "*pair*" kata bahasa Indonesia tersebut terdapat di dalam kamus
4. Jika "*pair*" tersebut benar berada dalam kamus, maka pindahkan makna kata dari *english word* yang bersesuaian.

Ilustrasi dari proses tersebut pada kata "halaman" adalah sebagai berikut:

1. Jika misalkan kata "halaman" pada kamus memiliki pasangan kata "page" dan "courtyard".
2. Pada sebuah kalimat "... halaman kedua ..." dimana "*pair*" pada kalimat tersebut diantaranya adalah

```

..<pair>second||kedua</pair>##<pair>page||halaman</pair>..

```

3. Pasangan kata yang didapat dari "halaman" dari *pair* tersebut adalah "page". Berdasarkan hasil tersebut kata "page" kemudian diperiksa pada kamus yang ada.
4. Karena kata "page" merupakan salah satu terjemahan untuk kata "halaman", maka pindahkan makna "page" dari *sense tagged corpus* kalimat tersebut ke kata "halaman" pada kalimat Indonesia dengan indeks yang sama.
5. Dalam proses pemindahan makna tersebut, akan dilakukan pemeriksaan apakah untuk kata yang sama, terdapat makna kata yang serupa dari hasil transfer kalimat lain.

6. Jika "kemiripan" makna kata yang akan dipindahkan melebihi threshold (0.5), maka akan digunakan *sense key* yang lama karena kedua *sense key* dianggap mempunyai makna yang sama. Penghitungan seberapa dekat "makna" dari kedua *sense key* tersebut dilakukan dengan bantuan *method path_similarity* dari *tool* NLTK dengan korpus wordnet.

4.7 Sistem WSD

Sistem WSD dibangun dengan menggunakan *supervised learning*. *Machine learning tool* yang digunakan untuk membangun sistem ini adalah Scikit (Pedregosa et al., 2011). Pada sistem ini terdapat tiga buah bagian utama yaitu pemilihan fitur serta *classifier*, ekstraksi fitur, dan evaluasi hasil *classifier*. Proses yang pertama kali dilakukan oleh sistem adalah membaca korpus dan memilih *target word*, mendapatkan *sense key* dari kata tersebut untuk dijadikan sebagai *class* dari klasifikasi.

4.7.1 Pemilihan Fitur dan Classifier

Terdapat tiga buah fitur pada penelitian ini yang terdiri dari:

1. Fitur *bag of words*
2. Fitur *POS tagging*
3. Vektor dari hasil *word embedding*

Classifier yang digunakan pada penelitian ini adalah SVM dengan *library* Python yaitu Scikit dengan parameter *default* berupa kernel linear dan $C=1$.

4.7.2 Ekstraksi Fitur

Fitur *bag of words* menggunakan pendekatan kemunculan kata-kata pada konteks kalimat sebagai fitur. Kata yang diambil untuk dijadikan fitur adalah dua buah kata di kiri dan di kanan dari *target word*. Pada penelitian ini, kata-kata yang merupakan bagian dari *stopwords* dalam bahasa Indonesia tidak dimasukkan sebagai fitur. Contoh dari fitur ini dapat dilihat pada kalimat dengan kata tujuan "bisa" berikut:

- Ani digigit ular dengan bisa yang berbahaya

Pada contoh kalimat tersebut, *bag of words* yang diambil adalah ["digigit", "ular", "berbahaya", NULL]

Setiap fitur *bag of words* dari setiap kalimat tersebut dikumpulkan untuk menjadi satu fitur besar yang mendeteksi kemunculan kata-kata tersebut pada setiap kalimat.

Proses yang dilakukan dalam pengambilan kata konteks untuk fitur *bag of words* dilakukan dengan tahap-tahap berikut.

Kode 4.6: Fitur Bag of Words

```

bag_of_words []

for each sentence
    split sentence by whitespace into words
    for each word
        if word == target word
            for x in [-2,-1,1,2]
                if word(x) exist and word(x) not in bag_of_words
                    add word(x) into bag_of_words

return bag_of_words

```

Fitur POS Tagging menggunakan *tool* dari Stanford bernama "A Part-Of-Speech Tagger" yang dilatih dengan menggunakan model untuk bahasa Indonesia. Proses *tagging* ini dilakukan sebelum memasuki sistem WSD terhadap keseluruhan kalimat dalam korpus identik yang berisi bahasa Indonesia saja. Terdapat *pre-processing* awal pada korpus bahasa Indonesia tersebut untuk menghilangkan beberapa tanda baca seperti titik, koma, tanda tanya, tanda seru, dan beberapa tanda baca lainnya. Setelah proses tersebut, diberikan tanda baca berupa titik pada akhir kalimat sebagai indikator akhir sebagai kebutuhan dari komabilitas *tool* (tidak semua kalimat pada korpus memiliki tanda baca akhir kalimat). Perintah yang dilakukan untuk melakukan *POS Tagging* tersebut adalah:

Kode 4.7: Stanford POS Tagger

```

java -mx300m -cp 'stanford-postagger.jar:lib/*' edu.stanford.nlp.
tagger.maxent.MaxentTagger -model <model_bahasa_indonesia> -
textFile <korpus_bahasa_indonesia>

```

Hasil dari proses tersebut merupakan korpus dengan setiap kata yang sudah memiliki POS Tag dengan format:

```

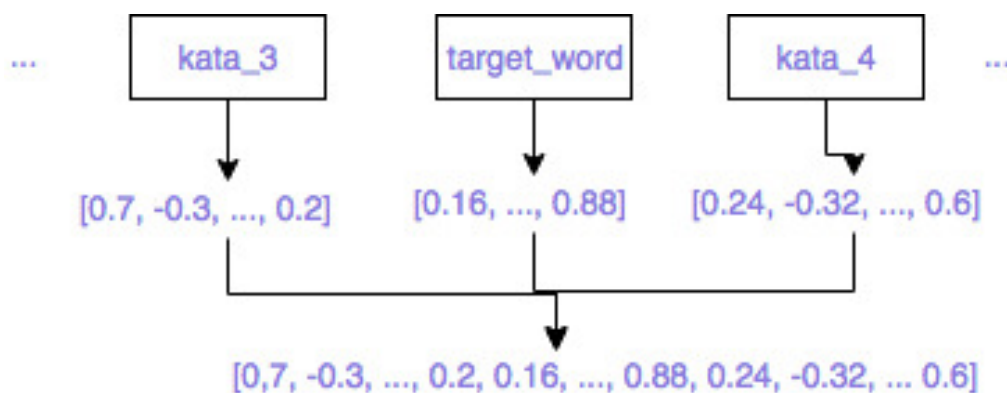
<kata_1>_<postag_1> <kata_2>_<postag_2> ... <kata_n>_<postag_n>
<kata_x>_<postag_x> <kata_y>_<postag_y> ... <kata_z>_<postag_z>
...

```

Kelas kata yang diambil adalah POS Tag dari *target word* kata sebelum, dan kata sesudahnya. Kelas kata dari *target word* diperhitungkan karena pada beberapa kasus kata polisemi dapat dibedakan maknanya berdasarkan POS Tag yang dimilikinya. POS Tag yang digunakan mengacu pada kelas-kelas yang ada pada POS Tag Penn Treebank seperti NN(Noun), NNP(Proper Noun), VB(verb), CC(Coordinating conjunction), dan lain-lain. Pembentukan kelas POS Tag untuk menjadi fitur dilakukan dengan proses yang serupa pada fitur *bag of words* dimana kombinasi kombinasi dari POS Tag yang mungkin direpresentasikan dalam *one hot representation*. Hal ini dapat diilustrasikan dengan proses berikut:

1. Simpan POS Tag dari indeks kata masing-masing (*target word*-1, *target word*, *target word*+1)
2. Untuk masing-masing indeks baik itu -1,0,dan 1
3. Kumpulkan POS Tag yang *distinct*, populasikan dalam *array*
4. *Array* ini nantinya akan merepresentasikan keberadaan POS Tag tertentu pada kata dengan indeks terkait tersebut

Fitur ketiga yang dicoba adalah *word embedding*. Model dari *word embedding* yang digunakan sudah dilatih dengan menggunakan korpus Wikipedia bahasa Indonesia. Vektor *word embedding* yang dijadikan fitur adalah vektor dari semua kata pada kalimat tersebut. Semua vektor nilai dari kata-kata tersebut kemudian dikonkatenasi menjadi satu buah vektor besar. Untuk menjamin panjang vektor yang sama untuk setiap kalimat, panjang vektor disesuaikan dengan kalimat terpanjang dari semua kalimat yang mengandung *target word* tersebut. Proses tersebut dapat diilustrasikan pada gambar berikut:



Gambar 4.3: Ilustrasi Fitur Word Embedding

4.7.3 Evaluasi Sistem

Target word untuk evaluasi dipilih secara manual yang memenuhi kriteria bahwa kata tersebut memiliki kata *translation* lebih dari satu dengan makna yang berbeda. Evaluasi dilakukan dengan *cross validation* menggunakan perhitungan F1-score dari hasil klasifikasi yang dilakukan *classifier* terhadap *target word*. *Cross validation* dilakukan dengan iterasi sebanyak tiga kali dengan perbandingan antara *training* dan *test set* sebesar 0,7:0,3.

Sebuah algoritma sederhana digunakan sebagai *baseline* untuk pembandingan dari sistem WSD yang dibangun. Baseline menggunakan pendekatan *most frequent sense* sebagai cara untuk menentukan makna terbaik dari suatu kata. Bila diberikan *training data* untuk kata "bisa" dengan makna "dapat melakukan" sebanyak 4 buah dan makna "racun ular" sebanyak 6 buah, maka algoritma baseline ini akan mengklasifikasikan semua kata "bisa" menjadi "racun ular".

BAB 5

HASIL DAN ANALISIS

Bab ini menjelaskan mengenai hasil yang didapatkan dari eksperimen, serta evaluasi dan analisis terkait hasil tersebut.

5.1 Korpus Identik

Korpus identik berisi pasangan-pasangan kalimat Indonesia-Inggris sebanyak 88.919 kalimat untuk masing-masing bahasa.

5.2 Pembuatan *Sense Tagged Corpus* Bahasa Inggris

Tabel 5.1 menunjukkan jumlah token (kata) pada korpus bahasa Inggris dan yang diberikan *tag sense* oleh IMS

Tabel 5.1: Jumlah *instance* korpus bahasa Inggris

No	Tipe	Jumlah
1	Token (kata)	1.801.484
2	Kata yang diberikan <i>tag</i> oleh IMS	1.024.797

Berdasarkan proses pembuatan dan hasil dari *sense tagged corpus* tersebut, dapat dilihat bahwa tidak semua kata diberikan *sense* oleh IMS. Kata-kata sapaan seperti "I", "you", dan kata *articles* yaitu "a", "the", "an". Selain itu, kata yang tidak terdapat pada model juga tidak diberi *tag* (dilewati) oleh IMS. Tingkat kebenaran dari *sense tag* yang diberikan bergantung dari model yang digunakan pada penelitian. Terdapat banyak kasus-kasus dimana pemberian *sense* yang dilakukan adalah benar seperti misalnya pada kata "visitor" yang diberikan *tag* dengan *sense key* 1:18:00::, yang mana berdasarkan wordnet Princeton "visitor%1:18:00::" memiliki arti sebagai "someone who visits". Contoh lain dari kata yang diberikan *tag* dengan benar adalah "company" pada konteks potongan kalimat "Plantation company PT ...". Kata "company" tersebut diberikan tag "company%1:14:01::" yang berdasarkan wordnet Princeton memiliki makna "an institution created to conduct business". Namun demikian, terjadinya kesalahan pemberian *tag* pada kata terjadi pada kasus-kasus seperti:

1. Sebuah entitas diberikan *tag* dimana entitas tersebut dianggap kata biasa. Contohnya adalah kata "Scotland Yard" dimana "Yard" pada kata tersebut diberikan *tag* yang diartikan sebagai "*a unit of length equal to 3 feet*". Hal ini menunjukkan bahwa *tool* belum dapat membedakan antara entitas yang memang tidak perlu diberikan *tag* dan kata biasa (walaupun kata tersebut sudah memiliki huruf kapital).
2. Kesalahan *tag* dikarenakan *training data* yang digunakan oleh model. Pada potongan kalimat "... *FASB rule will cover such financial instruments as interest rate swaps financial ...*", kata "*interest*" diberikan tag dengan makna "*a sense of concern with and curiosity about someone or something*". Berdasarkan konteks kalimat tersebut, dapat diketahui bahwa makna yang seharusnya didapat untuk kata "*interest*" diatas ialah "bunga bank". Hal ini sepertinya terjadi karena data yang digunakan untuk *training* model IMS memiliki ketidakseimbangan data untuk model kata "*interest*" sehingga *tag* yang diberikan lebih cenderung kepada "ketertarikan".
3. Pemberian *tag* pada *multi word* token seperti "*make up*" masih diberikan pada setiap kata. Berdasarkan percobaan untuk *tagging* pada kata tersebut, kata "*make*" dan "*up*" masing-masing diberikan tag yang berbeda. Hal ini terjadi karena IMS mengolah kata demi kata dengan proses tokenisasi *by default* menggunakan spasi. Setelah dilakukan pemeriksaan pada kata-kata yang terdapat pada model, kata *make up* ternyata disimpan sebagai "*make_up*". Berdasarkan pemeriksaan tersebut, diperlukan adanya *pre-processing* terlebih dahulu untuk mengganti *separator* kata multiword yang umumnya menggunakan spasi dengan "_" agar IMS dapat memberikan *tag multi word* tersebut dengan benar. Selain *pre-processing*, IMS juga dapat melakukan *tagging* dengan input dalam format XML. Bentuk kata-kata dan kalimat dalam format XML tersebut biasanya memiliki multi word yang sudah dijadikan satu token sehingga mempermudah penyelesaian masalah tersebut.

5.3 Evaluasi Word Alignment

Hasil dari proses *word alignment* yang dilakukan Giza dibandingkan dengan hasil *alignment* yang dibuat oleh dua orang anotator. Jumlah yang akan dibandingkan adalah 200 buah pasangan data yang didapat dengan *random sampling*. Indikator performa dari perbandingan tersebut adalah nilai dari *precision* dan *recall*.

5.4 *Sense Transferring*

Proses *transfer* makna kata dari bahasa Inggris ke bahasa Indonesia yang dilakukan sangat bergantung dari hasil *alignment* kata pada proses sebelumnya. Untuk sebagian besar kata yang memiliki pasangan kata yang benar, proses *transfer* dapat menghasilkan makna yang benar juga. Hal tersebut didukung jika *sense tagged word* pada korpus bahasa Inggris juga benar). Terdapat beberapa kata yang dipilih sebagai *sampling* untuk mengevaluasi hasil *sense transferring*. Kelompok ini dibagi menjadi:

1. Jumlah Kelas
 - (a) 3-5 kelas kata
 - (b) lebih dari 5 kelas kata
2. sebaran jumlah *instance* dalam kelasnya
 - (a) *balance*
 - (b) *imbalance*
3. Bentuk morfologi dari kata tersebut
 - (a) Lemma (kata dasar)
 - (b) Berimbuhan baik itu infleksional ataupun *derivative*

Pada jumlah kelas sebanyak 3-5 kelas kata (*sense key*), *target word* yang diambil adalah "memecahkan". Kata tersebut memiliki 4 buah kelas total dengan *sense key* yang didapat yaitu 'solve%2:31:00::', 'resolve%2:31:01::', 'break%2:30:03::', dan 'split%2:38:00::'. Kata "menolak" mewakili kelas kata sebanyak 10 buah yang diantaranya mengandung kelas 'refuse%2:32:00::', 'reject%2:40:00::', 'decline%2:32:00::', dan beberapa kelas lainnya. Tabel 5.2 menunjukkan contoh beberapa kata tersebut dalam beberapa konteks kalimat yang bersesuaian.

Tabel 5.2: Evaluasi *Sense Transferring* Berdasarkan Jumlah Kelas

<i>Sense Key</i>	Makna	Kalimat
solve%2:31:00::	<i>find the solution to (a problem or question) or understand the meaning of</i>	salah satu cara untuk memecahkan persoalan yang pelik...
resolve%2:31:01::	<i>bring to an end / settle conclusively</i>	evolusionis masih belum bisa memecahkan permasalahan darwin...
break%2:30:03::	<i>terminate</i>	...base mereka memecahkan rekor untuk...
split%2:38:00::	<i>go one's own way; move apart;</i>	senat mereka memecahkan perbedaan antara skenario 1 dan 3 dengan
decline%2:32:00::	<i>show unwillingness towards</i>	wells rich menolak untuk berkomentar...
refuse%2:32:00::	<i>show unwillingness towards</i>	kelompok pemberontak yang menolak menandatangani perjanjian ...
reject%2:40:00::	<i>refuse to accept</i>	..dua pekan lalu menolak tawaran pemerintah...

Makna kata *split* yang diberikan hanya berjumlah satu buah dari keseluruhan korpus, hal ini disebabkan karena kata bahasa Inggris yang digunakan pada kalimat bahasa Inggrisnya menggunakan kata *split*. Berdasarkan *sampling* yang dilakukan, jumlah kelas kata yang ada bergantung pada sebanyak apa sebuah kata di bahasa Indonesia dipasangkan dengan kata bahasa Inggris yang berbeda dan memiliki makna pada *sense tagged english corpus*. Jumlah kelas ini dapat bergantung pada seberapa akurasi *alignment* kata yang dilakukan pada proses sebelumnya.

Pada sebaran jumlah *instance* di dalam kelas-kelasnya, kata "kehadiran" mewakili data yang jumlahnya relatif tidak seimbang. Dimana jumlah kelas *attendance* (19 buah) dan *presence* (64 buah). Kedua *sense* tersebut memiliki

makna yang kurang lebih menyatakan sebuah *state* dimana seseorang datang/hadir. Perbandingan jumlah *instance* yang lebih *balance* dari kata "kehadiran" salah satunya adalah kata "rumahnya". *sense key*. Tabel 5.3 menunjukkan makna kata yang dipindahkan berdasarkan *sampling* berdasarkan sebaran *instance* dalam kelas.

Tabel 5.3: Evaluasi *Sense Transferring* Berdasarkan Sebaran Kelas

Kata (Sense key)	Makna	Kalimat	Jumlah
Kehadiran (attendance%1:04:00::)	<i>the act of being present (at a meeting or event etc.)</i>	...tingkat kehadiran guru di sekolah...	19
Kehadiran (presence%1:09:00::)	<i>the impression that something is present</i>	...berkurangnya kehadiran pria dewasa...	64
rumahnya (house%1:14:02::)	<i>an official assembly having legislative powers</i>	...kebakaran yang melanda rumahnya ...	32
rumahnya (home%1:06:00::)	<i>Housing that someone is living in</i>	...ia pulang ke rumahnya pada sabtu...	19

Kedua makna pada kata "kehadiran" memiliki makna yang relatif dekat dan masuk dalam konteks kalimat kata tersebut muncul. Namun demikian, sense key "house%1:14:02::" tersebut memiliki makna yang salah, dimana sense key yang lebih tepat semestinya adalah house%1:06:01:: dengan makna "*a building in which something is sheltered or located*". Kesalahan makna kata yang dipindahkan tersebut disebabkan karena kata *house* pada korpus inggris diberikan tag 'house%1:06:01::', hal ini sepertinya terjadi karena data yang digunakan untuk training model tersebut lebih banyak mengandung kata 'house' dengan makna tersebut.

Kelompok *sampling* lain adalah makna yang akan dilihat pada kata dengan bentuk morfologi yang berbeda. Kata "makan", "makanan", dan "memakan" merupakan kata yang mewakili kasus bentuk morfologi dalam bentuk lemma maupun berimbuhan. Pada kata "makan", *sense key* yang diterima dari hasil *transfer* adalah eat%2:34:00:: yang memiliki makna "*take in solid food*". Kata "makanan" pada kalimat-kalimat yang ada diberikan *sense key* food%1:03:00:: yang diartikan sebagai "*any substance that can be metabolized by an animal*".

to give energy and build tissue". Kata "memakan" sendiri memiliki beberapa *sense key* seperti *consume%2:34:02::*, *eat%2:34:00::*, dan *feed%1:13:00::*. Dari *sense key* yang didapat tersebut, *consume%2:34:02::* ("spend extravagantly") bukan merupakan *sense* yang tepat (seharusnya memiliki makna mengonsumsi makanan), dan *feed%1:13:00::* ("*food for domestic livestock*") yang semestinya adalah "memberikan makanan". Makna kata yang tidak tepat pada hasil-hasil tersebut merupakan kesalahan dari baik itu variasi *alignment* suatu kata yang dipasangkan dengan kata lainnya, dan juga model IMS yang kurang pas dengan domain data pada penelitian ini.

5.5 Sistem WSD

Untuk melihat seberapa baik performa sistem WSD dengan menggunakan *sense tagged corpus* hasil dari penelitian, terdapat kata-kata yang dipilih secara manual sebagai sampling dari *target word* yang akan dievaluasi berdasarkan nilai F-score dari hasil rata-rata *cross validation*. Kata yang dipilih merupakan *instance* yang memiliki pasangan kata lebih dari satu dalam bahasa Inggris dan mempunyai makna yang berbeda dari hasil *sense transferring*. *Target word* yang dipilih tersebut memiliki kriteria bahwa pasangan bahasa Inggris kata tersebut lebih dari satu ataupun juga pasangan bahasa Inggrisnya memiliki makna yang tidak dekat (ambigu). Fitur yang dilakukan percobaan pada penelitian ini adalah F1(*bag of words*), F2 (*word embedding*), F3 (*pos-tag*), F4(*pos tagging* dan *bag of words*). Hasil evaluasi dapat dilihat pada tabel akurasi 5.4 berikut.

Tabel 5.4: Evaluasi Sistem WSD

<i>Target Word</i>	Baseline	F1	F2	F3	F4
meninggalkan	0.89	0.89	0.88	0.89	0.88
memecahkan	0.50	0.50	0.54	0.50	0.46
menolak	0.36	0.52	0.49	0.52	0.60
obat	0.49	0.70	0.56	0.61	0.78
linkungan	0.54	0.54	0.43	0.67	0.66
halaman	0.93	0.93	0.93	0.91	0.93
kehadiran	0.67	0.91	0.77	0.73	0.91
hati	0.72	0.84	0.77	0.73	0.84
coklat	0.11	0.39	0.22	0.17	0.33
acara	0.58	0.60	0.53	0.45	0.55
berat	0.53	0.61	0.52	0.68	0.70
jalan	0.66	0.75	0.73	0.68	0.74

Berdasarkan hasil *scoring* evaluasi tersebut, kata "coklat" mempunyai akurasi terendah dengan semua penilaian akurasi dibawah 50%. Makna kata yang diberikan pada kata "coklat" tersebut adalah chocolate%1:13:02:: sebanyak 5 buah, cocoa%1:13:02:: 5 buah, cacao%1:20:00:: 5 buah, brown%5:00:00:chromatic:00 4 buah. Dapat dilihat bahwa dari jumlah masing-masing *sense key* (kelas kata) tersebut seimbang. Namun demikian, jumlah *instance* masing-masing kelas yang relatif sedikit kemungkinan menyebabkan akurasi yang dihasilkan di bawah 50%. Pada kata "halaman" persebaran *instance* dalam kelasnya adalah page%1:10:00:: sebanyak 41 buah, yard%1:23:00:: 3 buah, dan courtyard%1:06:00:: 3 buah. Jumlah data yang tidak *balance* pada kelas-kelas tersebut bisa jadi merupakan penyebab tingginya akurasi yang ada bahkan pada tingkat *baseline*. Berbeda dengan kata "kehadiran" yang memiliki persebaran makna kata attendance%1:04:00:: sebanyak 19 buah, presence%1:09:00:: 64 buah, dan existence%1:26:00:: 1 buah. Dengan perbedaan antara kelas makna 'attendance' dan 'presence' yang tidak sejauh pada kasus kata 'halaman', akurasi baseline yang dihasilkan lebih rendah dari baseline 'halaman'. Namun demikian, akurasi dari sistem WSD yang dihasilkan untuk kata 'kehadiran' tersebut diatas baseline untuk semua fitur yang dicoba.

BAB 6

PENUTUP

6.1 Kesimpulan

Data dan *resource* berupa *sense tagged corpus* yang terbatas pada bahasa Indonesia merupakan penghambat dari penelitian WSD di bahasa Indonesia. Untuk mengatasi masalah tersebut, salah satu pendekatan WSI berupa *cross language* dapat dimanfaatkan untuk *transferring knowledge* dari salah satu bahasa dengan data yang mumpuni yaitu bahasa Inggris. Konsep pendekatan ini dapat menghasilkan *sense tagged corpus* bahasa Indonesia dengan memindahkan makna kata dari korpus bahasa Inggris ke kata-kata yang menjadi pasangannya (*translation* dari kata tersebut) di dalam bahasa Indonesia. Pendekatan yang dilakukan dengan proses-proses dari pembuatan *sense tagged corpus* bahasa Inggris sampai dengan percobaan sistem WSD ini dapat menghasilkan sebuah *sense tagged corpus* makna kata dalam bahasa Indonesia yang mempunyai hasil dengan kualitas cukup baik walaupun masih dapat ditingkatkan lagi. Sistem WSD yang dibangun sendiri telah dibuktikan dapat mengungguli sistem baseline untuk melakukan penentuan makna kata.

6.2 Saran

Setelah melakukan percobaan dan melakukan analisis hasil dari penelitian ini, terdapat beberapa saran untuk penelitian selanjutnya diantaranya sebagai berikut.

1. Penggunaan kualitas dari korpus paralel dapat mempengaruhi seberapa baik hasil *sense transferring* yang dilakukan. Pada korpus identik yang penelitian ini gunakan, terdapat beberapa kalimat yang terpotong maupun *comparable* (tidak sepenuhnya paralel), sehingga rawan menimbulkan kesalahan *alignment*
2. Fitur yang digunakan dalam sistem WSD bahasa Indonesia masih dapat ditambahkan dengan berbagai fitur lainnya yang dapat menunjang akurasi sistem.
3. *Classifier* pada sistem WSD juga dapat diubah-ubah untuk mencari hasil optimal pada penelitian selanjutnya.

4. Proses *word alignment* sebaiknya dilakukan dengan *tool* SMT yang lebih baru, salah satunya adalah *berkeley aligner*.

DAFTAR REFERENSI

- Denkowski, M. (2009). A survey of techniques for unsupervised word sense induction. *Language & Statistics II Literature Review*, pages 1–18.
- Fradkin, D. dan Muchnik, I. (2006). Support vector machines for classification. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 70:13–20.
- Gale, W. A., Church, K. W., dan Yarowsky, D. (1992). One sense per discourse. In *Proceedings of the workshop on Speech and Natural Language*, pages 233–237. Association for Computational Linguistics.
- Mihalcea, R. dan Pedersen, T. (2003). An evaluation exercise for word alignment. In *Proceedings of the HLT-NAACL 2003 Workshop on Building and using parallel texts: data driven machine translation and beyond-Volume 3*, pages 1–10. Association for Computational Linguistics.
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., dan Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.
- Nasiruddin, M. (2013). A state of the art of word sense induction: A way towards word sense disambiguation for under-resourced languages. *arXiv preprint arXiv:1310.1425*.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., dan Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- Rudnick, A. (2011). Towards cross-language word sense disambiguation for quechua. In *RANLP Student Research Workshop*, pages 133–138.
- Zhong, Z. dan Ng, H. T. (2010). It makes sense: A wide-coverage word sense disambiguation system for free text. In *Proceedings of the ACL 2010 System Demonstrations*, pages 78–83. Association for Computational Linguistics.