Ptakopět

Překlad tam a kontrolně zpět

Vilém Zouhar

December 2018
Rev. 3

1 Overview

Ptakopět (Překlad Tam A KOntrolně zPĚT) is an agile browser agnostic plugin, which aims to help people translate to languages, where they are unable to verify the quality of the translation, by offering backward translation of the result. The issue at hand is sometimes known as *Outbound Translation*, which is a complement to *Gisting*. Both in *Gisting* and *Outobound Translation*, the user speaks only one of the languages. In the former, the users is responsible for creating correct messages, while in the later the responsibility is to correctly interpret them.

1.1 The purpose of the project

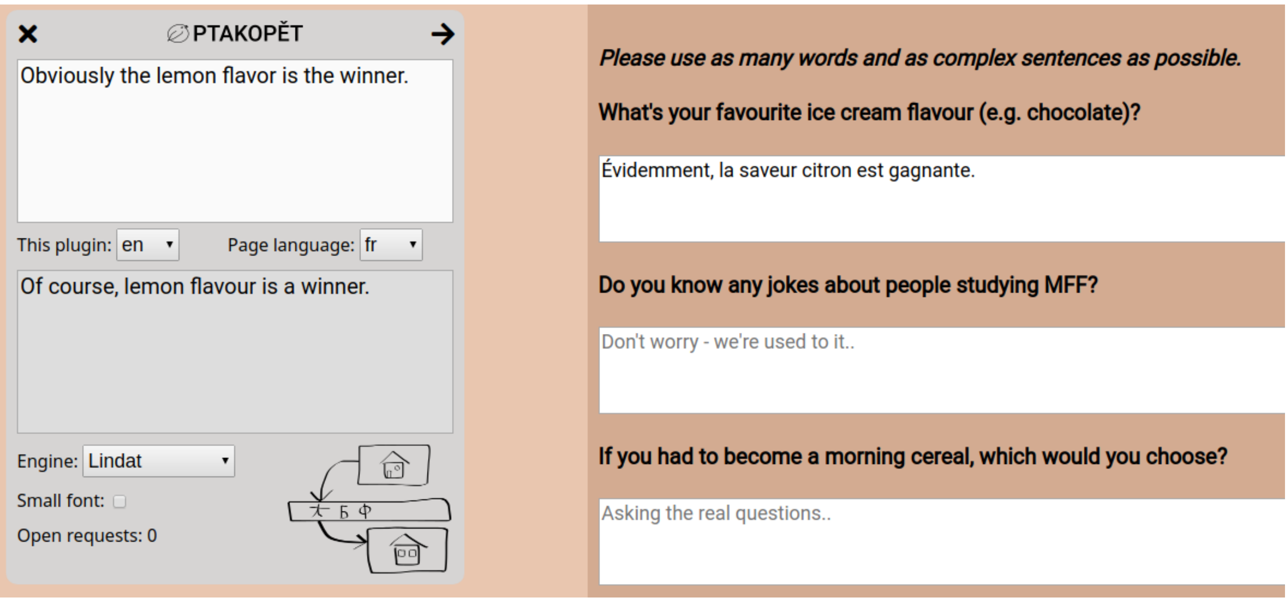
Many users use machine translation to verify their own translation, or can at least affirm, that the machine translation is valid. Translation systems are, however, not perfect and despite their great power, they tend to perform poorly on unseen phenomena (unknown words, unexpected word order, etc.). Users translating to languages, which they don't master enough to validate the translation could use Ptakopět to verify (and assure them), that the machine translation output is valid. Due to the fact that many encounters with foreign languages happen on the Internet, Ptakopět is developed as a browser extension.

1.2 Other solutions

Backwards translation is only one of the solutions to *Outbound Translation*. One of the possible solutions could include more autonomous systems, such as text highlighting or numeric responses. Backward translation relies on the ability of the user to assess the differences between texts.

1.3 The scope

Ptakopět aims to provide a backward translation interface to already existing translation engines. It does not do the translation itself in any way. Existing translation backends are provided by the Charles University, namely by Martin Popel and other associates (as of version 1.0.4, CS-EN and EN-FR language pairs are available). The interface is provided in a form of browser agnostic plugin, that can also be hardwired to website as a plain script.



PTAKOPĚT

Obviously the lemon flavor is the winner.

This plugin: en ▾ Page language: fr ▾

Of course, lemon flavour is a winner.

Engine: Lindat ▾

Small font: □

Open requests: 0

Please use as many words and as complex sentences as possible.

What's your favourite ice cream flavour (e.g. chocolate)?

Évidemment, la saveur citron est gagnante.

Do you know any jokes about people studying MFF?

Don't worry - we're used to it..

If you had to become a morning cereal, which would you choose?

Asking the real questions..

2 Implementation

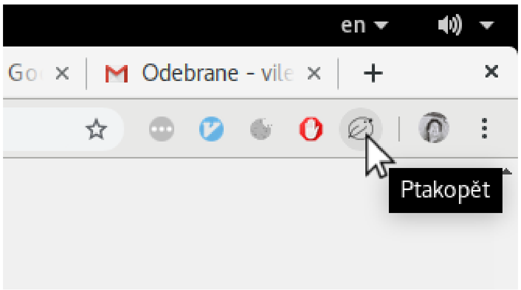
The plugin itself is written in JavaScript, but injects HTML, CSS and jQuery framework into web page. Making the plugin indifferent to whether it is run as a background sciprt (browser plugin) or as an user script (included in web page) posed a substantial obstacle. Resulting implementation works, but is unfortunately affected by web styling. Eg. website CSS can define the looks of the plugin, which is not intended and can result in usable, but not aesthetically pleasing design.

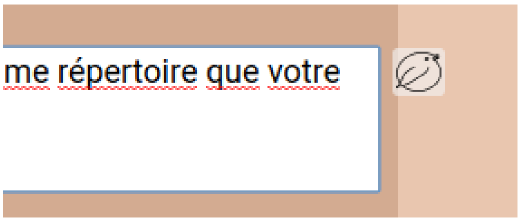
Ptakopět tries to capture all input events to *textareas* and *inputs*. This is a very robust approach, but about half of the websites (as of 2018) use custom input elements (to provide styling and extra functionaliy). As a result, these elements can't be captured using Ptakopět. This is obviously a radical usability issue, but we didn't find any universal solution to this problem.

Text events are processed and forwarded to translation engine using AJAX requests. This could eventually pose a threat to these backends should the plugin be used by many people simultaneously. The engines would be glutted by requests, as every keypress invokes one or two translations. They are indistinguishable from legit requests, as the plugin works user-side. This could be solved by a standard approach of postponing requests to some interval of no key press (*ptakopet_arch.js:89*).

2.1 Technical details

The code itself is located in the *src/* directory. The *manifest.json* file is obligatory for all plugin versions. When Ptakopět is run as a background script (browser plugin), then *ptakopet_background_bootstrap.js* is executed, which adds listeners to context menu and icons. Then *ptakopet_init.js* is run as a user script and from this point on, it is identical to having the script run from the web page. Due to asynchronous loading, the startup functions are called in *ptakopet_init_user.js*. Generally, *ptakopet_init.js* handles visual elements (eg. displaying *floater.html*, CSS etc.), while *ptakopet_translator.js* takes care of the translation flow. Ptakopět was designed to support fast addition of new translator backends or supported languages. Each translation object in Ptakopět manages it's own AJAX requests in a standardized form.





me répertoire que votre

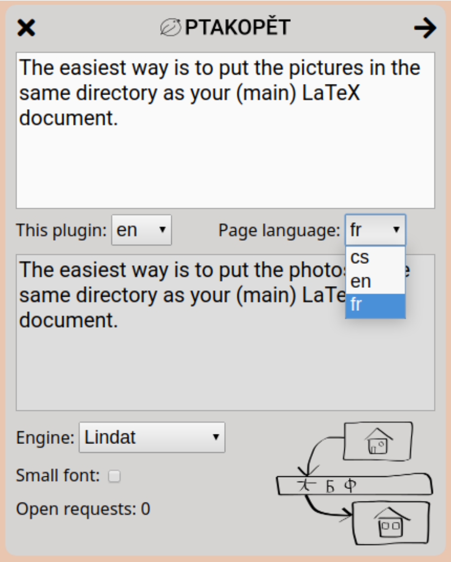
3 Standalone instalation

This plugin can be also hardwired into a web page (mostly for presentation purposes). It is sufficient to download all source files (the *src/* directory on GitHub) and add this line to your target HTML file:

```
<script id='ptakopet_init' src='dev/ptakopet/js/ptakopet_init.js'></script>
```

4 Browser plugins

Basic demonstration and info about Ptakopět online publication is located at ptakopet.vilda.net. Ptakopět was also presented at MFF Open Days (November 2018). The presentation page is located at vilda.net/s/dod_ptakopet. The plugin itself is distributed as a Chrome and Firefox extension. The plugin was also submitted to the Opera addon store, but still hasn't been reviewed. Source code is published at a GitHub repository: github.com/zouharvi/ptakopet, the license is not yet decided, but feedback and code contributions are welcome.



The easiest way is to put the pictures in the same directory as your (main) LaTeX document.

This plugin: en ▾ Page language: fr ▾

The easiest way is to put the photo
same directory as your (main) LaTe
document.

Engine: Lindat ▾

Small font: □

Open requests: 0

5 Possible extensions

Future versions could also include a full fledged web interface similar to that of Google Translate (one page). Since Ptakopět is somewhat limited to backend capabilities, their extension is also desired (more language pairs, more information about translation confidence, etc.).

6 Conclusion

The first version of Ptakopět met all imposed usability criteria and didn't exceed the expected amount of work required.