UNIVERSITAS INDONESIA

OPEN DOMAIN INFORMATION EXTRACTION OTOMATIS DARI TEKS BAHASA INDONESIA

TESIS

YOHANES GULTOM
1506706345

FAKULTAS ILMU KOMPUTER
PROGRAM STUDI MAGISTER ILMU KOMPUTER
DEPOK
JUNI 2017

UNIVERSITAS INDONESIA

OPEN DOMAIN INFORMATION EXTRACTION OTOMATIS DARI TEKS BAHASA INDONESIA

TESIS
Diajukan sebagai salah satu syarat untuk memperoleh gelar
Magister Ilmu Komputer

YOHANES GULTOM
1506706345

FAKULTAS ILMU KOMPUTER
PROGRAM STUDI MAGISTER ILMU KOMPUTER
DEPOK
JUNI 2017

ABSTRAK

Nama	:	Yohanes Gultom
Program Studi	:	Magister Ilmu Komputer
Judul	:	Open Domain Information Extraction Otomatis dari Teks Bahasa Indonesia

Banyaknya jumlah dokumen digital yang tersedia saat ini sudah melebihi kapasitas manusia untuk memprosesnya secara manual. Hal ini mendorong munculnya kebutuhan akan metode ekstrasi informasi (*information extraction*) otomatis dari teks atau dokumen digital dari berbagai domain (*open domain*). Sayangnya, sistem *open domain information extraction* (*open IE*) yang ada saat ini hanya berlaku untuk bahasa tertentu saja. Selain itu belum ada sistem *open IE* untuk bahasa Indonesia yang dipublikasikan. Pada penelitian ini Penulismemperkenalkan sebuah sistem untuk mengekstraksi relasi antar entitas dari teks bahasa Indonesia dari berbagai domain. Sistem ini menggunakan pembangkit kandidat *triple* (*triple candidates generator*) dan pengembang token (*token expander*) berbasis aturan serta pemilih *triple* berbasis *machine learning*. Setelah melakukan *cross-validation* terhadap empat kandidat model: *logistic regression*, SVM, MLP dan *Random Forest*, Penulismenemukan bahwa *Random Forest* adalah *classifier* yang terbaik untuk dijadikan *triple selector* denan skor F1 0.58 (*precision* 0.62 dan *recall* 0.58). Penyebab utama skor yang masih rendah ini adalah aturan pembangkitan kandidat yang masih sederhana dan cakupan pola *dataset* yang masih rendah.

Kata Kunci:
information extraction, open domain, natural language processing, bahasa Indonesia

ABSTRACT

Name	:	Yohanes Gultom
Program	:	Magister Ilmu Komputer
Title	:	Automatic Open Domain Information Extraction from Indonesian Text

The vast amount of digital documents, that have surpassed human processing capability, calls for an automatic information extraction method from any text document regardless of their domain. Unfortunately, open domain information extraction (open IE) systems are language-specific and there is no published system for Indonesian language. This paper introduces a system to extract entity relations from Indonesian text in triple format using rule-based candidates generator, token expander and machine-learning-based triple selector. We cross-validate four candidates: logistic regression, SVM, MLP, Random Forest using our dataset to discover that Random Forest is the best classifier for the triple selector achieving 0.58 F1 score (0.62 precision and 0.58 recall). The low score is largely due to the simplistic candidate generation rules and the coverage of dataset.

Keywords:
information extraction, open domain, natural language processing, Indonesian

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB 1
PENDAHULUAN

@todo
tambahkan kata-kata pengantar bab 1 disini

1.1 Latar Belakang

Ketersediaan dokumen bahasa Indonesia dalam jumlah banyak dengan domain yang beragam menuntut proses ekstraksi yang otomatis & generik

Kebutuhan akan representasi dokumen teks bahasa Indonesia yang mencerminkan informasi lintas domain di dalamnya. Contoh aplikasinya adalah deteksi plagiarisme, document retrieval .dsb

Pembangunan knowledge base bahasa Indonesia dengan ontologi yang matang dapat dimulai dari ekstraksi informasi lintas domain

Tabel 1.1: Perbandingan antara *information extraction* tradisional, *open domain extraction* dan *knowledge extraction*

	IE	**Open IE**	**KE**
Domain	Closed	Open	Open
Format	Depends on domain	Triples	RDF Triples
Ontology	Not available	Optional	Mandatory

1.2 Permasalahan

Pada bagian ini akan dijelaskan mengenai definisi permasalahan yang Penulis hadapi dan ingin diselesaikan serta asumsi dan batasan yang digunakan dalam menyelesaikannya.

1.2.1 Definisi Permasalahan

@todo
Tuliskan permasalahan yang ingin diselesaikan. Bisa juga berbentuk pertanyaan

1.2.2 Batasan Permasalahan

1. Ekstraksi dilakukan pada teks bahasa Indonesia yang telah melewati praproses untuk memisahkan tiap kalimat menjadi satu baris
2. Hanya mengkstraksi triples yang eksplisit secara struktur dependency relation. Contoh: Universitas Indonesia berada di Depok, Jawa Barat, Indonesia (Universitas Indonesia-terletak di-Depok)
3. Dataset yang dikembangkan dalam penelitian ini hanya dataset untuk triple selector sedangkan dataset lainnya menggunakan yang sudah tersedia

1.3 Tujuan dan Manfaat

Penelitian ini bertujuan untuk:

1. Menghasilkan kakas open domain information extraction otomatis untuk teks bahasa Indonesia
2. Mendefinisikan model open domain information extraction untuk teks bahasa Indonesia
3. Menentukan aturan-aturan (rules) yang dibutuhkan model
4. Menentukan fitur-fitur yang dapat digunakan pada model
5. Menentukan hyperparameter yang optimal untuk model

Manfaat yang diharapkan dari penelitian ini adalah:

1. Menyediakan kakas yang menghasilkan representasi dan/atau informasi yang dapat dimanfaatkan untuk aplikasi lain
2. Memberikan informasi mengenai model open domain information extraction untuk bahasa Indonesia
3. Mendorong pengembangan sumber daya (language resources) bahasa Indonesia

1.4 Metodologi Penelitian

@todo

Tuliskan metodologi penelitian yang digunakan.

1.5 Sistematika Penulisan

Sistematika penulisan laporan adalah sebagai berikut:

- Bab 1 PENDAHULUAN
- Bab 2 LANDASAN TEORI
- Bab 3 METODE PENELITIAN
- Bab 4 HASIL DAN ANALISIS
- Bab 5 PENUTUP

@todo

Tambahkan penjelasan singkat mengenai isi masing-masing bab.

BAB 2
LANDASAN TEORI

@todo

tambahkan kata-kata pengantar bab 2 disini

2.1 LaTeX Secara Singkat

Definisi dari LaTeX (**?**) adalah:

> LaTeX is a family of programs designed to produce publication-quality typeset documents. It is particularly strong when working with mathematical symbols.
> The history of LaTeX begins with a program called TEX. In 1978, a computer scientist by the name of Donald Knuth grew frustrated with the mistakes that his publishers made in typesetting his work. He decided to create a typesetting program that everyone could easily use to typeset documents, particularly those that include formulae, and made it freely available. The result is TEX. Knuth's product is an immensely powerful program, but one that does focus very much on small details. A mathematician and computer scientist by the name of Leslie Lamport wrote a variant of TEX called LaTeX that focuses on document structure rather than such details.

Contoh sitasi lainnya menggunakan `\citep` adalah saat kita mau mensitasi pekerjaan tentang *machine learning* (**?**) dan *dynamic programming* (**?**).

Dokumen LaTeX sangat mudah, seperti halnya membuat dokumen teks biasa. Ada beberapa perintah yang diawali dengan tanda '\'. Seperti perintah \\ yang digunakan untuk memberi baris baru. Perintah tersebut juga sama dengan perintah \newline. Pada bagian ini akan sedikit dijelaskan cara manipulasi teks dan perintah-perintah LaTeX yang mungkin akan sering digunakan. Jika ingin belajar hal-hal dasar mengenai LaTeX, silahkan kunjungi:

- http://frodo.elon.edu/tutorial/tutorial/, atau

- http://www.maths.tcd.ie/~dwilkins/LaTeXPrimer/

2.2 LaTeX Kompiler dan IDE

Agar dapat menggunakan LaTeX (pada konteks hanya sebagai pengguna), Anda tidak perlu banyak tahu mengenai hal-hal didalamnya. Seperti halnya pembuatan dokumen secara visual (contohnya Open Office (OO) Writer), Anda dapat menggunakan LaTeX dengan cara yang sama. Orang-orang yang menggunakan LaTeX relatif lebih teliti dan terstruktur mengenai cara penulisan yang dia gunakan, LaTeX memaksa Anda untuk seperti itu.

Kembali pada bahasan utama, untuk mencoba LaTeX Anda cukup mendownload kompiler dan IDE. Saya menyarankan menggunakan Texlive dan Texmaker. Texlive dapat didownload dari http://www.tug.org/texlive/. Sedangkan Texmaker dapat didownload dari http://www.xm1math.net/texmaker/. Untuk pertama kali, coba buka berkas thesis.tex dalam template yang Anda miliki pada Texmaker. Dokumen ini adalah dokumen utama. Tekan F6 (PDFLaTeX) dan Texmaker akan mengkompilasi berkas tersebut menjadi berkas PDF. Jika tidak bisa, pastikan Anda sudah menginstall Texlive. Buka berkas tersebut dengan menekan F7. Hasilnya adalah sebuah dokumen yang sama seperti dokumen yang Anda baca saat ini.

2.3 Bold, Italic, dan Underline

Hal pertama yang mungkin ditanyakan adalah bagaimana membuat huruf tercetak tebal, miring, atau memiliki garis bawah. Pada Texmaker, Anda bisa melakukan hal ini seperti halnya saat mengubah dokumen dengan OO Writer. Namun jika tetap masih tertarik dengan cara lain, ini dia:

- **Bold**
  Gunakan perintah \textbf{} atau \bo{}.
- *Italic*
  Gunakan perintah \textit{} atau \f{}.
- Underline
  Gunakan perintah \underline{}.
- $\overline{Overline}$
  Gunakan perintah \overline.

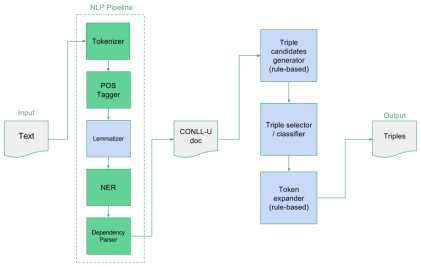
- *superscript*
  Gunakan perintah \^{}.

- *subscript*
  Gunakan perintah _{}.

Perintah \f dan \bo hanya dapat digunakan jika package uithesis digunakan.

2.4 Memasukan Gambar

Setiap gambar dapat diberikan caption dan diberikan label. Label dapat digunakan untuk menunjuk gambar tertentu. Jika posisi gambar berubah, maka nomor gambar juga akan diubah secara otomatis. Begitu juga dengan seluruh referensi yang menunjuk pada gambar tersebut. Contoh sederhana adalah Gambar 2.1. Silahkan lihat code LaTeX dengan nama bab2.tex untuk melihat kode lengkapnya. Harap diingat bahwa caption untuk gambar selalu terletak dibawah gambar.



Gambar 2.1: *Creative Common License 1.0 Generic.*

2.5 Membuat Tabel

Seperti pada gambar, tabel juga dapat diberi label dan caption. Caption pada tabel terletak pada bagian atas tabel. Contoh tabel sederhana dapat dilihat pada Tabel 2.1.

Tabel 2.1: Contoh Tabel

	kol 1	kol 2
baris 1	1	2
baris 2	3	4
baris 3	5	6
jumlah	9	12

Ada jenis tabel lain yang dapat dibuat dengan LaTeX berikut beberapa diantaranya. Contoh-contoh ini bersumber dari http://en.wikibooks.org/wiki/LaTeX/Tables

Tabel 2.2: An Example of Rows Spanning Multiple Columns

No	Name	Week 1			Week 2		
		A	B	C	A	B	C
1	Lala	1	2	3	4	5	6
2	Lili	1	2	3	4	5	6
3	Lulu	1	2	3	4	5	6

Tabel 2.3: An Example of Columns Spanning Multiple Rows

Percobaan	Iterasi	Waktu
Pertama	1	0.1 sec
Kedua	1	0.1 sec
	3	0.15 sec
Ketiga	1	0.09 sec
	2	0.16 sec
	3	0.21 sec

Tabel 2.4: An Example of Spanning in Both Directions Simultaneously

		Title			
		A	B	C	D
Type	X	1	2	3	4
	Y	0.5	1.0	1.5	2.0
Resource	I	10	20	30	40
	J	5	10	15	20

BAB 3
METODE PENELITIAN

@todo
tambahkan kata-kata pengantar bab 1 disini

3.1 Satu Persamaan

$$\frac{y-y_1}{y_2-y_1}=\frac{x-x_1}{x_2-x_1} \tag{3.1}$$

Persamaan 3.1 diatas adalah persamaan garis. Persamaan 3.1 dan 3.2 sama-sama dibuat dengan perintah \align. Perintah ini juga dapat digunakan untuk menulis lebih dari satu persamaan.

$$\underbrace{|\overline{ab}|}_{\text{pada bola } |\overline{ab}|=r} = \sqrt[2]{(x_b-x_a)^2+(y_b-y_a)^2+||(z_b-z_a)^2} \tag{3.2}$$

3.2 Lebih dari Satu Persamaan

$$|\overline{a}*\overline{b}| = |\overline{a}||\overline{b}|\sin\theta \tag{3.3}$$

$$\begin{aligned}\overline{a}*\overline{b} &= \begin{vmatrix} \hat{i} & x_1 & x_2 \\ \hat{j} & y_1 & y_2 \\ \hat{k} & z_1 & z_2 \end{vmatrix} \\ &= \hat{i}\begin{vmatrix} y_1 & y_2 \\ z_1 & z_2 \end{vmatrix} + \hat{j}\begin{vmatrix} z_1 & z_2 \\ x_1 & x_2 \end{vmatrix} + \hat{k}\begin{vmatrix} x_1 & x_2 \\ y_1 & y_2 \end{vmatrix}\end{aligned}$$

Pada Persamaan 3.3 dapat dilihat beberapa baris menjadi satu bagian dari Persamaan 3.3. Sedangkan dibawah ini dapat dilihat bahwa dengan cara yang sama, Persamaan 3.4, 3.5, dan 3.6 memiliki nomor persamaannya masing-masing.

$$\int_a^b f(x)\,dx + \int_b^c f(x)\,dx = \int_a^c f(x)\,dx \tag{3.4}$$

$$\lim_{x\to\infty} \frac{f(x)}{g(x)} = 0 \qquad \text{jika pangkat } f(x) < \text{pangkat } g(x) \tag{3.5}$$

$$a^{m^{a^n \log b}} = b^{\frac{m}{n}} \tag{3.6}$$

BAB 4
HASIL DAN ANALISIS

@todo
tambahkan kata-kata pengantar bab 1 disini

4.1 thesis.tex

Berkas ini berisi seluruh berkas Latex yang dibaca, jadi bisa dikatakan sebagai berkas utama. Dari berkas ini kita dapat mengatur bab apa saja yang ingin kita tampilkan dalam dokumen.

4.2 laporan_setting.tex

Berkas ini berguna untuk mempermudah pembuatan beberapa template standar. Anda diminta untuk menuliskan judul laporan, nama, npm, dan hal-hal lain yang dibutuhkan untuk pembuatan template.

4.3 istilah.tex

Berkas istilah digunakan untuk mencatat istilah-istilah yang digunakan. Fungsinya hanya untuk memudahkan penulisan. Pada beberapa kasus, ada kata-kata yang harus selalu muncul dengan tercetak miring atau tercetak tebal. Dengan menjadikan kata-kata tersebut sebagai sebuah perintah LaTeX tentu akan mempercepat dan mempermudah pengerjaan laporan.

4.4 hype.indonesia.tex

Berkas ini berisi cara pemenggalan beberapa kata dalam bahasa Indonesia. LaTeX memiliki algoritma untuk memenggal kata-kata sendiri, namun untuk beberapa kasus algoritma ini memenggal dengan cara yang salah. Untuk memperbaiki pemenggalan yang salah inilah cara pemenggalan yang benar ditulis dalam berkas hype.indonesia.tex.

4.5 pustaka.tex

Berkas pustaka.tex berisi seluruh daftar referensi yang digunakan dalam laporan. Anda bisa membuat model daftar referensi lain dengan menggunakan bibtex. Untuk mempelajari bibtex lebih lanjut, silahkan buka http://www.bibtex.org/Format. Untuk merujuk pada salah satu referensi yang ada, gunakan perintah \cite, e.g. \cite{latex.intro} yang akan akan memunculkan **?**

4.6 bab[1 - 6].tex

Berkas ini berisi isi laporan yang Anda tulis. Setiap nama berkas e.g. bab1.tex merepresentasikan bab dimana tulisan tersebut akan muncul. Sebagai contoh, kode dimana tulisan ini dibaut berada dalam berkas dengan nama bab4.tex. Ada enam buah berkas yang telah disiapkan untuk mengakomodir enam bab dari laporan Anda, diluar bab kesimpulan dan saran. Jika Anda tidak membutuhkan sebanyak itu, silahkan hapus kode dalam berkas thesis.tex yang memasukan berkas LaTeX yang tidak dibutuhkan; contohnya perintah \include{bab6.tex} merupakan kode untuk memasukan berkas bab6.tex kedalam laporan.

4.7 Penulisan *code* atau *pseudocode* program

4.7.1 *Inline*

Dengan perintah `\verb`: `System.out.println("Hello, World");`
Dengan perintah *custom* `\code`: `System.out.println("Hello, World");`

4.7.2 *Multiline*

Dengan perintah `verbatim`:

```
public class HelloWorld {
    public static void main(String[] args) {
        // Prints "Hello, World" to the terminal window.
        System.out.println("Hello, World");
    }
}
```

Dengan perintah `lstlisting`:

```
1 public class HelloWorld {
2  public static void main(String[] args) {
3      // Prints "Hello, World" to the terminal window.
4      System.out.println("Hello, World");
5  }
6 }
```

Konfigurasi tampilan bisa dilakukan di `uithesis.sty` dengan referensi dokumentasi di https://en.wikibooks.org/wiki/LaTeX/Source_Code_Listings

BAB 5
PENUTUP

@todo
Tambahkan kata-kata pengantar bab 5 disini.

5.1 Mengubah Tampilan Teks

Beberapa perintah yang dapat digunakan untuk mengubah tampilan adalah:

- \f
  Merupakan alias untuk perintah \textit, contoh *contoh hasil tulisan*.
- \bi
  Contoh hasil tulisan.
- \bo
  Contoh hasil tulisan.
- \m
  Contohhasiltulisan.
- \mc

  Contohhasiltulisan

  .
- \code
  `Contoh hasil tulisan`.

5.2 Memberikan Catatan

Ada dua perintah untuk memberikan catatan penulisan dalam dokumen yang Anda kerjakan, yaitu:

- \todo

  Contoh:

  @todo

  Contoh bentuk todo.

- \todoCite

  Contoh:

  @todo

  Referensi

5.3 Menambah Isi Daftar Isi

Terkadang ada kebutuhan untuk memasukan kata-kata tertentu kedalam Daftar Isi. Perintah \addChapter dapat digunakan untuk judul bab dalam Daftar isi. Contohnya dapat dilihat pada berkas thesis.tex.

5.4 Memasukan PDF

Untuk memasukan PDF dapat menggunakan perintah \inpdf yang menerima satu buah argumen. Argumen ini berisi nama berkas yang akan digabungkan dalam laporan. PDF yang dimasukan degnan cara ini akan memiliki header dan footer seperti pada halaman lainnya.

Untitled

Ini adalah berkas pdf yang dimasukan dalam dokumen laporan.

Cara lain untuk memasukan PDF adalah dengan menggunakan perintah \putpdf dengan satu argumen yang berisi nama berkas pdf. Berbeda dengan perintah sebelumnya, PDF yang dimasukan dengan cara ini tidak akan memiliki footer atau header seperti pada halaman lainnya.

Ini adalah berkas pdf yang dimasukan dalam dokumen laporan.

5.5 Membuat Perintah Baru

Ada dua perintah yang dapat digunakan untuk membuat perintah baru, yaitu:

- \Var
  Digunakan untuk membuat perintah baru, namun setiap kata yang diberikan akan diproses dahulu menjadi huruf kapital. Contoh jika perintahnya adalah \Var{adalah} makan ketika perintah \Var dipanggil, yang akan muncul adalah ADALAH.

- \var
  Digunakan untuk membuat perintah atau baru.

DAFTAR REFERENSI

LAMPIRAN

LAMPIRAN 1