

Ptakopět

Překlad tam a kontrolně zpět

Vilém Zouhar, Ondřej Bojar (advisor)

December 2018
Rev. 4

1 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 issue at hand could be referred to as *Outbound Translation*, which is a complement to *Gisting*. Both in *Gisting* and *Outbound Translation*, a message is transferred between an author and a recipient the user has sufficient knowledge of only one of the languages. In *Outbound Translation*, the user is responsible for creating correct messages, while in *Gisting* the responsibility is to correctly interpret them.

1.1 The purpose of the project

When translating to foreign languages users cooperate with machine translation tools to produce the best result. 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

Backward translation is only one of the solutions to *Outbound Translation*. One of the possible solutions could include more autonomous systems, that would require less effort on the user side. This could be done with text highlighting, multiple translations or numeric responses (confidence intervals or just single number describing the system's certainty). Backward translation relies heavily 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. The interface is provided in a form of browser-agnostic plugin, that can also be hardwired to website as a plain script.

2 User interface

2.1 Usage

Ptakopět runs as a small plugin located, if active, in top left or right corner. The intended workflow is to write text to the top textarea and validate against the backward translation in the bottom one. During this process, the translated text appears in the selected area on the web page. Active target input element is changed every time a possible element (all textareas and text inputs) gets focus.

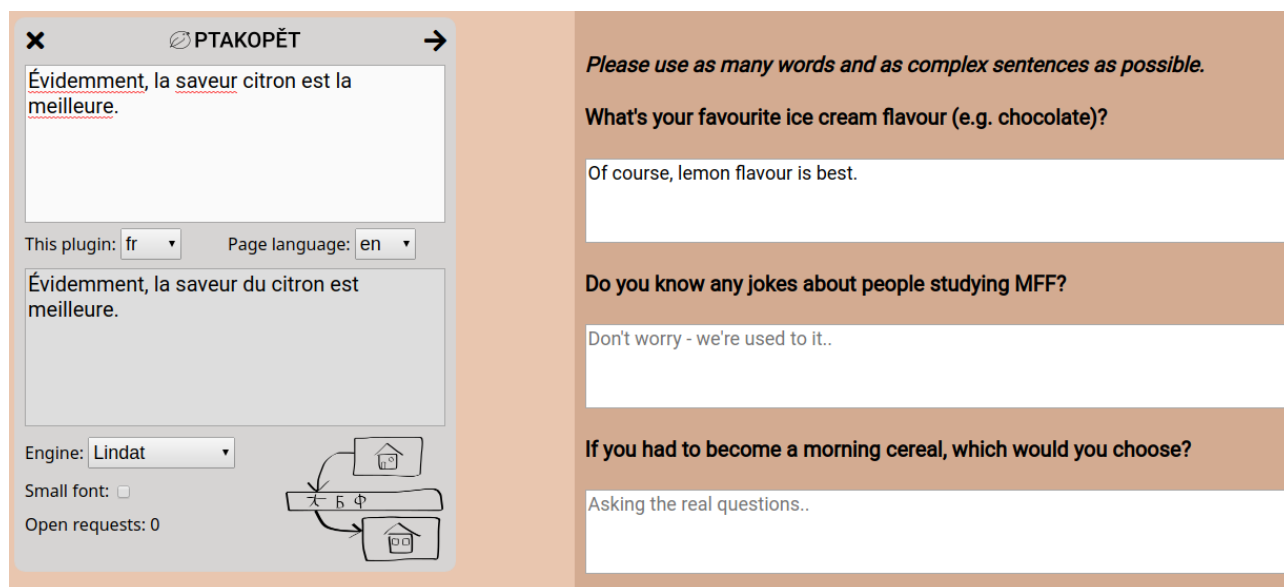


Figure 1: Ptakopět helps an French speaking user with filling in an English form

As seen in Figure 1, the user selected the first web page input element, hence making it active to Ptakopět and wrote a text in their native language (French) to the first Ptakopět input. The backward translation removed a definite article for an uncountable noun, otherwise the sentence matches the original.

The English translation of the input sentence appeared in the target textarea, as expected.

Should the user now edit the text in English, translation would appear in the bottom Ptakopět textarea. Writing into the top Ptakopět input field would overwrite the manual changes, as hinted in the bottom right diagram in the Ptakopět window in Figure 1.

2.2 Activation, miscellaneous

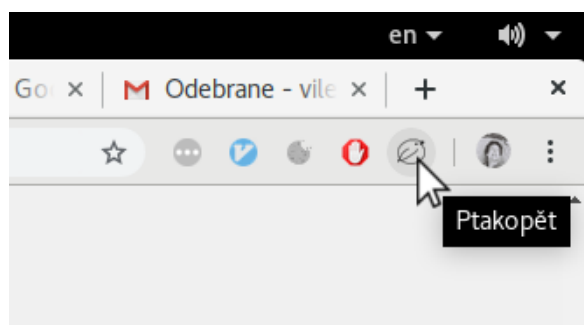


Figure 2: Ptakopět can be launched from toolbar

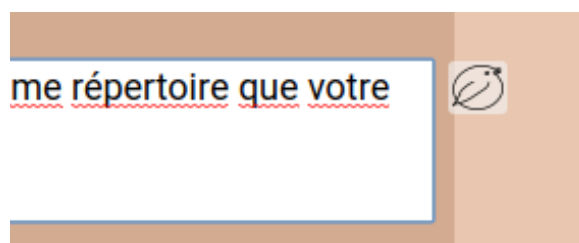


Figure 3: An icon for input activation appears next to possible text elements

Ptakopět can be invoked in multiple ways: by clicking the toolbar button, as shown in Figure 2, by launching it from the context menu (right mouse button) or by clicking an icon, that appears next to all web page input elements. The last two methods also set the active input to the current input element.

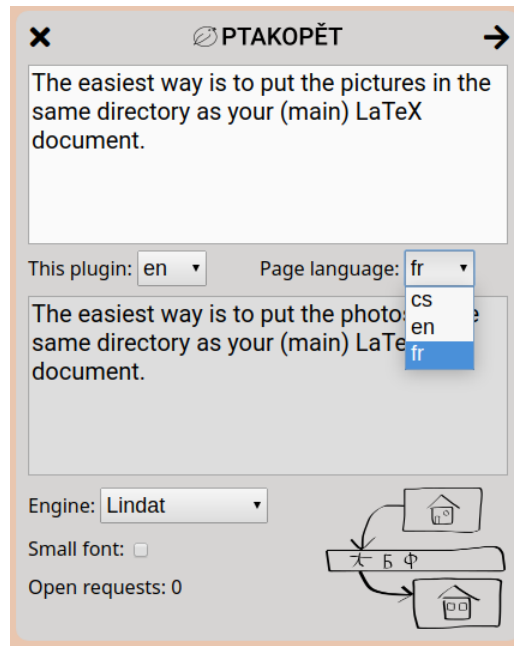


Figure 4: Ptakopět plugin user interface

Source and target languages and backend engines can be swapped by the *select* element, as shown in Figure 4. As of version 1.0.5, CS-EN and EN-FR language pairs are available for the Transformer Lindat backend [1] and CS-EN, EN-ES, EN-DE, EN-FR for the Khresmoi service [2]. For convenience reasons, *Small font* checkbox was added to allow work on larger texts.

3 Implementation

3.1 Base structure

The plugin itself is written in JavaScript, but injects HTML, CSS and jQuery framework into the web page. Making the plugin agnostic to whether it is run as a background script (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.

3.2 Text inputs

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

3.3 Translation requests

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*).

3.4 GitHub repository structure

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 its own AJAX requests in a standardized form.

4 Deployment

4.1 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) to some publicly available server foldre and add this line to your target HTML file:

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

All web pages containing this line of HTML will have the plugin activated. If a user has already Ptakopět installed, the server local version is used.

4.2 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.

5 Future work

Possible extensions could also include a full fledged web interface similar to that of Google Translate (one page). This could result in better user interface and overall clarity. 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.

References

- [1] Martin Popel, Dušan Variš, Ondřej Košarko, Translation neural net model, build on top of Tensor2Tensor, ÚFAL, Lindat Clarin, 17th December 2018 <https://lindat.mff.cuni.cz/services/transformer/>
- [2] Ondřej Bojar, Medical information analysis and retrieval, ÚFAL, 17th December 2018 <https://ufallab.ms.mff.cuni.cz/bojar/mt/khresmoi.php>