

NATURAL LANGUAGE PROCESSING DENGAN TEKNIK STATE MACHINE PARSER

Hanif Al Fatta

Abstraksi

Aplikasi konsep AI terus berkembang. Salah satunya adalah bagaimana membuat komputer memahami perintah yang diberikan pengguna dalam bahasa sehari-hari (natural language), dan merespon perintah tersebut dalam bahasa sehari-hari. Banyak metode yang digunakan untuk menjembatani hal ini, konsepnya adalah membuat suatu metode sehingga minimal komputer bisa mengekstrak perintah dalam natural language, ke dalam format perintah yang dikenalnya. Salah satu konsep yang ditawarkan adalah Teknik State Machine Parser yang menggunakan pendekatan teori bahasa dalam bentuk G1 grammar. Teknik ini masih punya banyak keterbatasan untuk implementasi pada bahasa sehari-hari tapi cukup baik untuk proses-proses dengan batasan bahasa yang kuat.

***Kata Kunci** : AI, bahasa natural, State Machine Parser, Teori bahasa, G1 grammar*

1. What is Natural Language Processing?

Natural language processing, biasanya disingkat dengan NLP, mencoba membuat komputer mampu memahami suatu perintah yang dituliskan dalam bentuk bahasa sehari-hari (Pada tulisan ini mengasumsikan bahasa yang dipakai adalah bahasa inggris) dan diharapkan komputer juga merespon dalam bahasa yang mirip dengan bahasa natural. Setelah komputer bisa memahami perintah dalam bahasa natural, maka diharapkan sistem komputer juga dapat memberikan respon dalam bahasa natural pula. Tulisan ini dibatasi untuk tidak membahas bagaimana menghasilkan respon.

Perlu diketahui pula, speech synthesis dan speech recognition sebenarnya bukan merupakan bagian dari NLP. Natural Language processor tidak memperdulikan bagaimana suatu kalimat diinputkan ke komputer. Tugasnya adalah mengekstrak informasi dari kalimat. NLP tidak bisa digunakan sendirian, kecuali dalam riset. Tetapi NLP dapat menyediakan front end untuk program komputer yang lain-terutama database manager dan generalized problem solver. Juga untuk menjalankan sistem operasi, banyak riset yang mengarah ke NLP driven Operating System. Dan yang paling utama adalah pada pengembangan robotika, dimana dituntut interaksi yang efektif antara mesin dan manusia.

2. Pendekatan untuk NLP

Inti dari sistem NLP adalah parser. Parser adalah bagian dari program atau sistem yang membaca setiap kalimat, kata demi kata, untuk menentukan "what is what". Ada banyak jenis parser yang bisa digunakan, pada tulisan ini akan dibahas teknik *State machine Parser*. Ada 2 pendekatan yang bertolak belakang pada NLP. Pendekatan yang pertama mencoba untuk menggunakan semua informasi dalam satu kalimat seperti yang dilakukan manusia. Tujuan dari pendekatan ini adalah memungkinkan komputer melakukan percakapan. Pendekatan yang lain memungkinkan komputer menerima perintah dalam bahasa natural, tetapi cukup dengan cara meng-ekstrak informasi yang esensial dari perintah tersebut.

3. Pembatasan Bahasa

Salah satu aspek tersulit dalam mengembangkan sistem berbasis NLP adalah kompleksitas dan fleksibilitas dari bahasa manusia. Ketika kita membuat suatu Natural language processor, maka kita perlu membatasi type dari kalimat dan grammar yang diterima. Sehingga processor hanya akan memahami salah satu atau beberapa dari subset bahasa natural saja. Pada tulisan ini grammar akan dibatasi dengan asumsi semua kalimat yang dipakai adalah deklaratif dan non-interogatif dan mengikuti pola standard :

Subject, predicate, object

Juga diasumsikan bahwa:

- Semua adjective harus mendahului nouns
- Semua adverb mengikuti verb
- Semua kalimat akan diakhiri tanda baca titik

Sehingga kalimat-kalimat berikut akan diterima secara valid :

The child runs to the house

The large child runs quickly to the window

Tetapi parser tidak akan menerima kalimat seperti :

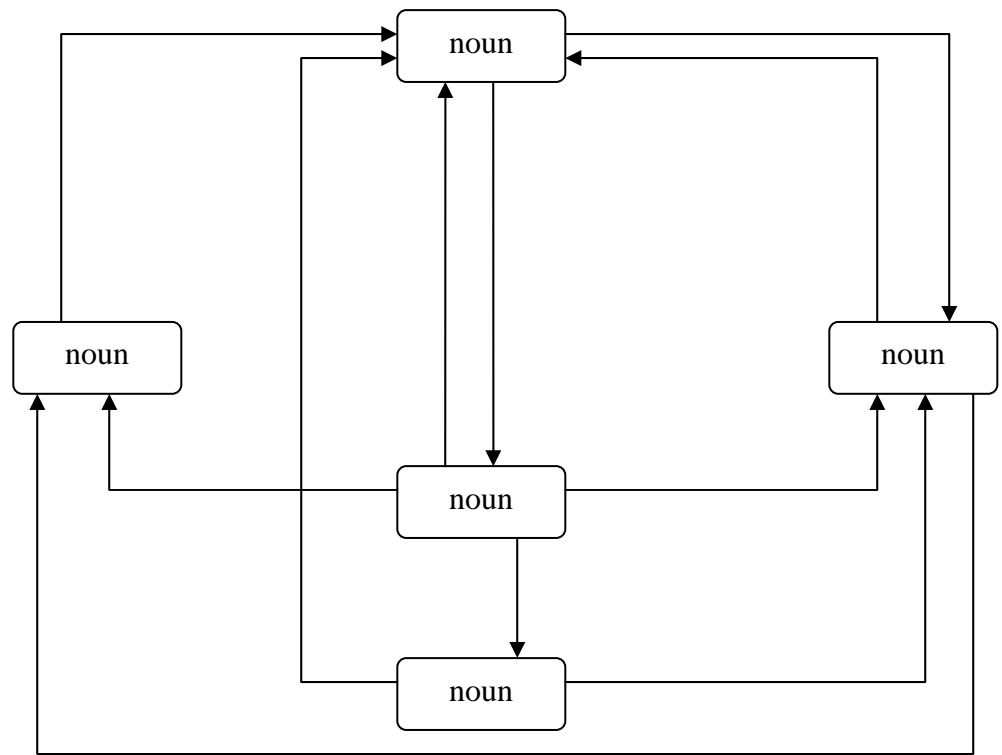
The child quickly runs to the house

Karena adverb quickly mendahului adverb runs. Untuk semua bagian dari tulisan ini akan mengacu ke pola seperti diatas yang dikenal dengan **G1 grammar**. Sebagai tambahan kita memerlukan kosakata yang bisa diterima. Untuk membuat lebih sederhana pada tulisan ini akan memakai kosakata sebagai berikut:

Word	Type
Door	Noun
Window	Noun
House	Noun
Child	Noun
Has	Verb
Runs	Verb
Plays	Verb
Large	Adjective
Quickly	Adverb
The	Determiner
A	Determiner
To	Preposition

4. State Machine NLP Parser

State machine parser menggunakan "keadaan terkini (current state)" dari suatu kalimat untuk meramalkan jenis kata apa yang secara legal mengikuti. Gambar 1 menunjukkan suatu graph berarah yang menunjukkan transisi yang valid dari suatu keadaan ke keadaan yang lain.



Gambar 1 : State Machine untuk G1 grammar terbatas

Dari contoh di atas, suatu noun dapat diikuti oleh suatu *verb* atau *preposition*. State machine ini merefleksikan G1 grammar yang kita bahas sebelumnya. Misalkan kita memasukkan suatu kalimat dengan kata-kata yang terdaftar dalam database kosakata yang kita punya sebelumnya, seperti :

The child runs quickly to the large house.

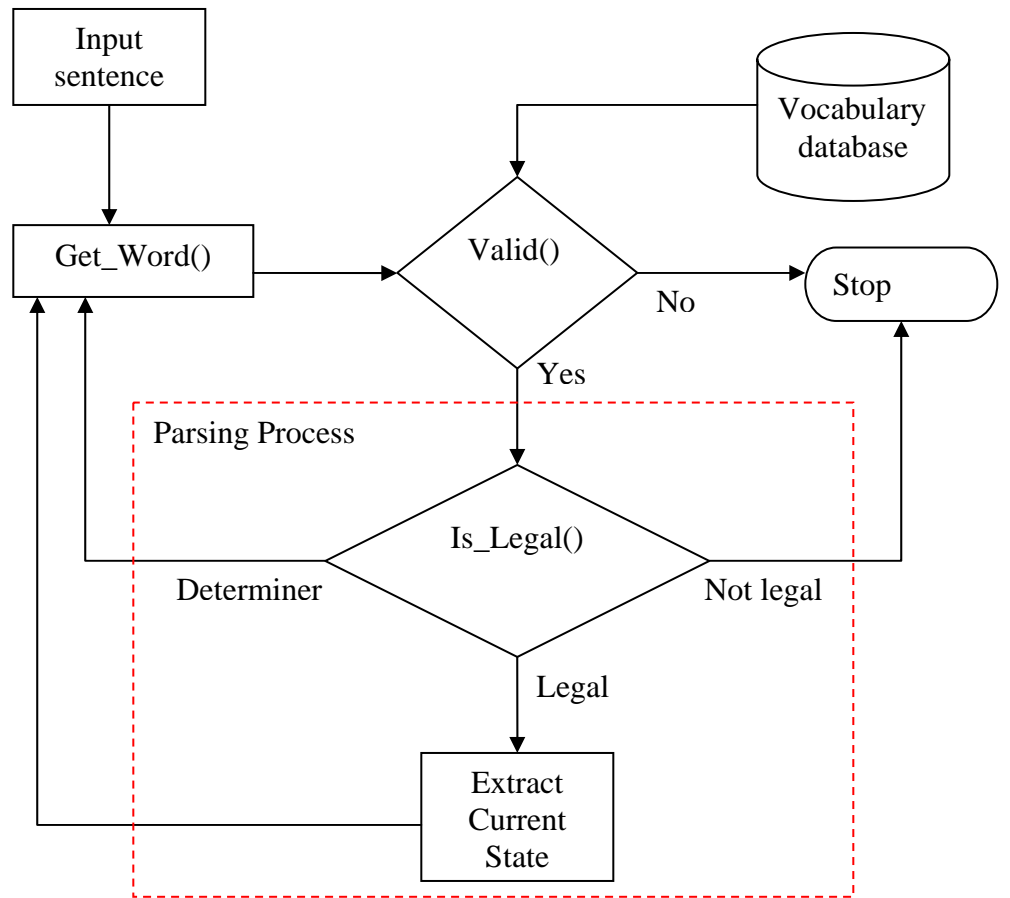
Diagram yang kita buat akan mencocokkan bahwa kalimat tersebut mengikuti G1 grammar :

the	child	runs	quickly	to	the	large	house
Det	Noun	Verb	Adverb	Prep	Det	Adj	noun

Ingat Diagram state machine pada gambar 1, kata pertama yang diambil adalah **the**. Karena the adalah sebuah *determiner* , maka parser akan membuangnya karena tidak ada perubahan keadaan yang terjadi. Kata berikutnya adalah **child**. Sebuah *noun*, yang akan merubah keadaan dari state machine menjadi NOUN. State machine sekarang dimulai. Kata berikutnya adalah **runs**. Pada gambar 1 ditunjukkan ada 2 kemungkinan perubahan keadaan dari *noun*, ke *verb* atau ke *preposition*. Karena **runs** adalah *verb* maka transisi sukses dan menempatkan keadaan baru VERB pada mesin, sehingga sekarang mesin berada pada node VERB.

Kata berikutnya adalah **quickly**. Dengan melihat diagram pada gambar 1, salah satu transisi yang valid dari *verb* adalah ke posisi *adverb*, sehingga sistem melanjutkan ke ADVERB. Dari Adverb, hanya ada 2 keadaan yang bisa mengikuti yaitu: NOUN atau PREPOSITION. Karena kata berikutnya adalah *preposition*, maka *parser* berubah ke keadaan PREPOSITION, yang harus diikuti oleh *noun* atau *adjective*. Kata berikutnya adalah sebuah *determiner* yang tidak mengakibatkan perubahan pada mesin. Kemudian kata berikutnya adalah **large**, yang menyebabkan transisi ke ADJECTIVE dan *adjective* hanya boleh diikuti oleh NOUN, yang terjadi ketika mengecek kata berikutnya yaitu **house**, terakhir parser membaca tanda titik yang mengindikasikan semua kata telah di-parse.

5. Rancangan Sistem



gambar 2: rancangan Sistem NLP berbasis Machine State

Keterangan :

1. Sistem dimulai dari kalimat yang diinputkan oleh user
2. Getword() akan mengekstrak dari kalimat yang dimasukkan user per kata.
3. Kata hasil ekstraksi Getword() dicocokkan apakah ada dalam vocabulary database yang kita miliki
 - a. Jika tidak ada proses berhenti
 - b. Jika ada lanjutkan ke langkah berikutnya
4. Kata tersebut kemudian akan masuk ke prosedur IsLegal() untuk melakukan "state transition", Jika transisi legal maka akan menghasilkan perubahan state, jika tidak ada 2 kondisi:
 - a. Jika kata yang dimasukkan merupakan determiner, maka tidak ada perubahan state.
 - b. Jika terminator seperti tanda titik maka proses berhenti
5. Jika selain kedua hal di atas, IsLegal() akan membandingkan kata yang dimasukkan dengan current State, jika legal maka kata diterima dan perubahan current state diperbolehkan. Berdasar diagram State Machine pada Gambar 1, kita bisa membuat prosedur transisi keadaan sebagai berikut:

```
case NOUN:  
if (Type==verb) return VERB  
if (Type==prep) return PREP  
break;  
case VERB:  
if (Type==prep) return PREP  
if (Type==noun) return NOUN  
if (Type==adv) return ADV  
if (Type==adj) return ADJ  
break;  
case ADV:  
if (Type==noun) return NOUN  
if (Type==prep) return PREP  
break;  
case ADJ:  
if (Type==noun) return NOUN  
break  
case PREP:  
if (Type==adj) return ADJ  
if (Type==noun) return NOUN  
break
```

6. Analisis untuk State machine NLP Parser

1. Masalah utama pada parser jenis ini adalah kompleksitas . bahkan untuk suatu G1 Grammar yang sederhana, diperlukan banyak sekali pernyataan kondisional untuk menentukan legalitas dari suatu state transition. Dapat dibayangkan berapa banyak pernyataan kondisional yang harus dibuat untuk semua grammar dalam bahasa inggris.
2. Masalah lainnya adalah, penanganan Frase (Phrase) yang merupakan modifikasi dari suatu noun. Misalnya kita punya noun frase berupa *large green beautiful house* yang merupakan modifikasi dari noun *house*, maka parser akan gagal mengenali frase ini karena pada pada current state dari mesin hanya bisa menentukan current state berupa ADJECTIVE dan NOUN saja, padahal frase *large green beautiful house* punya kedudukan sama dengan noun **house**
3. Sisi positifnya, state machine parser ideal dipakai untuk beberapa aplikasi khusus seperti NLP driven Operating system dan aplikasi database, karena dalam lingkungan ini dapat dipaksakan aturan yang menjamin user memasukkan command dalam suatu format valid dan komputer mengenal setiap kata dari format command tersebut. Dalam situasi ini state machine dapat digunakan karena hanya dipakai sedikit type kalimat yang valid, sehingga hanya perlu dibuat sedikit keadaan, berbeda dengan penggunaan dalam percakapan sehari-hari.

Daftar Pustaka

Schildt, Herbert, *Artificial Intelligence Using C*, Osborne-McGraw Hill, California, 1987