Computer-Assisted Language Comparison in Practice

Tutorials on Computational Approaches to the History and Diversity of Languages

Mapping Multi-SimLex to Concepticon

Johann-Mattis List
Department of Linguistic and Cultural Evolution
Max Planck Institute for Evolutionary Anthropology

Multi-SimLex (https://multisimlex.com) is a multilingual resource which provides user ratings for word pairs translated into different languages. The data is important for the evaluation of methods that derive word embeddings from large corpora. While it is on the one hand desirable to link such a large dataset to Concepticon, it is difficult to do so in concrete, given that the datasets represents word similarity ratings withouth any clear reference to concepts. In this post, I will show how the data can nevertheless be linked to Concepticon, and how the original Multi-SimLex data can be represented without losing any information in the form of a Concepticon Concept List.

Multi-SimLex

Multi-SimLex (Vulić et al. 2020) is a dataset that provides user ratings for word similarities for a base list of 1888 word pairs that were translated into 12 languages and rated for similarity by human participants. The authors distribute the database in two files, one containing the translated word pairs (`translation.csv`, https://multisimlex.com/data/translation.csv) and one containing the ratings for the words pairs in individual languages (`scores.csv`, https://multisimlex.com/data/scores.csv). Word pairs have a unique identifier, which is provided in numerical form. This means in turn, that every word in the Multi-SimLex dataset can be represented by the base identifier and an index, which indicates the position of the word in the word pair (0 or 1, or 1 or 2, depending on one's preferences). Words in word pairs can recur across the datasets. Thus, the English word "leg" occurs both in the word pair 728 *leg / arm* (position 1) and in the word pair 1025 *limb / leg* (position 2).

Apart from each word pair itself, no clear hints regarding the intended meaning of the word are provided. In many cases, the meaning, however, can be clearly inferred from the word pair itself, and one should also assume that users infer the meaning when they are being ask to rate the words for their similarity. As an example consider 704 *car /*

Computer-Assisted Language Comparison in Practice
Volume 4, Number 3
https://calc.hypotheses.org/2684/

Published on 10/03/2021

bridge, where *bridge* clearly refers to the building that you use to cross a river, and 1018 *card / bridge*, where *bridge* clearly refers to the card game.

The translation process does not seem to be based on ratings, but has rather been done by individual native speakers (this is clearly indicated in the paper, see 859f). It is not clear, which tools were used for the translation (e.g., if the translation was done with the help of an automatic lookup first and later manually corrected, or whether the whole process was done manually, etc.). These decisions were left to the collaborators, and no information on the translation process is provided to my knowledge.

Occasionally, translations are problematic, be it that they misinterprete the intended meaning of the words in a given word pair (consider 1018 *card / bridge*, which is translated as *tarjeta / puente* in Spanish, although, as mentioned before, the best translation for *bridge* is the name of the card game in Spanish, bridge or be it that the word in the target language shows a different part of speech (consider 89 *monk / cross* and the Spanish translation *cruzar* for *cross*, which is a verb, not the noun, which is clearly indicated). Since the authors clearly state that translations should reflect the semantic relations in English as best as possible, these cases can be treated as translation errors. One translation (Estonian “zero” in 953 *infinity /zero*) is missing.

The authors of Multi-SimLex emphasize that one of the major restrictions of the translations is that all word pair translations must be unique and that no pair can consist of two identical words (see ibid. 859). Thus, when encountering a word pair like 58 *automobile / car*, one cannot translate this as *Auto / Auto* in German.

Translators can use more words in their translation for the same word in the English list of word pairs. Thus, *leg*, which occurs twice in Multi-SimLex, is translated one time as *saar* in Estonian (728 *leg / arm*), and one time as *jalg* (1025 *limb / leg*). Multiple translations for the same English word may be warranted in some cases, where English words suffer from a missing specification (such as *add*, which has two counterparts in Russian, one being mathematical, cf. Russian складывать for 1731 *add / divide*, one not, cf. Russian соединять for 1410 *join / add*), and in some cases they may also reflect misunderstandings, as in 220 *man / husband* vs. 338 *man /victor* for example, where the first “man” is translated as мужчина “man” while the second one is translated as человека “human being”, which — judging from the translations in the other languages — is not the first reading of “man” what one would think of in the word pair *man / victor*.

This short description of the data shows that it is not clear if it is possible to infer the concepts underlying the word pairs in Multi-SimLex. It is clear that a conceptual reading

will be invoked in many cases, when readers are asked to compare one word with another word, but it is at the same time difficult for us to judge if multiple translations for the same word in English are due to free variation in the target language, or due to underspecification of the English word forms.

Testing Multi-SimLex

Given the two explicit translation requirements mentioned before (no identical word pairs per language, and no identical words per pair), it is useful to test formally if the data conforms to this criterion. This can be quickly done with a small Python script in which we use the `csvw` package to read the CSV file (`pip install csvw`) and assume that the two files have been downloaded in advance and placed into the same folder.

```
from csvw.dsv import UnicodeDictReader from collections import defaultdict

languages = [
   "ENG", "ARA", "CMN", "CYM", "EST", "FIN", "FRA", "HEB", "POL",
   "RUS", "SPA", "YUE"]
visited = {language: defaultdict(list) for language in languages}
errors = []
with UnicodeDictReader('translation.csv') as reader:
   for row in reader:
      for language in languages:
         w1, w2 = row[language+' 1'], row[language+' 2']
         if w1 == w2: errors += [(language, w1, w2, row['ID'])]
         visited[language][w1, w2] += [row['ID']]

if not errors:
   print("all checks on same word passed")
else:
   print("found {0} errors on same word".format(len(errors)))

count = 1
for language in languages:
   for w1, w2 in visited[language]:
      if len(visited[language][w1, w2]) != 1:
         print("{0:4} | {1:15} | {2:15} | {3:15} | {4}".format(
               count, language, w1, w2, ' '.join(
                  visited[language][w1, w2])))
         count += 1
```

If you run this script, you will receive something like the following output:

```
found 25 errors on same word
...
16 | CMN      | 书        | 主题       | 249 983
17 | CMN      | 书        | 文章       | 397 708
18 | CMN      | 现实      | 幻想       | 431 871
19 | CMN      | 选择      | 选举       | 528 1423
20 | CMN      | 肌肉      | 舌头       | 690 696
21 | CMN      | 毁灭      | 建造       | 826 1498
22 | CMN      | 快        | 迅速       | 1075 1130
23 | CMN      | 容易      | 困难       | 1093 1097
24 | CMN      | 困难      | 简单       | 1110 1265
25 | CMN      | 奇怪      | 古怪       | 1143 1195
...
37 | FRA      | stupide   | idiot      | 1061 1204
38 | FRA      | vite      | rapide     | 1075 1130
39 | FRA      | bizarre   | étrange    | 1091 1143 1175
40 | FRA      | difficile | simple     | 1110 1265
41 | FRA      | stupide   | intelligent| 1155 1169
...
55 | RUS      | проход    | коридор    | 94 686 702
56 | RUS      | одежда    | ткань      | 209 629
57 | RUS      | книга     | тема       | 249 983
...
74 | YUE      | 笨        | 醒目       | 1155 1169
75 | YUE      | 接受      | 拒絕       | 1311 1485
76 | YUE      | 給予      | 借         | 1344 1442
77 | YUE      | 假設      | 預測       | 1389 1520
78 | YUE      | 刺激      | 對抗       | 1658 1708
```

As you can see, there are 25 cases where the same translation was used, for example, 632 *freedom / liberty*, both translated as свобода in Russian, and 78 cases where pairs are not unique, for example, Russian плохой / ужасный, which we find in 1123 *bad / awful* and 1272 *bad / terrible*. This example shows that the Multi-SimLex approach so far does not make any use of formal testing approaches when compiling the data.

Linking Multi-SimLex to Concepticon

After several trials with the whole Multi-SimLex data, we decided to concentrate on a small subset where we had enough native speakers and multi-linguals to check the data for consistency in order to make a mapping experiment. For this experiment, we first parsed all Multi-SimLex data automatically and inferred those cases where the same

translation for Chinese (Mandarin), Russian, Spanish, and French was used for multiple occurrences of the same word in the English word pairs. Those cases where one or more languages would further distinguish the English word were placed into different rows, but automatically linked to the same Concepticon identifier, where available.

Our multilingual team (consisting of Concepticon editors Kristina Pianykh, Mei-Shin Wu, Annika Tjuka, Tiago Tresoldi, and myself) then went through all the data and tried to check if the multiple translations for the same English words were rather due to variation by near or full synonymy, or if the variation was rather due to the fact that the English words should be further disambiguated, or if other reasons could be found to explain the divergence.

We systematically marked wrong translations in our small sample, discussed many cases, and ended up by providing 2240 individually refined and unique elicitation glosses corresponding to all word pairs in the Multi-SimLex database. About 900 could be linked to the Concepticon without problems. More glosses may be linked in the future, when we expand the scope of the Concepticon further.

Representing Multi-SimLex as a Concepticon Concept List

In order to represent the complete Multi-SimLex data in the form of a concept list, we employed a format that gives each of the 2240 elicitation glosses as a row, and adds individual data for the individual languages, as well as for the word pairs in the original data, and the underlying semantic network along with the indidivual ratings.

The result is a large table with 2241 rows (the first row is our header) and many columns for individual aspects of the data. The following table provides an excerpt of the data that illustrates the basic structure for the Russian data alone.

ID	NUMBER	GLOSS	POS	CONCEPTICON ID	CONCEPTICON GLOSS	SIMLEX IDS	SIMLEX GLOSSES	LINKS	ENGLISH	RUSSIAN	ENGLISH IN SOURCE	RUSSIAN IN SOURCE	ENGLISH SCORE	RUSSIAN SCORE	RUSSIAN ERRATUM
Vulic-2020-2244-1	1	statue	nouns	1002	STATUE	203:2	sculpture	1989	statue	статуя	statue	статуя	4.92	3.9	0
Vulic-2020-2244-2	2	permit	verbs	1003	PERMIT	1443:1	approve	945	permit	разрешать	permit	разрешать	4.77	2.8	0
Vulic-2020-2244-3	3	muscle	nouns	1004	MUSCLE	1:2 690:1 696:1	arm tongue bone	306 62 172	muscle	мускул	muscle muscle muscle	мускул мускул мускул	0.69 1.62 0.154	1.1 0.8 0.6	0 0 0

Loading the Multi-SimLex Data with Python

It is important to be able to work with the data in the structure described here. For this reason, we should provide a small test or proof of concept that shows that the data in this tabular form can be parsed easily and also used for active tasks. Since there are quite a few possibilities of what can be done with the data, I decided to concentrate on the very specific case of exporting the individual word pairs for one language in the table along with the Concepticon identifiers. This task may come in handy if scholars want to test to

which degree similarities between concepts based on colexifications (François 2008, List et al. 2018) are similar to similarities between words based on user ratings for a given language.

In order to get started, we first load the `pyconcepticon` library and initiate the Concepticon API (see Tjuka 2020 for details on the installation of `pyconcepticon` and obtaining concepticon data). As the Concepticon by now (Version 2.4) has not yet been released in the version that contains the Multi-SimLex data, you need to clone the most recent version.

```
from pyconcepticon import Concepticon
concepticon = Concepticon()  # or Concepticon("path/to/concepticon")
```

To access the Multi-SimLex data, we load the respective concept list:

```
cl = {
      concept.number: concept for concept in concepticon.conceptlists[
            "Vulic-2020-2244"].concepts.values()}
```

Once this has been done, we need to extract the data and store the relevant values in a dictionary. Not in this context that the specific values stored in a concept's `attributes` are already represented as Python lists, thanks to our metadata specification for the Multi-SimLex data in Concepticon. For this reason, we can iterate over all values without having to use any specific split operations or similar.

```
msl = {}
for concept in cl.values():
   for (idx, link, eng, rus, eng_score, russ_score) in zip(
         concept.attributes["simlex_ids"],
         concept.attributes["links"],
         concept.attributes["english_in_source"],
         concept.attributes["russian_in_source"],
         concept.attributes["english_score"],
         concept.attributes["russian_score"],
      ):
      msl[idx] = [
            concept.concepticon_id or "",
            concept.concepticon_gloss or "",
             eng,
             rus,
             eng_score,
```

```
            russ_score,
            ]
```

Extracting the word pairs in Multi-SimLex fashion is now straightforward, as we have stored the words by their ID and their index. We can therefore directly write them to file. Note again, that our metadata specification in the Concepticon metadata for the Multi-SimLex concept list interprets the scores as floats, so we have to convert them to strings when writing to TSV format. In order to make sure our conversion was correct, we also check if the scores, which are represented redundantly in our concept list, are in fact identical.

```
pairs = []
with open("scores-russian.tsv", "w") as f:
f.write(
   "\t".join(
      [
      "ID",
      "CONCEPTICON_ID_1",
      "CONCEPTICON_GLOSS_1",
      "CONCEPTICON_ID_2",
      "CONCEPTICON_GLOSS_2",
      "ENGLISH_1",
      "ENGLISH_2",
      "RUSSIAN_1",
      "RUSSIAN_2",
      "ENGLISH_SCORE",
      "RUSSIAN_SCORE",
      ]
      )+"\n")
for i in range(1, 1889):
   cidA, cglA, engA, rusA, eng_scoreA, rus_scoreA = msl[str(i) + ":1"]
   cidB, cglB, engB, rusB, eng_scoreB, rus_scoreB = msl[str(i) + ":2"]
   assert rus_scoreA == rus_scoreB
   assert eng_scoreA == eng_scoreB
   f.write(
      "\t".join(
         [
            str(i),
            cidA,
            cglA,
            cidB,
```

```
            cglB,
            engA,
            engB,
            rusA,
            rusB,
            "{0:.2f}".format(eng_scoreA),
            "{0:.2f}".format(rus_scoreA),
            ]
        )+"\n"
    )
```

That is all that needs to be done, the resulting concept list `scores-russian.tsv` looks much like a single-language Multi-SimLex file, but it contains links to the concepticon. Adding the information on colexification frequency from our CLICS database (List et al. 2019) would of course also be possible, but we keep this for another blog post to be written in the future.

The code used here with some instructions is also available as a GitHub Gist, which you can find here: https://gist.github.com/LinguList/2a2fd0709fcf1b22150cfba4f2aec3fc.

References

List, Johann-Mattis and Christoph Rzymski and Tiago Tresoldi and Simon Greenhill and Robert Forkel (2019): CLICS: Database of Cross-Linguistic Colexifications. Version 3.0. Max Planck Institute for the Science of Human History. Jena: http://clics.clld.org/.

François, Alexandre (2008): Semantic maps and the typology of colexification: intertwining polysemous networks across languages. In: Vanhove, Martine (ed.): *From polysemy to semantic change*. Amsterdam:Benjamins. 163-215.

List, Johann-Mattis and Walworth, Mary and Greenhill, Simon J. and Tresoldi, Tiago and Forkel, Robert (2018): Sequence comparison in computational historical linguistics. *Journal of Language Evolution* 3.2. 130–144.

Tjuka, Annika (2020): Adding concept lists to Concepticon: A guide for beginners. *Computer-Assisted Language Comparison in Practice* 3.1. .

Vulić, Ivan and Baker, Simon and Ponti, Edoardo Maria and Petti, Ulla and Leviant, Ira and Wing, Kelly and Majewska, Olga and Bar, Eden and Malone, Matt and Poibeau, Thierry and Reichart, Roi and Korhonen, Anna (2020): Multi-SimLex: A large-scale evaluation of multilingual and cross-lingual lexical semantic similarity. *Computational Linguistics* 46.4. 847-897.