

Ptakopět

Překlad tam a kntrolně zpět

Vilém Zouhar

November to December 2018
Rev. 2

Overview

Ptakopět (Překlad Tam A KOntrlně 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 purpose of the project

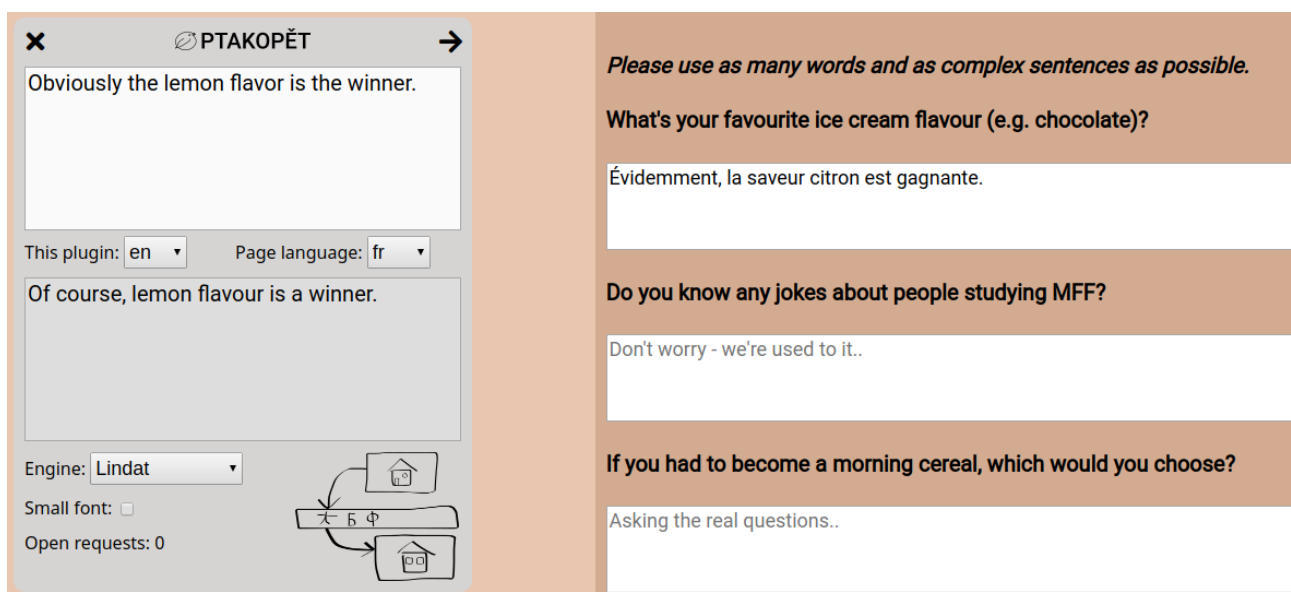
The issue at hand is known as *Outbound Translation*. Many users use machine translation to verify their own translation, or can at least affirm, that the machine translation is valid. Translation systems are, however, not perfect and despite their great power, they tend to perform poorly on unprecedented 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. As many encounters with foreign languages happen on the Internet, Ptakopět is developed as a browser extension.

Other solutions

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

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. The interface is provided in a form of browser agnostic plugin, that can also be hardwired to website as a plain script.



Implementation

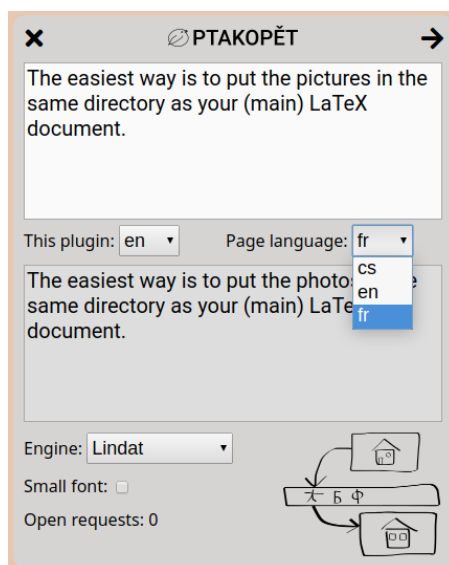
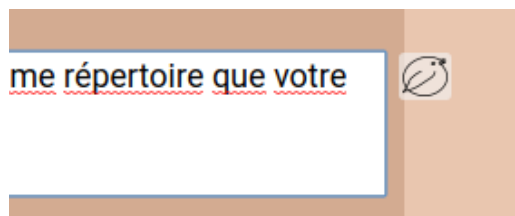
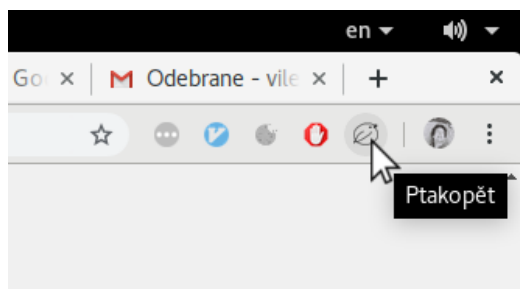
The plugin itself is written in JavaScript, but injects HTML, CSS and jQuery framework into webpage. Making the plugin indifferent to whether it is run as a background script (browser plugin) or as an user script (included in webpage) posed a substantial obstacle. Resulting implementation works, but 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 functionality). 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, if the plugin was 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 postponing requests to some interval of no key press (standard approach).

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 *ptakopte_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 webpage. Due to asynchronous loading, the startup functions are called in *ptakopte_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 its own AJAX requests in standardized form.



Publishing

Ptakopět was published online. Basic demonstration and info is located at ptakopet.vilda.net. Ptakopět was also presented at MFF Open Days, but for technical reasons we were unable to collect data. 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 add-on store, but still hasn't been reviewed. Source code is published at a GitHub repository: github.com/zouharvi/ptakopet, the licence not yet decided, but feedback and code contributions welcome.

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.

Conclusion

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