

Peran Text Processing Dalam Aplikasi Penerjemah Multi Bahasa Menggunakan Ajax API Google

The Role of Text Processing in Multi-Language Translator Applications Using Google's Ajax API

Afrizal Zein

Dosen Teknik Informatika STMIK Eresha
Jl. Raya Puspitek Serpong No. 10 Tangerang Selatan Banten
zeinafrizal@gmail.com

Abstrak --- Mesin penerjemah adalah alat penerjemah otomatis pada sebuah teks yang dapat merubah dari satu bahasa ke bahasa yang berbeda. Mesin penerjemah adalah sebuah software dengan hasil terjemahan dihasilkan atas dasar model linier regresi yang parameter-parameternya diambil dari hasil analisis statistik teks bilingual. Sekarang kami memperkenalkan langkah berikutnya dalam membuat Mesin Penerjemah yang lebih baik menggunakan metode *Neural Machine Translation*.

Cara *Neural Machine Translation* menerjemahkan seluruh kalimat dalam satu waktu, bukan hanya memenggal sepotong demi sepotong. Menggunakan konteks yang lebih luas untuk membantu mencari tahu terjemahan yang paling relevan, yang kemudian menata kembali dan menyesuaikan untuk menjadi lebih seperti layaknya berbicara dengan manusia menggunakan tata bahasa yang benar.

Program aplikasi ini dibuat menggunakan Bahasa pemrograman C# ditambah pustaka AJAX API Google untuk menerjemahkan teks dan mengambil terjemahan dengan mengurai konten JSON.

Dari hasil penelitian didapat sebuah terjemahan yang jauh lebih halus dan mudah dibaca, dan ini semua mungkin karena sistem pembelajaran *end-to-end* yang dibangun di atas *Neural Machine Translation* yang pada dasarnya berarti bahwa sistem belajar dari waktu ke waktu untuk membuat lebih baik, terjemahan yang lebih alami.

Kata Kunci : *Neural Machine Translation, Google Translate, Ajax API Google*

Abstract --- *Machine translator is an automatic translator tool on a text that is a change of language into a different language. Machine translator is a software with the result of translation is produced on the basis of linear regression model whose parameters are taken from the result of statistical analysis of bilingual text. Now we introduce the next step in making a better Translator Machine using Neural Machine Translation method.*

Neural Machine Translation translates all sentences at a time, not just piece by piece. Use the broader context to help figure out the most relevant translations, which then rearrange and adjust to become more like talking a human with the correct grammar.

This application program is built using C # programming language plus Google's AJAX API library to translate text and retrieve translations by parsing JSON content.

From the research results obtained a much smoother and easier to read translation, and this is all possible because the end-to-end learning system built on Neural Machine Translation basically means that the learning system from time to time to make better, the translation which is more natural.

Keywords: *Neural Machine Translation, Google Translate, Google Ajax API*

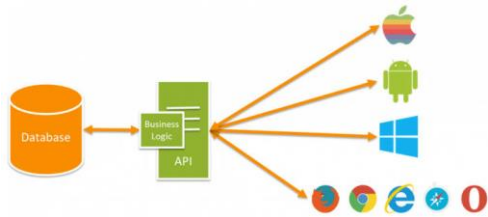
1. PENDAHULUAN

API kepanjangan dari Application Programming Interface adalah gabungan dari sekumpulan fungsi, protocol, perintah, yang dapat digunakan programmer di saat membuat program perangkat lunak yang menggunakan sebuah sistem operasi tertentu. Dengan menggunakan dan merancang menggunakan dan merancang menggunakan dan merancang menggunakan aplikasi penerjemah buatan sendiri ini, kita tidak perlu lagi membuka web situs Google Translate setiap kali akan menterjemahkan sebuah teks, namun cukup dengan

menjalankan aplikasi yang telah kita dibuat sendiri ini. Bahkan, kita bisa menambahkan semacam text to speech sehingga terjemahan yang dihasilkan dapat didengar langsung ke telinga kita.

Digunakannya API bertujuan untuk mempercepat proses *development* dengan membuat/melakukan dan menyediakan *function* dan posedor secara terpisah sehingga *developer* bertindak untuk memindahkan perintah yang sejenis. API akan diterapkan sehingga sangat terasa jika fitur yang diminati sudah sangat kompleks, tentu membutuhkan waktu untuk membuat yang serupa dengannya. Misalnya: integrasi dengan

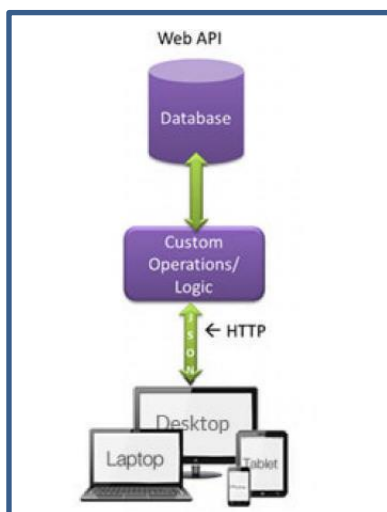
text to speech Terdapat berbagai jenis system *API* yang dapat digunakan, termasuk sistem operasi, *library*, dan web.



Gambar 1. *API* yang bekerja pada tingkat Sistem Operasi

Aplikasi yang berhubungan dan berinteraksi dengan sebuah *library* harus mematuhi serangkaian aturan yang telah ditentukan oleh *API*. Dengan menggunakan pendekatan ini memudahkan *software developer* untuk merancang dan membuat aplikasi yang berkomunikasi dengan berbagai *library* tanpa harus memikirkan kembali strategi yang digunakan selama semua *library* mengikut *API* yang sama. Kelebihan dari menggunakan metode ini adalah memperlihatkan betapa tidak sulitnya memakai *library* yang sama dengan bahasa pemrograman yang berbeda.

Web *API* adalah seperti yang telah diterangkan sebuah alamat web (*end point*) yang dirancang untuk menangani beberapa *task* sesuai *request* yang diterima, juga terkadang memiliki *parameter* yang dibutuhkan sebagai data agar dapat menampilkan hasil yang diinginkan, juga pada beberapa kasus untuk mengakses *API* dibutuhkan kode otentikasi yang diperbolehkan setelah mendapat izin untuk memanipulasi data yang telah diinginkan. Oleh Semua *rule* ini yang ditentukan oleh *programmer* yang telah membuat dan merancanganya.



Gambar 2. Web *API* Diagram

2. TINJAUAN PUSTAKA

Google AJAX Libraries *API*

AJAX Libraries *API* adalah jaringan distribusi konten dan arsitektur pemuatan untuk pustaka JavaScript open source yang paling populer. Dengan menggunakan metode Google.load Google *API*

Google Loader (), *API* menyediakan aplikasi Anda akses yang stabil, andal, berkecepatan tinggi, dan tersedia secara global ke semua pustaka JavaScript open source yang paling populer termasuk: jQuery, jQuery UI, prototipe, script.aculo.us, MooTools, dojo, SWFObject, Yahoo! Perpustakaan Antarmuka Pengguna (YUI), Ext Core dan Chrome Frame. Google bekerja langsung dengan pemegang saham utama untuk setiap upaya perpustakaan dan menerima versi stabil terbaru saat dirilis.

API Perpustakaan AJAX Google. Ini adalah sumber daya yang bagus untuk pengembang menggunakan salah satu kerangka kerja JavaScript yang populer termasuk Prototype, Script.aculo.us, jQuery, Dojo, dan MooTools. Daripada mengunduhnya ke server Anda sendiri dan menghostingnya dari sana, Anda dapat meminta perpustakaan JavaScript pilihan Anda dari ajax.googleapis.com.

Manfaat terbesar menurut saya adalah setiap pengembang dapat memanfaatkan CDN Google untuk mengirimkan sebagian besar JavaScript mereka. Dari YSlow dan buku saya, saya menerima banyak umpan balik bahwa Aturan 2: Menggunakan CDN tidak dapat dijangkau oleh banyak pengembang. Saya menggunakan jQuery dalam salah satu proyek pribadi saya dan menyajikannya di situs web yang saya hosting dari satu lokasi geografis. Aduh. Mampu memindahkan 21K lebih dekat ke pengguna saya sangat bagus.

Manfaat lain: Ada manfaat komunitas di sini. Karena semakin banyak situs yang pindah ke file ini, pengguna yang menavigasi di seluruh situs tersebut mungkin hanya perlu mengunduh file satu kali dan membacanya dari cache untuk semua situs lain. Juga, versi yang disajikan oleh Google memiliki tanggal Kedaluwarsa yang ditetapkan satu tahun di masa depan dan di-gzip. Tiga dari perpustakaan juga memiliki versi yang diperkecil yang mengurangi ukuran lain 13-21% dengan mengecilkan JavaScript sebelum gzipping.

Semua pengoptimalan kinerja ini sangat bagus untuk dilihat. YUI, alternatif hebat lainnya, sudah ditawarkan melalui CDN Yahoo! Dari yui.yahooapis.com dengan fitur kinerja yang serupa. Sudah ada hampir 50 komentar di pos Ajaxian. Kinerja adalah topik hangat sekarang saat kami mendorong browser ke batas mereka. Layanan seperti ini hanya apa yang dibutuhkan untuk membantu kami membuat pengalaman pengguna secepat mungkin.

3. METODE PENELITIAN

Neural Machine Translation

Seiring dengan peningkatan data yang sangat besar, ada dua faktor lain yang menciptakan pergeseran dalam industri lokalisasi. Pertama, pertumbuhan eksponensial kekuatan komputer, dan kedua, peningkatan minat dalam apa yang disebut pembelajaran mendalam: jenis pembelajaran mesin

yang digunakan oleh Google dalam algoritma pengenalan gambar dan suara.

Dengan mengingat faktor-faktor ini, tidak mengherankan bahwa pembelajaran mendalam baru-baru ini memicu badai dalam penerjemahan dan pelokalan untuk menciptakan apa yang sekarang kita kenal sebagai Terjemahan Mesin Neural (NMT). Setelah semua, sebagai volume data dan kemajuan teknologi meningkat, begitu juga bahan diterjemahkan. Tapi apa sebenarnya NMT, dan bagaimana cara meningkatkan efisiensi lokalisasi?

Dalam webinar baru-baru ini, Direktur Mesin Terjemahan Lionbridge, Jay Marciano, membahas penerapan metode terjemahan yang baru dan lebih akurat ini dan bagaimana hal itu mengarah pada kemajuan industri.

Cara kerja Mesin Neural Translator

Neural Machine Translation adalah paradigma yang relatif baru, pertama kali digali menjelang akhir 2014. Sebelum ini, terjemahan mesin dioperasikan pada model statistik di mana pembelajaran mesin tergantung pada database terjemahan sebelumnya, yang disebut memori terjemahan.

Sementara NMT masih melatih memori terjemahan seperti yang dilakukan oleh Mesin Terjemahan Statistik, NMT menggunakan pembelajaran mendalam — dan mungkin volume data pelatihan yang lebih tinggi — untuk membangun jaringan saraf tiruan.

Marciano menggunakan permainan catur untuk mengilustrasikan cara kerja Penerjemahan Mesin Statistik. Dalam program catur, ada alam semesta terbatas di mana sejumlah gerakan terbatas dapat dilakukan. Program ini hanya menghitung semua langkah yang mungkin untuk menemukan yang terbaik. Demikian pula, pembelajaran mesin yang terjadi dalam sistem SMT bekerja dengan membandingkan n-gram — atau 6-kata pengelompokan kata dalam sebuah kalimat — dari kalimat sumber hingga yang terjadi dalam bahasa target untuk menemukan korelasi.

Di sisi lain, Terjemahan Mesin Syaraf bisa digambarkan sebagai "membesarkan" sistem saraf, seperti dijelaskan Marciano. Ini seperti bermain piano: Saat Anda membuat kesalahan, Anda mencadangkan, coba lagi, dan ulangi sampai Anda memilikinya. Sistem Neural MT mencoba menemukan jalan mereka melalui jaringan saraf dengan cara yang sama

Dalam pengertian ini, Neural MT jauh lebih efektif daripada model yang terbatas, dan sering tidak akurat, n-gram-based. Untuk satu hal, sistem NMT dijalankan pada GPU yang kuat (unit pemrosesan grafis), daripada CPU (unit pemrosesan pusat) seperti sistem SMT. Dan meskipun Neural MT membutuhkan waktu lebih lama untuk menerjemahkan kalimat karena kekayaan data yang terlibat — karena sistem SMT membutuhkan waktu lebih lama daripada sistem berbasis aturan lama —

Statistik MT menyajikan masalah besar dengan bahasa di mana aturan terjadi di luar unit enam kata.

Tentu saja, NMT masih mengalami beberapa masalah: misalnya, ketika menerjemahkan konten yang sangat teknis. Tetapi materi sumber yang mengandung singkatan teknis yang tidak diketahui tidak akan diterjemahkan dengan baik oleh sistem terjemahan mesin, termasuk Neural MT. Untuk arah bahasa yang tidak memiliki banyak data pelatihan — misalnya, bahasa Jerman ke Korea — pembelajaran mendalam membuka kemungkinan menggunakan data pelatihan tidak langsung, atau "berputar", dari bahan sumber bahasa lain.

Perbedaan utama antara NMT dan SMT? Saat Anda menyajikan materi pelatihan ke algoritme pembelajaran mendalam, Anda tidak perlu memberi tahu mereka apa yang harus dicari. Anda membiarkan sistem menemukan pola-pola itu sendiri, seperti petunjuk kontekstual di sekitar kalimat sumber. Namun, spesifik prosesnya tetap misterius dalam banyak hal.

Neural MT dan data besar: membuang kemampuan yang terbatas Jaringan syaraf pertama kali digunakan dalam program pengenalan gambar dan suara, dengan sistem pelatihan dengan data yang diawasi — seperti gambar anjing dengan metadata yang dilampirkan. Dalam membaca metadatanya, sistem akan tahu untuk mengidentifikasi isi gambar sebagai seekor anjing.

Kemudian, sistem akan mencoba menemukan cara terbaik melalui jaringan syaraf untuk membuat tautan itu, mencadangkan dan menemukan jalur yang lebih baik jika menemukan jawaban yang salah, dan akhirnya mengembangkan jalur syaraf yang menghasilkan jawaban yang benar. Ini adalah jalur yang akan ditekan ke depan.

Dalam pengenalan suara, untuk kalimat yang terekam dalam bahasa tertentu, biasanya hanya ada satu transkripsi yang tepat untuk pembelajaran mendalam untuk menemukan — membuat pelatihan itu cukup jelas. Terjemahan melibatkan materi pelatihan "berisik" dan merupakan tugas yang lebih kompleks.

Namun, pembelajaran mendalam dan data besar, Marciano menggambarkan, memungkinkan kita untuk membuang kemampuan terbatas kita kepada orang lain

4. HASIL DAN PEMBAHASAN

Versi terbaru GoogleTranslator ini memanfaatkan API AJAX Google Terjemahan untuk menerjemahkan teks dan mengambil terjemahan dengan menguraikan konten JSON yang dikembalikan. Untuk mengarahkan saya ke API ini. Versi terbaru dari kode ini juga mencakup kemampuan untuk berbicara terjemahan dari aplikasi ini



GoogleTranslator adalah objek yang memungkinkan Anda menerjemahkan teks menggunakan kekuatan alat bahasa online Google. Aplikasi demo juga memungkinkan Anda melakukan penerjemahan balik dengan mudah. Aplikasi ini dapat digunakan sebagai penerjemah sumber daya orang miskin untuk frasa sederhana, tetapi Anda sebaiknya mengonfirmasikan terjemahan dengan penutur asli sebelum menggunakan hasilnya.

Anda menggunakan GoogleTranslator dengan membuatnya dan memanggil metode Terjemahan ().

```
using RavSoft.GoogleTranslator;

Translator t = new GoogleTranslator();
string translation = t.Translate ("Hello, Apa
khabar ?"
"English", "Indonesian");
Console.WriteLine (translation);
Console.WriteLine ("Translated in " +
t.TranslationTime.TotalMilliseconds + " mSec");
Console.WriteLine ("Translated speech = " +
t.TranslationSpeechUrl);
```

Google Translate bekerja dengan secara langsung menggunakan API terjemahan Google yang disebut dengan bentuk terjemahan online dan menguraikan hasilnya.

```
// Initialize
this.Error = null;
this.TranslationSpeechUrl = null;
this.TranslationTime = TimeSpan.Zero;
DateTime tmStart = DateTime.Now;
string translation = string.Empty;

try {
    // Download translation
    string url = string.Format
("https://translate.googleapis.com/translate_a/single
e?client=gtx&sl={0}&tl={1}&dt=t&q={2}",
```

```
Translator.LanguageEnumToIdentifier
(sourceLanguage),

Translator.LanguageEnumToIdentifier
(targetLanguage),
        HttpUtility.UrlEncode
(sourceText));
    string outputFile = Path.GetTempFileName();
    using (WebClient wc = new WebClient ()) {
        wc.Headers.Add ("user-agent",
"Mozilla/5.0 (Windows NT 6.1)
AppleWebKit/537.36 " +
"(KHTML, like Gecko)
Chrome/41.0.2228.0 Safari/537.36");
        wc.DownloadFile(url, outputFile);
    }

    // Get translated text
    if (File.Exists (outputFile)) {

        // Get phrase collection
        string text = File.ReadAllText(outputFile);
        int index = text.IndexOf
(string.Format(",,\"{0}\"",
Translator.LanguageEnumToIdentifier
(sourceLanguage)));
        if (index == -1) {
            // Translation of single word
            int startQuote = text.IndexOf("\"");
            if (startQuote != -1) {
                int endQuote = text.IndexOf("\"",
startQuote + 1);
                if (endQuote != -1) {
                    translation =
text.Substring(startQuote + 1, endQuote -
startQuote - 1);
                }
            }
        }
        else {
            // Translation of phrase
            text = text.Substring(0, index);
            text = text.Replace("[", ", ");
            text = text.Replace("]", string.Empty);
            text = text.Replace("[", string.Empty);
            text = text.Replace("\", \"");
        }

        // Get translated phrases
        string[] phrases = text.Split (new[] { "\" " },
StringSplitOptions.RemoveEmptyEntries);
        for (int i=0; (i < phrases.Count()); i += 2) {
            string translatedPhrase = phrases[i];
            if (translatedPhrase.StartsWith(", ")) {
                i--;
                continue;
            }
            translation += translatedPhrase + " ";
        }
    }
}
```

```

// Fix up translation
translation = translation.Trim();
translation = translation.Replace(" ?", "");
translation = translation.Replace(" !", "");
translation = translation.Replace(" ,", "");
translation = translation.Replace(" .", "");
translation = translation.Replace(" ;", "");

// And translation speech URL
this.TranslationSpeechUrl = string.Format
("https://translate.googleapis.com/translate_tts?ie
=UTF-
8&q={0}&tl={1}&total=1&idx=0&textlen={2}&
client=gtx",

HttpUtility.UrlEncode (translation),

Translator.LanguageEnumToIdentifier
(targetLanguage),

translation.Length);
}
}
catch (Exception ex) {
this.Error = ex;
}

```

Seperti yang Anda lihat, logika yang digunakan untuk mengurai hasil JSON sangat sederhana!

5. KESIMPULAN

- Berdasarkan pemaparan di atas, dapat disimpulkan bahwa penggunaan mesin translate menggunakan api ajax google dapat menjadi solusi bagi kita untuk memahami teks secara keseluruhan sehingga bisa menjadi solusi dalam memahami teks bahasa asing atau literatur-literatur dalam bahasa asing.
- Hasil terjemahan lebih baik dan lebih natural, namun, kita harus dapat memahami kedua struktur bahasa dan budaya dalam bahasa sumber dan bahasa sasaran sama baiknya sehingga kesalahan dalam memahami teks dapat diminimalisir.

DAFTAR PUSTAKA

- [1] Bell, Roger T.. 1991. Translation and Translating: Theory and Practice. New York: Routledge..
- [2] Bassnett, Susan. Translation Studies. London: Routledge, 2002.
- [3] Hatim, Basil dan Munday, Jeremy. 2004. Translation : An advanced resource book. London: Routledge.
- [4] Hoed, Benny Hoedoro. 2006. Penerjemahan dan Kebudayaan. Jakarta: PT Dunia Pustaka Jaya.
- [5] Hasyim, Muhammad. Perspektif Semiotika atas Aspek Budaya dalam Penerjemahan Teks Kuliner Perancis. Fakultas Sastra: Universitas Hasanuddin.