Web-based automatic translation: the Yandex.Translate API

Maarten van Hees
LIACS, Leiden University
m.van.hees@umail.leidenuniv.nl

Paulina Kozłowska
LIACS, Leiden University
pk.kozlowska@gmail.com

Nana Tian
LIACS, Leiden University
n.tian.2@umail.leidenuniv.nl

ABSTRACT

Yandex.Translate Application Programming Interface (API) is easy to use automatic translation service provided by Russian Internet Company Yandex. As a statistical machine translation system, it is based on statistics derived from the web sources. Context and purpose of this technology is presented and operating principles based on three pieces model are outlined. Furthermore, the paper describes strengths and weaknesses of the API and machine translation in general as well as selected existing applications of Yandex.Translate API. It is followed by the instruction how the API can be implemented in the simplest way. At the end, conclusions are made about further possible applications of this technology and its accessibility.

1. PURPOSE, CONTEXT AND HISTORY

Machine translation (MT) is a field of computer linguistics exploring the possibility of using algorithms automatically translating text or speech from one language to another. What is important, it applies to natural languages, which means only human written or spoken languages, as opposed to artificial ones [9].

The field of "machine translation" was described for the first time by Warren Weaver. He was an American mathematician, interested in probability and statistics, most known for writing the "Translation" memorandum in 1949 [11]. Weaver, inspired by the cryptography, assumed that every language developed by humankind have basic common elements and, as a consequence, can be decoded. He was the first scientist who described the possibility of using electronic computers to translate between natural languages. However, he wanted a machine to be able to not only translate a text word by word, like an automated dictionary, but also handle the problem of word order, ambiguous terms and context. By these means, Weaver introduced the idea of Statistical Machine Translation (SMT), able to learn from existing large corpus of existing text. It compares the parallel texts containing the same information in different languages then discovers the rules of translation and creates statistical model.

However, until 1990s, the most popular translating systems were rule-based: using dictionaries as well as grammar and syntax rules [2]. The simplest method is direct translation, mapping input to output with basic rules. More advanced, a transfer-based machine translation makes morphological and syntactic analysis. Finally, researchers were interested in systems using interlingua: an abstract, universal language which mediates in a translation between a source language and target language [5].

At the turn of the 20th and 21st century computing power and data storage increased significantly, Internet resources were growing and digital technology came to be relatively cheap and available to everyone. Furthermore, researchers started to understand that languages are too complex and context-sensitive to be reduced to an encoded set of rules. For these reasons, a research in data-driven translating methods was rapidly developing. Nowadays, SMT can easily crawl the Internet and use existing translated documents or extract text from HTML in several language versions of websites. Moreover, along with the ubiquity of personal computers and the Internet, the MT applications become more popular among non-professional users [5].

An example of statistical machine translation system is Yandex.Translate. Yandex.Translate was implemented in 2011 a part of is a Russian Internet company called Yandex [13]. It can be used in the web browser as well as in a mobile application. Yandex.Translate translates whole sentences, words or web pages between 46 languages.

- 1949: Warren Weaver describes the possibility of using computer based machine translation to translate text between natural human languages; idea of Statistical machine translation (SMT) [11].
- 1954: The Georgetown-IBM experiment: first public demonstration of a computer for translating languages using 250 words and six ‘grammar’ rules [4].
- 1966: publication of the ALPAC (Automatic Language Processing Advisory Committee) report criticizing MT methods because of poor quality and high costs [5].
- 1990s-2000: development of the statistical based MT
- 1997 The official launch date of the yandex.ru search engine.
- 2011: Implementation of machine translation system Yandex.Translate.
- 2013: Yandex releases the Android Yandex.Translate application.

2. OPERATING PRINCIPLES

The Yandex search engine is comprised of a three piece model [6]. A translation model, a language model and a decoder. The translation model is simply a list of all known words in a language with their translations into other languages. This means that each language has its own translation model. These translation models are built by cross-referencing texts and works that have been translated into different languages, also called a parallel corpus. First the system looks for correspondent phrases, then to groups of words or single words and uses these to calculate the probability based on previous encounters that the translation is the correct one. It continuously process new texts to increase the possibility to encounter the word in different contexts, this is also why you need so many sources.

As translation model handle the transferring parallel words or phrases from one language to another, the language model is responsible for knowledge about target language. It ensures that the output will resemble text written or said by a human. It uses sources in target language as a training data and learn how to arrange words in the right sequence by assessing the probability of occurring translated phrase in a text.

The third part of the model is the decoder. The decoder does the actual translation. It combines the user input and the translation model to generate a list of possible translations, sorted by probability. This probability is based on the amount of times the source text (or parts of the source text) where encountered in the source texts used to form the translation model. The decoder has an added HTML parser to translate whole web pages if the user wishes to do so. Moreover, the decoder uses the language model to evaluate the possible variations and choose the variation with the highest frequency of use [5].

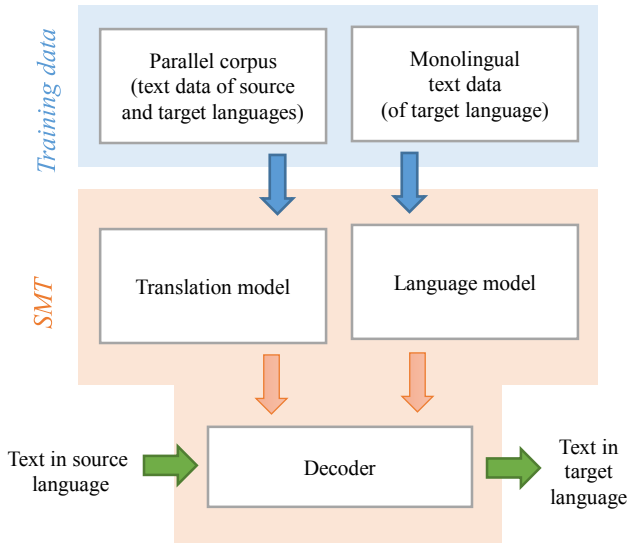

Figure 1. Three piece model of SMT

The Yandex Translation API works on the REST principle. REST stands for Representational State Transfer and is stateless. This means all the transactions (or URL calls) have no relation to one another, because no client context will be stored on the server. This means you have to authenticate every time you make a call to the service. This is not so much a problem, since the REST principle was designed with scalability in mind, and does that very well. The nature of the scalability lies in the way the data is provided. Every time a user makes a call to an URL you get a representation of the underlying resource, but the underlying resource does not change. This means you can have the same resource for everyone how calls it, but it will not necessarily look the same.

3. STRENGTHS AND WEAKNESSES

The most important strength of statistical machine translation such as Yandex.Translate is the way the system is setup. It means it can continuously improve the translation models when new sources become available. In the Internet, resources grow every day: SMT can crawl the web searching for parallel corpus of texts, for instance documents uploaded by multinational companies [5]. The advantage of using an official documentation is that the language is controlled, thus the output is more predictable. Next to corporate websites, commercial readme's are also a good source for translation. The advantage of SMT comparing to older, rule-based systems is that latter has finite dictionaries and are constructed by group of developers. The fact, that modern machine translation is based on statistic derived from the web sources, makes the translation more flexible and almost limitless. Thus, it also support language and cultural diversity [2].

Something influencing the quality of translation is the way the source texts are translated. Official text probably will be translated in better way, however in case of unusual structured phrases the meaning can become completely obscured. Commonly used expressions, proverbs and idioms cannot be decomposed into words without losing intended meaning. Therefore, MT translating each word in phrase separately will fail and probably produce an absurd sentence. Both: older rule-based and statistical machine translation have no internal representation of semantics, however SMT is able to better manage with translating whole expressions, because it can apply phrase alignment. Still, web-based translators cannot handle polysemy like human translator can, because people understand words in a context [9].

There is a drawback in the way the system is set up, using the REST principle. Some of the offered functionality has the ability of adding a callback function. The problem here is that REST does not support asynchronous invocation of functions, because a REST service is a stateless thing, there is no relation between transactions. This means the user needs to keep this in mind when designing his application, because it may disrupt the flow of the program.

4. TYPICAL APPLICATIONS

4.1. Written Text Translation

Yandex.Translate is the most commonly used in online websites, first of all in Yandex webpage [11] and mobile applications. In both cases, the user select source and target languages from a predefined list of 46 different languages or automatically identify a source language, then he types a text in source language in one box and afterward the output appears in second box. When translating individual words, the application can display both the definition and the full dictionary entry including comments and usage examples. What is more, the application provide additional functionality, which is speech synthesis from typed word or phrase.

Mobile application has the same functionality and moreover, there is a possibility to download an offline dictionary with examples of common usage. A user can also dictate the text, then the application recognize speech and translate the input. Furthermore, it provides visual text recognition that works for 11 languages so far, allowing instant translation from a picture. The application access the camera rolls in mobile devices and import the photos to quickly scan and identify foreign text (such as that found in a sign or a menu), and then translate and display the words in another language on the device's screen. The words are displayed in the original context on the original background, and the translation is performed in real-time (Figure 2) [14] [15].

Figure 2. Visual text translation

4.2. Instant Mail and Website Translation

Yandex.Mail has implemented Yandex.Translate API, giving an opportunity to translate a message into another language as a user is writing it. It works in similar way as online web application: source and target languages should be specified, there are also two boxes, message in source language should be typed in the left one and output immediately appears in the right one. Simultaneous translation works with a text up to 1000 characters in length, if a message is longer, a user need to use the "Translate" button which translate a whole, already written message [11].

Furthermore, Yandex.Translate can be implemented as an extension in a browser. If the page translator is enabled in the Yandex.Browser, it detects automatically the language used on a website. If a language is incorrectly recognized or a user wants to change a target language, he can select appropriate languages manually. As a result, the website will be translated as a whole. Moreover, if a user do not want to translate a whole web page, he can select only a phrase or word and translation will appear in a small box [10]. The Yandex.Translate extension can be also installed in other browsers, like Mozilla Firefox, Chrome or Opera, but existing plug-ins usually have limited functionalities and they only translate whole pages.

5. SURPRISING APPLICATIONS

5.1. Scratch Translation Block

Scratch is a free educational programming language designed for children and teenagers, however often used by non-professional programmers in every age. The language works with blocks, not with real written language, to create simple programs. These users are often not very versed in the technical intricacies surrounding programming, so things like API keys and handling HTTP status codes are not typically things these beginner programmers do. This is where a special translation block comes in. This translation block handles everything around the connection to Yandex and allows users to jump straight into translating stuff. This translation block is written in JavaScript and contains an API key for direct usage [3].

5.2. Bad Translator

An interesting example of the Yandex.Translate API unintended application is the Bad Translator website. The aim of its creators was to show how unreliable automatic translators can be and how sometimes they produce incoherent output. Bad Translator translate back-and-forth between English and various foreign languages using Yandex.Translate (or other online translators can be selected from list). It shows how translating back to English distorts the meaning of source text, usually at the end the result is surprising and very different from text typed in the beginning. In this way, Bad Translator exposes a flaw of the machine translation, which is rooted in the statistical nature of the system and the fact there is no syntactical or grammatical knowledge available [1].

6. GETTING STARTED

In this section we guide a user how to get started using the Yandex.Translate API. We will address the user using a form "you".

6.1 Getting API key

Before you can use the Yandex.Translate API you need an API key. An API key is a unique code that can be used to gain entrance to the API's methods, without one any call to the API will result in an access denied error. Getting an API key is easy enough, just sign up to the Yandex tech domain. After registering, logging in and agreeing to the terms you can visit the keys page.

```
https://tech.yandex.com/keys/
```

This page lists the available keys and whether if they are valid or not. You can also block keys from here if you suspect your key might be compromised. An API key looks like this:

```
trnsl.1.1.20150526T172412Z.10721aed43e0645d.
6b5543a47b77f071a299621b406981cac5a5f70f
```

Copy the key to your program, you will need this in when getting data from the API.

Now that we have gained access to the API we can start using it. Before we do though, let's have a look at the structure of the calling links. Being a web API, the methods the API can be used via web URLs. These URLs look like this:

```
https://translate.yandex.net/api/v1.5/tr/method?key=APIkey&param=value
```

There are three parts to take note of here. The bold printed *method* needs to be replaced with the actual method (or function) of the API you want to use. More on that later. Your API key into the place of the bold print *APIkey*, take care to copy it using CTRL+C from the website, as it is easy to make errors when manually typing it in. The final bold printed is variable, different methods require different parameters. Again, more on that later when we explain the different methods available in the API.

6.2 The available methods

In the previous section we have talked about getting an API key for Yandex.Translate and how the URLs look that we need to call in order to use the APIs functionality. The URL points to a method within the API, these methods are *detect*, *getLangs* and *translate*. Since Yandex.Translate is a statistical translation system, it depends on the available (data models) whether a translation is even possible. To find out if we can do a translation, we can use the *getLangs* function.

The *getLangs* method returns a list of languages that Yandex.Translate supports. This is a very useful function, when you incorporate this into your program, your program will be able to make a decision to not translate if a certain language is not supported. This function call looks like this:

```
https://translate.yandex.net/api/v1.5/tr/getLangs?key=APIkey&ui=en
```

The *ui* parameter here is important. Without it, you will not retrieve the list of supported languages. Yandex is able to translate between any of the languages that are in this list. The value of the parameter is the language the list will be in, if you want it in Russian, you put *ui=ru*. Currently only 4 languages are supported: Russian, Ukrainian, Belarusian and English. The list you retrieve contains pairs like this:

```
<Item key="ru" value="Russian"/>
<Item key="en" value="English"/>
<Item key="pl" value="Polish"/>
```

This is standard XML and can thus be easily parsed. The *detect* method does exactly what it implies, it attempts to detect the language of the input text. You can use this function in combination with the getLangs function to find out if a translation is possible. You can use the detect function with the same URL structure:

```
https://translate.yandex.net/api/v1.5/tr/detect?key=APIkey&text=Hello+world
```

The text is put in with the *text* parameter, words are delimited by a +, lines are delimited by new *text* parameters, you can input as much *text* parameters as you want. When calling, this returns a response code embedded in XML format, it looks like this:

```
<DetectedLang code="200" lang="en"/>
```

The 200 means the function succeeded in detecting the language. When failing to detect you will get a 40x error, depending on what went wrong.

The final function is the *translate* function. This function is obviously the meat and bones of the API. Using this function works again in the same way:

```
https://translate.yandex.net/api/v1.5/tr/translate?key=APIkey&lang=en-ru&text
=Text+to+translate
```

Before you start using this method, you need to know the translation direction you want to use. This is denoted by the *lang* parameter. The value of this parameter is a language-to-language pair, like en-ru would mean English to Russian. Now would be a good time to use the *getLangs* and *detect* functions to get the languages you need. The *text* parameter works the same as in the *detect* function.

Now that we know all that, let's create a little example: let's translate the Norwegian word "Trekirker" to German. The *lang* parameter will be *no-de,* Norwegian (no) to German (de) and the *text* parameter will be *trekirker*, thus resulting in this link:

```
https://translate.yandex.net/api/v1.5/tr/translate?key=APIkey&lang=no-de&text=trekirker
```

When we call this (do not forget the API key!) we will receive this output:

```
<Translation code="200" lang="no-de">
       <text>Holzkirchen</text>
</Translation>
```

Judging from the code 200, our translation was successful!

7. FINAL THOUGHTS

Yandex.Translate API has a potential to be used in various ways and on different devices. For example, for now, it is not used in social media as an instant translator. However, Yandex has already launched a commercial version of the API, which is able to translate around 100 billion characters per day and as a consequence, can be used by large amount of network users [7]. Yandex.Translate could be also implemented in chat applications or other programs based on communication between people.

Moreover, in more general terms, digital translators are cheap and ubiquitous and they may contribute to the globalization. The globalization that is accessible to everyone with smartphone and Wi-Fi or special hand-held devices. Everyone can travel without knowing foreign languages and easily translate everything using SMT, from text or even from speech or images [5]. Even if an output is not perfectly accurate, today's web-based translators help to understand the general meaning of the source text.

However, there is a challenging question: how to evaluate a text translated by a machine? Hiring a human translators to assess the output would elevate the costs and negatively influence the speed of translation. There should be an easier way to assess the quality of the system – such as fluency and adequacy of language - and to decide which one of possible translations is the most fitting. Researchers proposed automatic evaluation metrics who compare the output text made by machine translation with human translations [5]. Still, it requires additional time and system engaged to test the final result.

Apart from the obvious question surrounding the quality of translation that Yandex offers (which is also present in other translation systems) the Yandex API is an easy to use, readily available and well-structured API. The fact that you can create as many API keys as you want means it becomes very easy to structure and distribute the usage evenly among your applications while keeping things decentralized. The offered functionality is fairly limited, but it covers the drawbacks of Yandex' structure sufficiently. One little point of complaint is the fact that this translation machine is still very Russia (and thus Cyrillic) oriented, you can't get the list of languages in anything else than English and a few Cyrillic languages. But apart from this, the Yandex.Translate API is definitely a good alternative to other proven translation APIs such as Google Translate.

REFERENCES

1. Bad Translator. *Akuna*. http://ackuna.com/badtranslator. Accessed 06.10.2015.
2. Cronin, M. 2013. Translation in the Digital Age. New York: Routledge.
3. Hanning, Kreg. n.d. Scratch Translation Block. *YouTube.* https://www.youtube.com/watch?v=0cTEot5H-Y4. Accessed 06.10.2015.
4. Hutchins, J. 1954. The First Public Demonstration of Machine Translation: the Georgetown-IBM System. AMTA Conference.
5. Koehn, P. 2010. Statistical Machine Translation. New York: Cambridge University Press.
6. Machine Translation. *Yandex Company*. https://company.yandex.com/technologies/translation.xml. Accessed 06.10.2015.
7. Munby, R. Yandex Launches New Commercial Translation Service. 22.08.2014. http://rusbase.com/news/author/robinmunby/yandex-translation-service. Accessed 06.10.2015.
8. Slocum, J. 1985. A Survey of Machine Translation: its History, Current Status, and Future Prospects. *Computational Linguistics* 11(1).
9. Tohmetov, T. A. et al. 2014. The Problems of Machine Translation. Молодежь и современные информационные технологии: сборник трудов XII Всероссийской научно-практической конференции студентов, аспирантов и молодых ученых 2: 267-268.
10. Translating pages and words. *Yandex Company*. https://help.yandex.com/yabrowser/features/translators-settings.xml. Accessed 06.10.2015.
11. Weaver, W. 1955. Translation (1949). Reproduced in W.N. Locke, A.D. Booth (eds.). *Machine Translation of Languages*, 15–23. MIT Press.
12. Message translation. *Yandex Company.* https://help.yandex.com/mail/letter/translation.xml. Accessed 06.10.2015.
13. Yandex Translate. https://translate.yandex.com. Accessed 06.10.2015.
14. Yandex.Translate - Android Apps. https://play.google.com/store/apps/details?id=ru.yandex.translate. Accessed 06.10.2015.
15. Yandex.Translate iOS App. Mobile Action. https://www.mobileaction.co/app/ios/tr/yandex.translate/584291439. Accessed 06.10.2015.