

Parallel Corpus of Computer Terms

Radovan Garabík

Eudovít Štúr Institute of Linguistics
Slovak Academy of Sciences
Bratislava, Slovakia

Abstract

Presented Parallel Corpus of Computer Terms (no acronym) is a prototype of parallel corpus containing rather specialised kind of data. Translation of software user interfaces is especially convenient for our uses, because these translations are already aligned, quite a lot of software is available with licenses allowing us to use the texts without any additional burdens, and it is already present in digital form. Presented corpus contains 1.6 million phrases in 88 languages, containing together more than 11.5 million words. The sources were different OpenSource programs, mostly from KDE, GNOME and GNU Translation Project. The backend is built using MySQL relational database system, with simple WWW interface. The corpus is publicly available through the Slovak National Corpus WWW page.

1 Introduction

Corpora play an important rôle in modern linguistics[1], a situation greatly facilitated by current boom of cheap and widely available computing power. Parallel corpora form a smaller, but nevertheless important part of corpus linguistics, and have direct utilisation for end users dealing with bi- or multi-lingual texts.

When compared with “traditional” monolingual corpora, parallel corpora have several distinguishing features and their creators have to deal with specific problems. First of all, parallel corpora need parallel texts in several languages, which can be sometimes a big obstacle. To get the rights to use texts is inheritably much more difficult than is the case of monolingual

corpora¹, because we need to consider different copyright law(s) in different countries, which by itself is rather difficult subject.

Then there is the question of aligning, using manual alignment is often impractical (even if typical sizes of parallel corpora are of an order of magnitude smaller than typical sizes of monolingual corpora), and writing tools for automatic alignment is far from trivial. On the other hand, parallel corpora are not really expected to have such a detailed and elaborate linguistic markup as monolingual ones, since their main usage and area of interest is shifted away from intrinsic linguistic properties of given language, towards relation between the languages.

2 Translations in Software Products

Rather recent phenomenon in software world is the existence of *internationalization*² and *localization*³, which reflect the penetration of computers into many regions of human society, and the subsequent need to use the software either: 1. to work with language (text documents, DTP, databases) other than English, and 2. by people without adequate command of English. Point (1) is the aim of internationalization – modifying software to be able to deal with languages other than English, point (2) of localization – making software communicate with users in their respective languages. Of these, we are particularly interested in localization, because it implies the necessity of translating user interface(s) into targeted languages.

Unfortunately, in case of commercial software, to get the texts in electronic form suitable for inclusion into a corpus is probably even more difficult than with monolingual corpora. However, there exists a lot of translated software under different OpenSource licenses, such as GNU General Public License, GNU Lesser General Public License, BSD License, Artistic license, X11 license and derivatives, and these kinds of licenses allow us to include the translations without hindrances.

The standard system and API for translations in OpenSource world is the GNU gettext[2] system, although there are numerous exceptions. Using gettext computer translations has many advantages – the most important for a parallel corpora is the fact that the translations are perfectly aligned on expression level.

¹while, according to usual copyright laws, it is possible to use texts for educational purposes, it is not really clear if it is possible to make such a corpus publicly accessible

²often shortened to *i18n*

³or *l10n*



Figure 1: Example of a popular software translated into different languages. Notice how the translation affects the direction of menus and icons.

Using such a specialised area of translations, we have to be aware of several consequences. Overwhelming majority of translations (as of the software in general) is prepared by amateurs in linguistic profession, though professionals, or at least highly skilled in software engineering. This is strikingly different from translations of commercial software, where most translations are done by professionals in linguistic area, but untrained in computer skills, and therefore often unaware of true meaning of texts being translated (but, to be fair, often with the help of consulting computer specialists).

This can be seen as both the advantage and a disadvantage. Disadvantage, because the quality of translations is often very poor, with many mistakes and mistranslations. Advantage, because it better reflects the actual use of language by computer specialists, not as prescribed by institutional bodies and norms, often disconnected from real life⁴.

3 Sources of Translations

We used KDE[3], GNOME[4] and GNU Translation Project[5]. There are often several versions of a given software, sometimes with changes in UI, which

⁴For example, in the Slovak expressions of current version of the parallel corpus, there is exactly one (1) occurrence of (otherwise officially prescribed) Slovak word “lomka”, whereas the (officially forbidden) alternative “lomítko” occurs 28 times

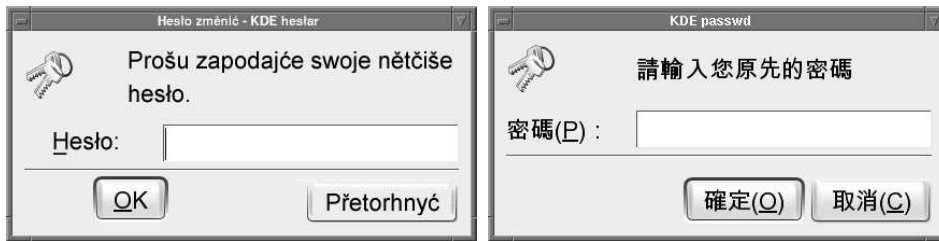


Figure 2: Password change program in two different translations.

are further reflected in translations. We decided to include the translations from the older versions as well, because it provides more translated expressions. The only slight disadvantage is the fact that new versions are often an impulse for translators to correct mistakes in their previous translations, and by keeping the old ones we keep also the less correct ones.

Using the above mentioned sources also means that the original language is always English. In fact, software with first language other than English is very difficult to find.

4 Connecting Translations

As an example, let's take original term "File". This is present in almost every GUI software, but can have different translations, according to contexts. For example, it has been translated (in different programs) into one target language as "Súbor", "SÚBOR", "file" and "zadaného súboru", respectively, while into the second target language it has been translated as "Fajl", "ДАТОТЕКА", "Датотека" and "Файл".

We decided to retrieve expressions up to second level of connections, e.g. for user query "zadaného" we get matching expression "zadaného súboru", which in turn corresponds to original expression "File", giving back to user the results "Súbor", "SÚBOR", "Fajl" . . .

The situation is illustrated on picture 3.

5 Plurals

Good internationalization of plural handling is surprisingly difficult from computers' (and programmers') point of view. We do not need to go into lengthy descriptions about languages having only singular, having dual, paucal and similar grammar categories – this has been described elsewhere[6, 7].

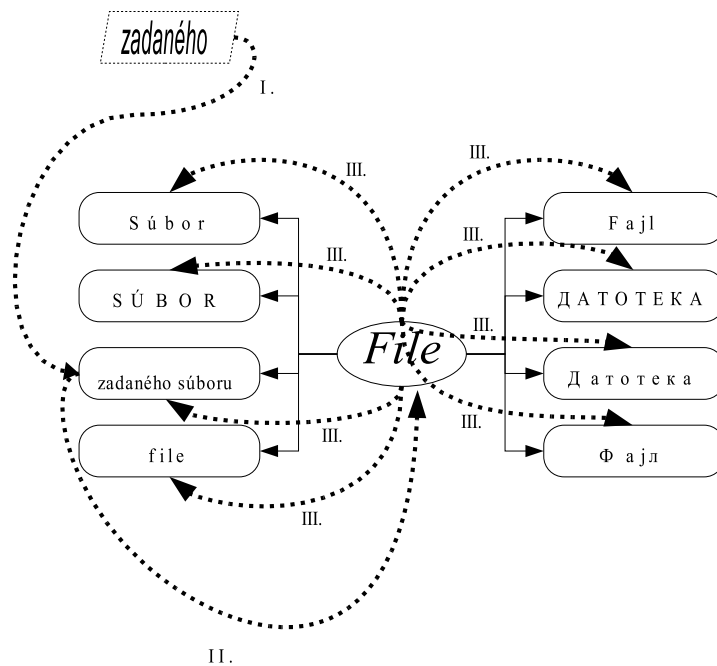


Figure 3: Full lines show the connections between original and translations, dotted lines the path of user query to a corresponding expression (I.), then to the original expression (II.) and back to all available translations (III.)

We just need to realize that the decisive factor for a computer system is the specific *form* of textual unit, not the division into grammar categories⁵. Just by looking at Slavic languages, we see different noun form for numeral 1, then a different form for 2, 3 and 4, a different form for numerals greater or equal than 5, and often repeating scheme for numerals greater than 20, according to modulo 10, with many exceptions.

While and English-speaking program uses simple piece of code like this,

```
print "Uploaded", number,
if number==1:
    print "file"
else:
    print "files"
```

typical Slavic-speaking equivalent would be lengthy and complicated. On the other hand, similar Chinese- or Hungarian-speaking program would be rather simpler. These issues have even more serious effects on internationalization, since we cannot reasonably expect program authors to know and use all the different rules for many different languages, not speaking about bloat and unmaintainability of such software.

GNU gettext solves the situation by requiring explicit enumeration of different forms, and by providing a rule for choosing the appropriate form. So the expression would be

```
Plural-Forms: nplurals=1; plural=0;
```

for Hungarian,

```
Plural-Forms: nplurals=2; plural=n != 1;
```

for English, and

```
Plural-Forms: nplurals=3; \
    plural=(n==1) ? 1 : (n>=2 && n<=4) ? 2 : 0;
```

for Slovak.

Situation gets even more complicated as we move into more exotic languages. We decided to keep things simple and we are grouping all the possible plural forms together, so for example query “súbory” would match “súbor” and vice versa.

⁵This is trivial and obvious for computer scientists and programmers, however this notion is rather unnatural for linguistically oriented persons. We felt obliged to elaborate on this.

6 Technical Details

Each expression has a unique ID number, and all the expressions are stored in a MySQL database. Two tables are used, the first one keeps the connection between expression and its ID, information about whether the expression is an original or a translation, and the language of given expression.

The second table contains pairs of expression number (be it original or a translation) and a link to corresponding translated (or original) expression. The lookup consists first of finding list of IDs matching user query (with optional restriction about expression language), and then for each of these ID a list of corresponding linked expressions is retrieved and transformed into expressions again. The search system has simple WWW cgi-bin interface (fig. 4) providing a way to search for a substring in the expressions, with a possibility to limit the search only for “interesting” languages. Everything is implemented in the Python programming language.

Though MySQL is probably the worst solution for corpus backend storage⁶, and we are leaving the searching for the substring to internal MySQL processing, which add up tot the overall inefficiency, the combined power and ease of use of Python and MySQL lead to very quick deployment of the whole corpus. As an additional feature, it is possible to update the expressions on the fly without taking the corpus down or reindexing it. The speed of resolving queries is quite acceptable, but the improvements (especially by using more corpus manager-like backend) are planned.

The corpus is publicly available via Slovak National Corpus WWW page⁷.

7 Statistics

Currently, the corpus contains 1.6 million different expressions in 88 languages. The number of words is more than 11.5 million – many scripts do not use any separators between words, we have counted only those that are separated by whitespace and common interpunction, so the number will be probably noticably higher. Only counting characters in Chinese expressions, we get additional million of “words”, if we can consider one chinese character to be one word. There are additional 2.3 million characters in Japanese expressions, however one Japanese word typically consists of several characters. And we are not speaking about other languages with scripts without word delimiters – though Japanese and Chinese are probably the most prominent ones.

⁶in author’s opinion, the second worst is the XML format

⁷<http://korpus.juls.savba.sk/>

Following table shows for each language the number of expressions present in the corpus, relative count with respect to the whole number of expressions, and relative count with respect to the number of English expressions (in other words, how many % of original expressions have been translated).

ISO 639	Language	Nº expressions	rel. [%]	rel.to English [%]
af	Afrikaans	5094	0.30	3.25
am	Amharic	4306	0.25	2.75
ar	Arabic	52109	3.08	33.24
az	Azerbaijani	18868	1.12	12.04
be	Byelorussian	19098	1.13	12.18
bg	Bulgarian	10732	0.63	6.85
bn	Bengali; Bangla	11663	0.69	7.44
br	Breton	289	0.02	0.18
bs	Bosnian	5027	0.30	3.21
ca	Catalan	41313	2.44	26.35
cs	Czech	32984	1.95	21.04
cy	Welsh	15448	0.91	9.85
da	Danish	59991	3.55	38.27
de	German	63785	3.77	40.69
el	Greek	25452	1.50	16.24
en	English	156769	9.27	100.00
en_GB	English (Great Britain)	17957	1.06	11.45
eo	Esperanto	8364	0.49	5.34
es	Spanish	69736	4.12	44.48
et	Estonian	38381	2.27	24.48
eu	Basque	8193	0.48	5.23
fa	Persian	6581	0.39	4.20
fi	Finnish	27884	1.65	17.79
fo	Faeroese	1702	0.10	1.09
fr	French	74611	4.41	47.59
ga	Irish	6995	0.41	4.46
gl	Galician	12036	0.71	7.68
he	Hebrew	18293	1.08	11.67
hi	Hindi	11197	0.66	7.14
hr	Croatian	7902	0.47	5.04
hsb	Upper Sorbian	5758	0.34	3.67
hu	Hungarian	31723	1.87	20.24
ia	Interlingua	70	0.00	0.04
id	Indonesian	12162	0.72	7.76

ISO 639	Language	N ^o expressions	rel. [%]	rel.to English [%]
is	Icelandic	5964	0.35	3.80
it	Italian	40189	2.38	25.64
ja	Japanese	39188	2.32	25.00
kn	Kannada	1380	0.08	0.88
ko	Korean	21143	1.25	13.49
ku	Kurdish	1198	0.07	0.76
lat	Latin	2667	0.16	1.70
lg	Ganda	202	0.01	0.13
li	Limburgish	5067	0.30	3.23
lo	Laothian	2658	0.16	1.70
lt	Lithuanian	19014	1.12	12.13
lv	Latvian, Lettish	15095	0.89	9.63
mi	Maori	310	0.02	0.20
mk	Macedonian	17600	1.04	11.23
ml	Malayalam	4887	0.29	3.12
mn	Mongolian	17972	1.06	11.46
mr	Marathi	687	0.04	0.44
ms	Malay	15499	0.92	9.89
mt	Maltese	5069	0.30	3.23
nb	Norwegian (Bokmål)	4854	0.29	3.10
ne	Nepali	2435	0.14	1.55
nl	Dutch	38978	2.30	24.86
nn	Norwegian (Nynorsk)	15263	0.90	9.74
no	Norwegian	16176	0.96	10.32
nso	Northern Sotho	3685	0.22	2.35
oc	Occitan	678	0.04	0.43
pl	Polish	32856	1.94	20.96
pt	Portuguese	33194	1.96	21.17
pt_BR	Portuguese (Brasil)	38004	2.25	24.24
ro	Romanian	24548	1.45	15.66
ru	Russian	43281	2.56	27.61
se	Northern Sámi	4823	0.29	3.08
sk	Slovak	43869	2.59	27.98
sl	Slovenian	28411	1.68	18.12
sq	Albanian	13000	0.77	8.29
sr	Serbian	19112	1.13	12.19
sr@Latn	Serbian (Latin)	14982	0.89	9.56
ss	Siswati	5017	0.30	3.20

ISO 639	Language	N ^o expressions	rel. [%]	rel.to English [%]
sv	Swedish	39819	2.35	25.40
ta	Tamil	9829	0.58	6.27
th	Thai	5593	0.33	3.57
tr	Turkish	40821	2.41	26.04
uk	Ukrainian	22201	1.31	14.16
ur	Urdu	100	0.01	0.06
ven	Venda	3151	0.19	2.01
vi	Vietnamese	14088	0.83	8.99
wa	Walloon	7969	0.47	5.08
xh	Xhosa	6492	0.38	4.14
yi	Yiddish	1739	0.10	1.11
zh	Chinese	96	0.01	0.06
zh_CN	Chinese (PRC)	28397	1.68	18.11
zh_TW	Chinese (Taiwan)	26368	1.56	16.82
zu	Zulu	3812	0.23	2.43
Total		1691903	100.00	1079.23

am ፋይል:-	fr Fichier :	ru Файл:
ar &ملفات:	hi फाइल:	se Fiila:
ar ملف:	hr Datoteka:	sl Datoteka:
ar ملف :	hsb Dataja:	sr Fajl:
az Fayl:	hu fájl:	sr Фajл:
be Файл:	ja ファイル :	ss Lifayela:
bg Файл:	ko 문서:	th แฟ้ม:
bn ফাইল:	ko 파일:	uk Файл:
bs Datoteka:	lat Scapus:	ur :فائل
cs Soubor:	lo ຝາຍ:	vi Tệp:
da Fil:	lt Byla:	yi פּרעקט
de Datei:	lv Fails:	zh 檔案 :
el Αρχείο:	mk Датотека:	
en File:	mn Файл:	
es Fichero:	mt Fajl:	
eu Fitxategia:	pl Plik:	
fa پرونده:	ro Fișier:	
fi Tiedosto:		

Figure 4: Screenshot showing the WWW interface. Notice several alternative translations existing in some languages.

References

- [1] M. Šimková: *Počítačové spracovanie prirodzeného jazyka a Slovenský národný korpus*. Budmerice: Počítačová podpora prekladu (2003)
- [2] *GNU gettext manual*.
<http://www.gnu.org/software/gettext/gettext.html>
- [3] T. Diehl: *The KDE Translation HOWTO*.
<http://i18n.kde.org/translation-howto/>
- [4] *The GNOME Translation Project*.
<http://developer.gnome.org/projects/gtp/>
- [5] *GNU Translation Project*.
<http://www2.iro.umontreal.ca/~gnutra/po/HTML/>
- [6] F. Čermák: *Jazyk a jazykověda*. Praha, Karolinum, 2001
- [7] “Grammatical number” Wikipedia: *The Free Encyclopedia*, Wikimedia Foundation Inc. Updated 14 April 2004.
http://en.wikipedia.org/wiki/Grammatical_number