Contents

Chapter 3

Compiling the Corpus

3.1 Introduction

The corpus at hand incorporates the press releases published by the Canton of Grisons/Graubünden. These press releases are a means of the cantonal government to publish news and information about topics such as politics, economy, health and culture. Graubünden, which is made up of German speaking, Italian speaking and Romansh speaking regions, is the only trilingual canton in Switzerland. As such, virtually all press releases are published in German, Italian and Romansh. This trilingual setting lends itself to be collected to a parallel trilingual corpus.

3.2 Collecting the Data

At first, I contacted the *Standeskanzlei* ("State Chancellery of Grisons") which is the "the general administrative authority for questions of office, coordination and liaison with the cantonal parliament ('Grosser Rat'), government and cantonal administration" (**staka**). The *Standeskanzlei*, with its *Übersetzungsdienst* ("Translation service"), is responsible for translating documents in service of the canton. I was hoping to receive the data directly from them – after all, we are not talking about private or commercial data, but about public translation work financed with taxpayers' money.

I spoke to Mr. Mirco Frepp from the communication sevices (*Kommunaktionsdienst*), which, although very friendly, had to inform me that it would be impossible for me to receive the data. The explanation was that the documents are not saved locally somewhere, but are saved in a database. The documents are extracted from the database and are generated as ad-hoc HTML documents whenever the website is accessed. It was also not possible to receive a dump of the database.

3.3 Web Scraping

Not being able to receive a dump of the database meant I had to scrape the canton's website, extract the relevant content from the HTML files and construct my own database. In order to achieve this, I wrote a series of Pyhton scripts that would take care of these tasks. All the scripts can be found on my GitHub/Gitlab repository. The scripts relevant for the database building are saved under the

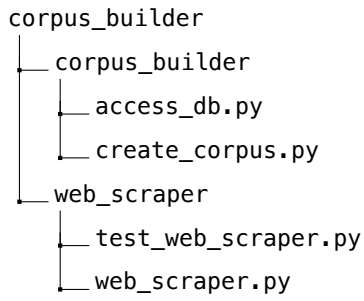

Figure 3.1: Directory tree of `corpus_builder`

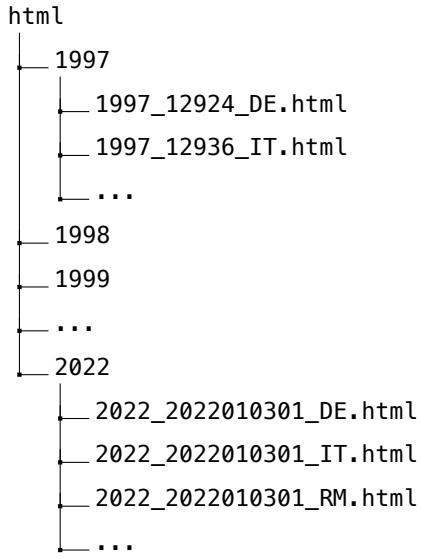

Figure 3.2: Directory scheme for saving the HTML files

folder `corpus_builder`.

Web Scraper

The script `web_scraper.py` goes to the index web page for each year and language. This page contains a links pointing to all the press releases that were released that year. It collects all those links and then downloads the HTML from each link. The HTML pages are saved to seperate folders for each year. The filenames have the following format: `year_file-id_language`, e.g., `1997_12924_DE.html`. The file-id is taken from the URL and will be later used to align the documents.

Since the script makes many requests to the website, one has to expect that server might stop responding, which will result in a request time-out. To avoid downloading HTML pages that were already downloaded, the script will skip and press realease that already exists locally, providing the file size is greater than 0 bytes. This way, the script can be run at a later stage, after new press releases were published, to complete the local repository. To make sure the local copy of the press releases is complete, the script can be simply run until a message is printed that no new press releases were downloaded.

By deafult, the script will download the press releases for the entire year range (1997 to the

current year) and in all three languages. This can be limited by using the following optional arguments:

- `--year` – limit the scraping to a year or to a range of years seperated by a comma, e.g., `--year 2022` or `--year 2020,2022`
- `--lang` – limit the scraping to one or more languages (comma seperated), e.g., `--lang de,it`

3.4 Building the Corpus

All the scripts responsible for buidling the corpus can be found under the folder `corpus_builder`.

3.4.1 HTML Parsing

After the creation of a local copy of the HTML files containing the press releases, the content needs to be extracted from the HTML files and saved in a format that would be suitable for later processing.

Using the Python package BeautifulSoup to parse the HTML files, I extracted from each HTML file the title and the text of the press release, as well as some meta data: date, language and the original file-id and the original file name (for debugging purposes). The data was then saved to a JSON file, one per each year. See listing 3.1 for an example.

```
{
    "0": {
        "id": "12924",
        "orig_file": "html/1997/1997_12924_DE.html",
        "lang": "DE",
        "title": "25 Jahre Arge Alp: Graubünden feiert tüchtig mit",
        "date": "31.12.1997",
        "content": "Die Arge Alp feiert heuer ihr 25-Jahre-Jubiläum. Aus diesem Grund
            ↪ finden vom 27. September bis 12. Oktober 1997 die Festwochen des
            ↪ Alpenraums in Telfs-Mösern, Tirol, statt. ..."
    },
    "1": {
        "id": "12926",
        "orig_file": "html/1997/1997_12926_DE.html",
        "lang": "DE",
        "title": "Kanton will auch personelles Engagement bei der Bündner Kraftwerke AG
            ↪ verstärken",
        "date": "31.12.1997",
        "content": "Nachdem der Kanton Graubünden letzten Herbst die Aktienmehrheit der
            ↪ Bündner Kraftwerke AG übernommen hat, will er nun auch seine Vertretung
            ↪ im Verwaltungsrat stärken...."
    },
    "2": {
        "id": "12927",
        "orig_file": "html/1997/1997_12927_DE.html",
```

```
21          "lang": "DE",
22          "title": "Graubünden trifft präventive Massnahmen zur Bekämpfung der illegalen
                ↪ Einwanderung",
23          "date": "31.12.1997",
24          "content": "Die Fremdenpolizei des Kantons Graubünden trifft im Einvernehmen
                ↪ mit dem kantonalen Sozialamt, dem Amt für Zivilschutz sowie der
                ↪ Kantonspolizei Graubünden Massnahmen, um die illegale Einwanderung in
                ↪ den Südtälern des Kantons Graubünden zu bekämpfen...."
25      },
```

Listing 3.1: Example for a JSON file

3.4.2 Document Alignment

After extracting the relevant data from the HTML files and saving them in JSON files, the core task can begin: aligning the documents to get document-triples which are translations of each other.

Linked vs. unlinked

For all releases published after mid-2009 this is pretty simple. The file-id extracted from the URLs is common to all three releases in the three languages (see figure 3.2). This file-id can be used to link the press releases with each other. I shall refer to these press releases as "linked releases".

For releases published prior to that, each release has a unique file-id. This means it can't be used for document alignment. I shall refer to these releases as "unlinked releases". For unlinked releases I used a simple heuristic: if on one single date exactly three releases were published in three different languages, I assume they are translations of each other. The titles of press releases that weren't aligned are saved to a CSV file which can be used for manual alignment.

Unfortunately, this means around 40% (TODO: what is the exact average?) of the releases each year prior to 2009 cannot be automatically added to the corpus, cf. figure 3.3.

Since the year 2009 contains both linked and unlinked releases, I wrote the script `split_2009.py` to split the data accordingly. It uses a very simple heuristic: if the file-id of a press release is longer than 5 digits, it is a linked press releases.

Aligned corpus

The aligned press releases are saved again to JSON files, with each row containing the three press releases in the three languages, along with metadata such as date and file-id. In the rare case that one language is missing (TODO: how rare?), i.e., the press releases wasn't translated into that language for some reason, it is simply left blank. Press releases that are available only in one language are discarded from the corpus.

The script `create_corpus.py` deals with this task. Using the Python library Pandas, the JSON files are read into a Dataframe. For linked releases, all the unique ID's are taken, and then for each ID the three languages are collected and saved into a new row. The dates are converted from their original format (DD.MM.YY) to an ISO 8601 format (YYYY-MM-DD) (**enwiki:1095673391**) for better compatibility and easier processing later.

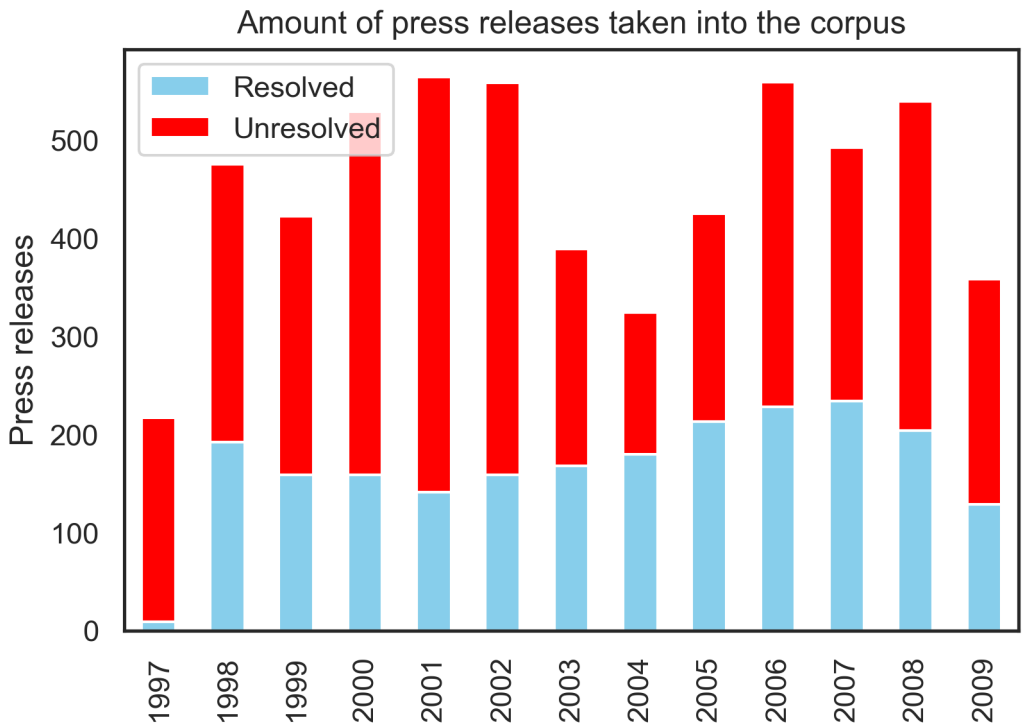

Figure 3.3: Portion of automatically aligned press releases up to 2009

For JSON files contatining unlinked documents, the script `create_corpus` has to be run with the switch `--by-date`, which tells the program to use the date for aligning the documents instead of the file ID.

For an example of the resulting JSON files, with each row containing the aligned documents, see listing 3.2.

```
{
    "0": {
        "id": "2010010501",
        "date": "2010-01-05",
        "DE_title": "Neues Online-Angebot für das Bündner Rechtsbuch",
        "DE_content": " Das im Internet verfügbare Bündner Rechtsbuch ist neu gestaltet
            ↪ worden und enthält neue Funktionalitäten. ...",
        "IT_title": "Nuova offerta online per la Collezione sistematica del diritto
            ↪ cantonale grigionese",
        "IT_content": " La Collezione sistematica del diritto cantonale grigionese
            ↪ disponibile in internet è stata ristrutturata e contiene nuove funzioni.
            ↪ ... ",
        "RM_title": "Nova purschida d'internet per il cudesch da dretg grischun",
        "RM_content": " Il cudesch da dretg grischun che stat a disposiziun en
            ↪ l'internet ha survegnì in nov concept e novas funcziuns. ... "
    },
    "1": {
        "id": "2010010502",
        "date": "2010-01-05",
```

```
        "DE_title": "Staupe bei Füchsen und Dachsen im Puschlav",
        "DE_content": " Nachdem sich im Verlaufe des letzten Herbstes die
            ↪ Staupe-Krankheit bei Wildtieren in Nord- und Mittelbünden verbreitete,
            ↪ sind im Laufe der letzten Wochen nun auch im Puschlav bei Füchsen und
            ↪ Dachsen Infektionen mit dem Staupevirus nachgewiesen worden. ... ",
        "IT_title": "Volpi e tassi affetti da cimurro in Valposchiavo",
        "IT_content": " Dopo che nel corso dell'autunno il cimurro si è diffuso tra gli
            ↪ animali selvatici del Grigioni settentrionale e centrale, nelle ultime
            ↪ settimane la presenza del virus è stata rilevata anche tra volpi e tassi
            ↪ della Valposchiavo. ...  ",
        "RM_title": "Pesta dals chauns tar vulps e tar tass en il Puschlav",
        "RM_content": " Suenter che la pesta da chauns è sa derasada tar la selvaschina
            ↪ dal Grischun dal nord e central en il decurs da l'atun passà, èn
            ↪ vegnidas cumprovadas en il decurs da las ultimas emnas ussa er
            ↪ infecziuns cun il virus da questa malsogna tar vulps e tar tass en il
            ↪ Puschlav. ... "
    },
    "2": {
        "id": "2010010801",
        "date": "2010-01-08",
        "DE_title": "Projekt Sicherheitsfunknetz POLYCOM Graubünden mit
            ↪ Vertragsunterzeichnung offiziell gestartet",
        "DE_content": " Die Vorsteherin des Departements für Justiz, Sicherheit und
            ↪ Gesundheit, Regierungsrätin Barbara Janom Steiner, und der Chef des
            ↪ Grenzwachtkorps, Jürg Noth, haben heute in Chur eine Vereinbarung zur
            ↪ Realisierung des Sicherheitsfunknetzes POLYCOM im Kanton unterzeichnet.
            ↪ ... ",
        "IT_title": "Avviato ufficialmente con la sottoscrizione del contratto il
            ↪ progetto di rete radio di sicurezza POLYCOM Grigioni",
        "IT_content": " La Consigliera di Stato Barbara Janom Steiner, direttrice del
            ↪ Dipartimento di giustizia, sicurezza e sanità, e il capo del Corpo delle
            ↪ guardie di confine, Jürg Noth, hanno sottoscritto oggi a Coira un
            ↪ accordo per la realizzazione nel Cantone della rete radio di sicurezza
            ↪ POLYCOM. ... ",
        "RM_title": "Il project per la rait radiofonica da segirezza POLYCOM dal
            ↪ Grischun è vegnì lantschà uffizialmain cun suttascriver il contract",
        "RM_content": " La scheffa dal departament da giustia, segirezza e sanadad,
            ↪ cussegliera guvernativa Barbara Janom Steiner, ed il schef dal corp da
            ↪ guardias da cunfin, Jürg Noth, han suttascrit oz a Cuira ina cunvegna
            ↪ per realisar la rait radiofonica da segirezza POLYCOM en il chantun. ...
            ↪ "
    },
}
```

Listing 3.2: Example for a JSON file containing aligned documents

3.5 Manual alignment of unlinked documents

As can be seen in figure 3.3, a big portion of the unlinked documents cannot be automatically aligned using the simple heuristic described in section 3.4.2. To deal with that, the script `create_corpus.py` will write the titles and file IDs of all the discarded documents to a CSV file, one for each year.

These CSV files can be used for manually aligning the corresponding documents. This is done by enumerating the documents while using the same digit for corresponding documents. The CSV files containing the now enumerated documents can be given to the script `create_corpus.py` using the argument `add-from-csv` to combine the enumerated documents as linked documents into the JSON file.

3.6 SQLite database

The query language SQL offers flexible and complex way to query datbases. For this reason, I decided to save the resulting corpus in an SQLite database. I opted for SQLite because it doesn't require running a seperate server and can be SQLite databases can be easily built, edited and accessed using `sqlite3`[1], a Python module delivered in the Python standard library[2].

The SQLite database contains two tables, `corpus` and `raw` with the exact same structure as the two JSON files described in listings 3.1 and 3.2.

3.7 Summary

For compiling the corpus, the following steps were taken:

1. Scrape website and save HTML documents locally
2. Extract relevant content from HTML files (date, language, title and content) and save to JSON files
3. Read the JSON files using Pandas Dataframe, align the documents and save to new JSON files
4. Feed both types of JSON files to an sqlite database

The final result is an sqlite database (`corpus.db`) containing two tables:

- `corpus`: all the aligned documents from 1997 until today. Each row contains following columns:
  - id: automatically incremented unique ID
  - file_id: original file ID
  - date: Release date
  - DE_title: Title of German document

[1] `https://docs.python.org/3/library/sqlite3.html`

[2] `https://docs.python.org/3/tutorial/stdlib.html`

Year	Documents	Year	Documents
1997	3	2010	184
1998	64	2011	167
1999	53	2012	207
2000	53	2013	219
2001	47	2014	218
2002	53	2015	183
2003	56	2016	190
2004	60	2017	207
2005	71	2018	221
2006	76	2019	216
2007	78	2020	286
2008	68	2021	294
2009	109	2022	153

Table 3.1: Number of parallel documents per year, as of July 20, 2022.

  - DE_content: Content of German document
  - IT_title: Title of Italian document
  - IT_content: Content of Italian document
  - RM_title: Title of Romansh document
  - RM_content: Content of Romansh document

- `raw`: all the documents contained in the HTML files scraped from the website. Each row contains the following columns:
  - id: Automatically incremented unique ID
  - file_id: Original file ID
  - orig_file: Original filename
  - lang: Document language (DE for German, IT for Italian, RM for Romansh)
  - title: Document title
  - date: Release date
  - content: Document content

3.7.1 Statistics

The corpus contains 3,536 parallel documents.

Glossary

Graubünden The Canton of Grisons. 1, 5

Acronyms

AER Average Error Rate. 33, 41, 43, 44, 46, 53

NER Named Entity Recognition. 2

POS Part of Speech. 2

List of Tables

List of Figures