Raw data overview:

OPUS parallel corpora that are **bilingual** collections: English paired with one other language such as polish, czech, slovak and croatian in this case, respectively found in en-cs, en-pl…

Goal:

Merge these bilingual collections into multilingual collections based on the english fragment as the key.

English is the obvious pivot since it's present in every collection making it a meaningful merge key.

Input files overview:

Line N of the text file is the finished text of alignment link N. Where a link groups several source sentences together, they are already concatenated onto that single line.

This matters because the first assumption I made was to parse the xml and link each sentence to each target sentence. This assumption was proven false when i checked out 'EUbookshop.cs-en' where source unit s5 is split across two consecutive links:

row 3: cs: s5.1 s5.2 -> en:s5.1
row 4: cs: s5.3 -> en:s5.2

if the files were one line per source unit all three Czech sentences would sit on one line. So these lines have already been merged to match the English ones. We sadly can not split it up further in cases where we might both have s5.1, s5.2 respectively for english as the meaning of s5.1_english does not necessarily line up with s5.1_czech so the only option is to match line N_eng to line N_czech

Design Decisions

Links are atomic: For links such as A B ; X it is always made such that fragment A B <-> X and not A -> X or even in the case of A B ; X Y we cannot assume that A <-> X and B <-> Y without breaking semantic meaning this is down to the aligner at the end of the day.

The key is english text, not an id: This is quite obvious as s19.1 in one corpora has no link to s19.1 in another one.

The key is a hash of normalised text:
merge_key = md5(lowercase(collapse_whitespace(trim(en_text))))

Normalization is done very basic:
case, leading/trailing spaces, whitespace runs, no semantic matching. The reason i did this is so that the data stay relatively pure and as correct as possible. If we wanted to do harsher normalization all i would need to do is remerge the data which is not a problem at all.

I checked out how many more merges we would get if i dropped punctuation and it was only 460 additional keys out of 4 million (0.11%) which seems not worth it.

Processing was done bucketed
As the data is very large and exceed the servers memory capacity every row is hash partitioned into 128 buckets first. Because hashing is deterministic and always hashes to the same bucket each bucket can be merged on its own.

How the merge runs

stage 1: extract per corpus X language pair -> long format parquet file.

- read both text files, drop line pairs when blank on both sides
- parse .xml if present / .ids purely for counting column
- abort corpus if link count != line count, something went wrong there (only 1 file)
- drop links with empty english side, if only target language is empty keep the key, can still be dropped later.

stage 2: bucket

- compute merge_key, assign bucket = hash(merge_key) % 128
- write only no merge yet

stage 3: merge per bucket separately
So here it's a bit more complex, we now might have for each english key multiple candidates in each language (text repeats) so we have to choose the correct one, which is not evident. I used the following algorithm:

```
max(CASE WHEN lang = 'cs' THEN struct_pack(
is_clean := clean,
txt := lang_text,
n_en := en_n_ids,
n_tgt := lang_n_ids,
src := corpus ) END)
```

The only real choice here is to prefer clean mappings over not clean ones, meaning that one english sentence maps to one target language one instead of a combination of either if the option is present.

Results

XX_text	the translation or NULL
XX_clean	the link was 1 to 1
XX_en_n_ids/XX_n_ids	Grouping counts on each side of link
XX_corpus	which corpus the translation came from

XX_n_rows	How many sources supplied a translation
XX_n_distinct	how many different translations among them
XX_n_corpora	How many distinct corpora combined

All stored as parquet files, queried with duckDB

numbers:
514 184 046 unique english keys

Languages present

2	77.5%
3	13.1%
4	5.3%
5	2.9%
6	1.1%

Per-language coverage: cs 220.4M · pl 194.2M · sk 109.2M · hr 103.9M · sl 76.7M.

Quality

<15 chars	8.6%
15-29	21.7%
30-59	22.4%
60-119	21.9%
120+	25.4%

Meaning it’s not just key words but actual proper fragments for a large part of the database.

problems
Some corpora are the same data twice, but this is ofcourse solved by our collapse per bucket without inflating key count.

Are the translations correct?

Well, this question is a bit harder, i queried 50 lines from the top 12 largest corpora and used claude to assess if the translations were done correctly. The results can be found in the associated pdf.