Universität Zürich[UZH]

Masterarbeit

zur Erlangung des akademischen Grades

Master of Arts

der Philosophischen Fakultät der Universität Zürich

Using Multilingual Word Embeddings for Similarity-Based Word Alignments in a Zero-Shot Setting

Tested on the Case of German–Romansh

Verfasser: Eyal Dolev

Matrikel-Nr: 20-713-897

Referent: Prof. Dr. Martin Volk

Institut für Computerlinguistik

Abgabedatum: 01.01.2023

Abstract

Using multilingual word embeddings for computing word alignments has been shown to be competetive with statistical word alignment methods. However, the languages on which the experiments were made on were all "seen" languages, i.e., they were part of the training data for the embeddings. In this thesis I show that multilingual word embeddings taken from mBERT can be used for computing word alignments for the "unseen" language Romansh, aligned against German. The performance is on par with a baseline statistical model (fast_align). The thesis also describes the creation of a gold standard for evaluating the quality of word alignments for German–Romansh. This thesis additionally describes the process of data collection for compiling a trilingual corpus containing press releases in German, Italian and Romansh, published by the Swiss Canton of Grisons. From this corpus, I extracted around 80,000 unique sentence pairs for each language combination.

Acknowledgements

I would like to thank... wird ergänzt

Contents

Chapter 1

Introduction

1.1 Motivation

Romansh is a Romance language spoken in Switzerland, primarily in the Canton of Grisons (henceforth Graubünden) (Bossong 1998, p. 173). Graubünden is the only canton in Switzerland with three official languages—German, Italian and Romansh. The number of Romansh speakers, 40,000, has been decreasing in the last decades (Bundesamt für Statistik 2020). In order to protect Romansh from extinction, Graubünden committed itself in its constitution to the protection and the promotion of multilinguality within its borders:

> Kanton und Gemeinden unterstützen und ergreifen die erforderlichen Massnahmen zur Erhaltung und Förderung der rätoromanischen und der italienischen Sprache[1] . (Art. 3 Abs. 2 der Bündner Verfassung[2])

Additionally, in 2006 a language law (*Sprachengesetz*) was passed, with the aim of further promoting and protecting the multilinguality of the canton:

> Dieses Gesetz bezweckt: ... e) die bedrohte Landessprache Rätoromanisch mit besonderen Massnahmen zu unterstützen[3]; (Abs. 1 Art. 1 Bst. e des Sprachengesetz des Kantons Graubündens[4])

Since 1998, the majority of all press releases published by the Canton Graubünden were released in these three languages. Such parallel documents in three languages lend

[1] The canton and the communities shall support and take the required measures to maintain and promote the Romansh language and the Italian language.

[2] https://www.gr-lex.gr.ch/app/de/texts_of_law/110.100

[3] The law of languages of the Canton Graubünden is meant to: e) to support the endangered national language Romansh.

[4] https://www.gr-lex.gr.ch/app/de/texts_of_law/492.100#structured_documentingress_foundation_fn_4417_2_2_c

themselves to the collection and the compilation of a trilingual parallel corpus. Of special interest is here the Romansh language, which, having such a low number of speakers and due to the fact that not many natural language processing (NLP) resources exist (more on that later), should be seen as a "low-resource language".

1.2 Research Question and Goals

1.2.1 Research Question

Given two sentences which are mutual translations, word alignment is a mapping of the words in the sentence of the source language to the words in the sentence of the target language (Koehn 2009, p. 84). Jalili Sabet et al. 2020 were able to show that their algorithm for word alignment (SimAlign), which is similarity-based and uses multilingual word embeddings to compute similarity, outperforms statistical models.

But not only that the model outperforms the existing statistical models, its biggest advantage, as propagated by Jalili Sabet et al. 2020, is that it requires no parallel training data (pairs of sentences which are mutual translations), but only monolingual training data— statistical models will only reach good performance with enough parallel training data (Jalili Sabet et al. 2020; Och and Ney 2000). Using word embeddings, words in just one single sentence pair can be aligned with high accuracy, without the need of a large set of sentence pairs for first training a word alignment model. However, all of this works presuming we already have a multilingual language model, trained on monolingual data, whose learned embeddings we can leverage for this task. There exist some language models that were trained on multilingual data: mBERT was trained on 104 languages[5], LASER was trained on 93 languages (Artetxe and Schwenk 2019) and XLM-RoBERTa base was trained on 100 languages (Conneau et al. 2020). Romansh, however, is not part of any of the training data for these models.

Multilingual language models were shown to also perform well in various tasks on unseen languages, dubbed as "zero-shot setting". mBERT achieves reasonable results out-of-the-box (without further training) on unseen languages in a variety of tasks such as named entity recognition (NER) and part of speech (POS) tagging (Pires, Schlinger, and Garrette 2019). And although the LASER model was pretrained on 93 languages, it obtained strong results for sentence embeddings in 112 languages (Artetxe and Schwenk 2019).

There is, thus, good reason to believe that similarity-based word alignment using multilingual word embeddings would work also for the case of German–Romansh or Italian–Romansh, in spite of Romansh not being part of the training data, especially since vocabu-

[5] https://github.com/google-research/bert/blob/master/multilingual.md

lary overlaps between unseen and seen languages favor performance in zero-shot settings (Pires, Schlinger, and Garrette 2019), and since Romansh displays a high similarity with other seen Romance languages, e.g., Italian, French, Spanish. English, although not a Romance language, also has a large portion of Romance-based vocabulary.

The research question at hand is therefore: **Will similarity-based word alignment perform as well as statistical word alignment models for the language pair German-Romansh?**

1.2.2 Goals

My goals for this thesis are twofold:

- Test whether similarity-based word alignment using multilingual word embeddings will perform on par with statistical word alignment models on Romansh;
- Collect the press releases of the canton Graubünden, published in German, Romansh and Italian, and compile a parallel trilingual corpus.

To test the quality of the word alignments, I will create a gold standard and manually annotate word alignment for German-Romansh sentence pairs.

After finishing my work, I will make my gold standard and the corpus I compiled available for further research by future students.

1.3 Structure

In the course of the following pages I will first give a short introduction to the Romansh language (Chapter 2), then describe how I collected the data and aligned the documents (Chapter 3) and how I further aligned the sentences to extract sentence pairs (Chapter 4). I will shortly explain the mechanism behind statistical and similarity-based word alignment methods (Chapter 5). Finally, I will explain how and according to which guidelines I created the gold standard (Chapter 6) and display the results of my experiments in which I compared different word aligning systems (Chapter 7).

Throughout this work, I went to effort to not become too technical in details, always writing to an imaginary fellow student of linguistics, such that this work, if it ever falls in the hands of a future student, will be comprehensible and readable. I hope that it *will* be read by and inspire future students, in the same way I that was inspired by works written by students before me.

1.4 GitHub repository

The code I wrote and the data I collected in the course of this work is available on my GitHub repository. Please contact me in order to gain access to it.

Chapter 2

Romansh

In this chapter, I will provide a short context about Romansh, the language that is a third of the resulting corpus and conceptually the main motivation for this work.

2.1 Rhaeto-Romance

In 1873, an Italian linguist by the name of Graziadio Ascoli pointed out a shared number of characterizing phenomena in a number of Romance dialects spoken in parts of Switzerland and Italy (but without a geographical continuum) and named this group of dialects "Ladino". Since 1883, due to the influence of the Austrian linguist Theodor Gartner's publication *Raetoromanische Grammatik* describing this group of dialects, this name (German *Rätoromanisch*, English "Rhaeto-Romance") became associated with this group of dialects.

Rhaeto-Romance is spoken in three areas, separated from each other, and is made up of three super-dialects: Romansh, spoken in parts of the Swiss canton of Grisons (Graubünden), Ladin, spoken in the Dolomotic Alps in northern Italy (Südtirol), and Friualian, spoken around the drainage basin of the Tagliamento river, between Venice and Trieste (Haiman and Benincà 1992, p. 1).

There have been long discussions in Romance linguistics about whether Rhaeto-Romance can be seen as a unity of dialects, or whether such a unity is merely a linguistic construct, lacking a socio-linguistic and historical basis. This dispute is referred to as the *questione ladina* ("the Ladin question") (Liver 1999).

Ascoli, the grounder of the idea of a Rhaeto-Romance unity, made his classifications at a time when language researchers were fascinated by the regularity of sound changes. At the time, common historical sound changes were used as the main means to group languages and dialects together. Ascoli therefore based his grouping of these three dialects on the grounds of sound changes common to all three dialects. His followers propagate a narrative according to which the three dialects once occupied one geographical area,

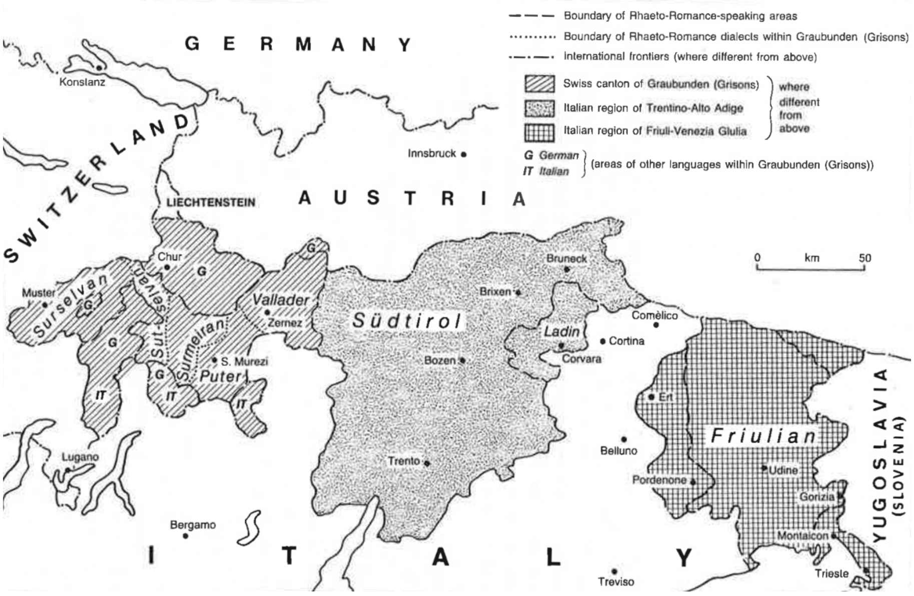

Figure 2.1: Distribution of Rhaeto-Romance, taken from Haiman and Benincà 1992, p. 2

but were separated by the Germanic incursions in the years CE 250-800 (Bossong 1998, p. 174; Haiman and Benincà 1992, p. 11).

An opposing group of researchers believes that the three Rhaeto-Romance dialects show decisive features common to their respective neighboring Italian dialects. They should therefore be classified as north-Italian dialects and be seen as parts of the Italian dialect continuum (Bossong 1998, p. 174).

This question, as interesting as it may be, is not of importance to this thesis and will not bother us for the rest of it. It is nonetheless important to remember that names and definitions posed by researchers are never as simple as they might seem, nor do they always correspond to the feelings of the speakers and their own sense of identity. In the case of Rhaeto-Romance, the speakers of these dialects do not feel as though they all belong to some greater unity (Bossong 1998, p. 175).

2.2 Romansh

The term Romansh is a collective name referring to the Rhaeto-Romance dialects spoken in Switzerland and are recognized as a single language. There are five different dialects (Surselvan, Sutselvan, Suermiran, Puter, Vallader), each having normative grammars and distinct orthographic norms (motivated by the Reformation, for translating the Bible and other religious texts) (Haiman and Benincà 1992, p. 1; Bossong 1998, p. 178).

Romansh was officially acknowledged as a fourth official language in Switzerland (besides German, French and Italian) in a federal referendum that took place in 1938, in the eve of the Second World War, with a whopping majority of 92% Yes votes. It has been hypothesized that this referendum played in the hands of the Rhaeto-Romans in Graubünden to promote their nationalistic political postulate, but was also instrumentalised by the Swiss federal government to counteract Mussolini's pretenses to "Italian" territories in Switzerland (referred to as the Italian irredentism[1]) (Valär 2012).

Romansh is currently spoken by around 40,000 people (Bundesamt für Statistik 2020). This number has been diminishing constantly—30 years ago there were 50,000 speakers (Haiman and Benincà 1992). There is however hardly a single person who speaks only Romansh. In Switzerland, as in the other regions of Rhaeto-Romance, there is always a "prestige" language surrounding Rhaeto-Romance, in which Rhaeto-Romance speakers are fluent in (Haiman and Benincà 1992, p. 3).

2.3 Rumantsch Grischun

2.3.1 Lia Rumantscha

In the past hundred years there has been a Rhaeto-Romance revival. In Switzerland, a major force in this language movement was the founding of the Lia Rumantscha ("The Romansh League") in 1919, which was also a counter-force to the Italian irredentism[1]. It is an umbrella organization devoted to promoting and perserving the Rhaeto-Romance language and culture. Its goals include creating and promoting a common language awareness and identity among the Rhaeto-Romans. The organization is responsible for developing a language standard, as well as for language renovation, and generally representing the interests of the Romansh and its speakers, in Graubünden and in the Swiss diaspora (Dazzi 2012).

2.3.2 Rumantsch Grischun

The endeavors of the Lia Rumantscha in the field of language planning and standardization led to the official launching of a pan-Romansh language—*Rumantsch Grischun* (Haiman and Benincà 1992, p. 5). Its goal was not to replace the local dialects, but be available for persons, institutions, government agencies, companies etc., that want to use Romansh but require a language variant that would be inter-regional and intelligible by speakers of all dialects. The main motivation for planning an inter-regional standard was the failure of Romansh to establish itself as a fourth national language due to the lack of a written standard, despite the great willingness of the people. The existence of a written standard was intended to make Romansh be better respected and incorporated in the canton of Graubünden, as well as on a federal level; it would also elevate its prestige in the eyes of its speakers (Schmid 1982).

[1] The nationalistic claim of lands inhabited by persons who the Italian nationalists saw as ethnic Italians.

2.3.3 Features

Rumantsch Grischun was suggested in 1982 by the Zurich-born Romance linguist Heinrich Schmid. It was, however, not the first attempt to harmonize the Romansh dialects. In the 19th century, a high school teacher named Gion Antoni Bühler, made failed attempts to propagate for a *Romonsch fusionau*; in the 1960's, a Swiss author from the canton of Graubünden, Leza Uffer, suggested *Interrumantsch*, which was mainly based on the Surmiran dialect, but failed similarly (Liver 1999, p. 39).

Rumantsch Grischun's success has been hypothesized to be mainly due to the favorable timing—the socioeconomical situation at the time as well as a change in the approach of many Rhaeto-Romans to their own language; but also due to the fact that Rumantsch Grischun, contrary to previous suggestions for a standard language, is more consistent and balanced between the dialects (Liver 1999, p. 69). It never systematically favors one dialect over the other.

Without going too much into detail, Rumantsch Grischun favors the greatest common denominator by taking the word forms common to the three most important written dialects (Sursilvan, Surmiran and Vallader). For instance, in all three dialects the word for "key" is *clav*, hence, this is also the Rumantsch Grischun word for "key". In case the dialects do not agree, the word form common to the majority of dialects is taken, in a sort of "majority vote". That way, one dialect over is never preferred over the others throughout.

Clarity and transparency also play a major role. This means that forms which exhibit stem alternations, for instance between singular and plural, are abandoned in favor simpler, more regular forms. Further, phenomenons that are specific to just one dialect are left out, such as the rounded front-vowels [y] and [ø] typical of the dialects of the Engadine, or the closing diphthong [ɪw][2], which is unique to Sursilvan (Liver 1999, p. 70). See table 2.1 for some examples.

This new language fulfills the requirements of its authors: it can be read and understood by any Rhaeto-Roman without them having to elaborately learn it and the differences to the specific dialects are minimal (Liver 1999, p. 72).

2.3.4 Today

Rumantsch Grischun has become one of the most ambitious endeavours in the history of Romansh. Since its invention, Romansh and the people promoting it have had notable success achieving their goals. In 1999, Romansh became a "partially official language" (*Teilamtssprache*) of the Swiss confederation. In 2003, it was recognized in the cantonal constitution of Graubünden as an equal cantonal language, and the protection of the traditional language regions was guaranteed. Nowadays, Romansh is in use in many domains,

[2]The diphthong starts with an open vowel [ɪ] and ends with a closed vowel [w], hence "closing"

Sursilvan	Surmiran	Vallader	Rumantsch Grischun	Principle
clav	**clav**	**clav**	**clav** "key"	Greatest common denominator
tschiel	**tschiel**	tschel	**tschiel** "sky"	Majority vote
siat	**set**	**set**	**set** "seven"	"
cor	**cor**	cour	**cor** "heart"	"
vendiu	vendia	vendü	**vendi** "bought"	Favor simplicity
sg./pl. *iert/orts*	**iert/ierts**	üert/üerts	iert/ierts "garden"	"

Table 2.1: Examples for choosing the forms for Rumanstch Grischun, based on Liver 1999, pp. 70–71

not only in the public administration, but also in economy. Many works were written in Rumantsch Grischun. People learn to read and write in Rumantsch Grischun and in some schools, classes are held in it. The extent of radio and television in Romansh has been growing. There is a radio station broadcasting 24/7, television programs in Romansh are broadcast in all public channels of the Swiss Broadcating Corporation (SSG SSR), and there are also internet portals, e.g., https://www.rtr.ch/. All of this wouldn't have been possible if it weren't for the political "upgrade" that was aspired for by the Romansh language movement (Cathomas 2012).

The canton of Graubünden has been releasing most or all of its press releases since 1998 in three languages: German, Italian and Romansh using the Rumantsch Grischun standard. I therefore decided to collect these press releases and use them to compile a parallel corpus.

From this point on, the term *Romansh* will refer to the standard variant *Rumantsch Grischun*.

2.4 Romansh in NLP

2.4.1 Low-resource languages

The field of natural language processing (NLP) relies on the existence of digital language resources, such as collections of written or spoken texts, or a gold standard with labels of the desired output of a system. There is a dichotomy in the field of NLP between high-resource languages and low-resource languages. High-resource languages, such as English and Chinese, have large accessible amounts of digitized texts and annotated data, but also off-the-shelf working tools for various NLP tasks (POS taggers, named entity recognizers) (Bender 2019).

The term low-resource refers to a variety of scenarios and there is no clear definition of what a low-resource language is. It may refer to endangered languages with a low number of speakers, but also to widely spoken languages which are seldom addressed by the NLP community. There are also different thresholds of amounts of data for defining a language as "low-resource" (Hedderich et al. 2021). As the case may be, "low-resource language" means the amount of digital resources available for that language are scarce in comparison to high-resource languages.

2.4.2 Romansh as a Low-Resource Language

Although Romansh is an endangered language with an ever diminishing number of speakers, it did receive some attention from the NLP community. One could say that Romansh "got lucky": it has a written standard, it is spoken in a highly-modernized country, it is promoted and protected by law, and last but not least, the Universities of Zurich and Geneva have departments dedicated to NLP, whose attention was often drawn towards Romansh.

I was indeed not the first person to have the idea of collecting parallel data including Romansh. Scherrer and Cartoni 2012 also created a trilingual corpus using the press releases published by the canton of Graubünden[3]. Weibel 2014 compiled two sentence- and word-aligned corpora (German-Romansh) based on legal texts and on the same press releases, and made them available on multilignwis, an online concordance search system (Graën, Sandoz, and Volk 2017). Last year, another student at the University of Zurich collected parallel data in German and Romansh as part of a seminar dealing with Rhaeto-Roman culture.

Romansh was also used for evaluating performance of out-of-domain machine translation[4] (Müller, Rios, and Sennrich 2020) or for evaluating code-switching detection within a multilingual corpus (Volk and Clematide 2014).

Most recently, TextShuttle, a Zurich-based company specializing on machine translation,, developed and released a machine translation system for Romansh (translating to or from German, French, Italian and English) (TextShuttle AG 2022).

Although Romansh is in a better situation than other low-resource languages, collecting more data and running experiments with it, especially in a zero-shot setting using multilingual language models (cf., Section 1.2.1 and Section 7.4), is worthwhile.

[3]However, only until 2012; The corpus was then used for the task of induction of bilingual lexicons

[4]Translating texts of unseen domains

Chapter 3

Compiling the Corpus

3.1 Introduction

The corpus at hand incorporates the press releases published by the canton of 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 these three languages. 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" (Standeskanzlei Graubünden 2022). 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, this is not private or commercial data, but public translation work financed with taxpayers' money.

I spoke to Mr. Mirco Frepp from the communication services (*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 rather 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 Python scripts that would take care of these tasks. All the scripts can be found on my GitHub repository[1]. The scripts relevant for the database building are saved under the folder `corpus_builder`.

Web Scraper

The script `web_scraper.py` goes to the index web page for each year and language. This page contains the links pointing to all the press releases that were released that year. It collects all those links, and then downloads the HTML file from each link. The HTML pages are saved in separate folders for each year. The filenames are saved using 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.

```
html
├── 1997
│   ├── 1997_12924_DE.html
│   ├── 1997_12936_IT.html
│   └── ...
├── 1998
├── 1999
├── ...
└── 2022
    ├── 2022_2022010301_DE.html
    ├── 2022_2022010301_IT.html
    ├── 2022_2022010301_RM.html
    └── ...
```

Figure 3.1: Directory scheme for saving the HTML files

Since the script makes many requests to the website, one has to anticipate that the server might stop responding, which will result in a request time-out. This means the script will have to be run a couple of times. To avoid downloading HTML pages that were already downloaded, the script will skip any press releases that already exist locally, providing the file size is greater than 0 bytes. This way, the script can also be run at a later stage, after additional new press releases were published, in order to update the local repository.

[1] https://github.com/eyldlv/de_rm_it_corpus

To make sure the local copy of the press releases is complete, the script can simply be run repeatedly until a message is printed to the console that no new press releases were downloaded.

By default, 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 separated by a comma, e.g., `--year 2022` or `--year 2020,2022`
- `--lang` – limit the scraping to one or more languages (comma separated), e.g., `--lang de,it`

3.4 Building the Corpus

All the scripts responsible for building 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 text containing the press releases 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[2] 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[3] file, one file per year. See listing A.1 on page 75 for an example.

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 mutual translations.

Linked vs. Unlinked

For all releases published after mid-2009, document alignment is pretty simple. The file ID extracted from the URLs is **common** to all three releases in the three languages (see

[2] `https://beautiful-soup-4.readthedocs.io/en/latest/`

[3] JavaScript Object Notation (JSON) is one of the most popular formats for organizing text data in a hierarchical form. Its syntax is almost identical with that of Python list and dictionaries (Kofler 2019, p. 279).

example under the folder 2022 in Figure 3.1). 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** URL, hence also a unique file ID. This means it cannot 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.

Unfortunately, this means more than half of the of the releases in the years prior to 2009 cannot be automatically added to the corpus, cf. Figure 3.2.

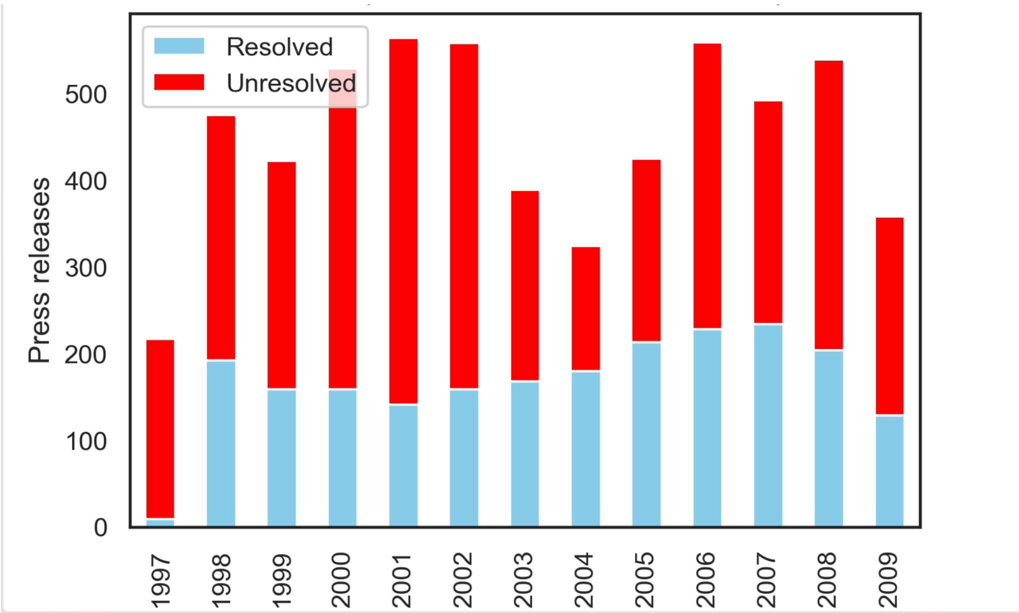

Figure 3.2: Portion of automatically aligned press releases up to 2009. “Resolved” are releases the were added to the corpus according to the heuristic described in Section 3.4.2 (exactly three on one date of three different languages).

Since the year 2009 contains both “linked” and “unlinked” releases, the script `split_2009.py` will 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 entry in the file 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, i.e., a press release 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 aligned corpus.

The script `create_corpus.py` deals with this task. Using the Python library Pandas[4],

[4]https://pandas.pydata.org

the JSON files are read into a DataFrame (a two-dimensional, table-like data structure). For linked releases, all the unique ID's are queried, 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) (Wikipedia contributors 2022) for better compatibility and easier processing later.

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

For an example of the resulting JSON files, with each row containing the aligned documents, see Listing A.2 on page 76.

3.5 SQLite database

The query language SQL offers flexible and complex ways to query databases. 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 separate server and SQLite databases can be easily built, edited and accessed using `sqlite3`[5], a Python module delivered with the Python standard library[6].

The SQLite database contains two tables, `corpus` and `raw` with the exact same structure as the two JSON files described in Listings A.1 and A.2.

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

- `corpus`: All the aligned documents from 1997 until today. See Table 3.1 for details.
- `raw`: All the documents contained in the HTML files scraped from the website. See Table 3.2 for details.

This way, fast and efficient corpus queries can be made. For instance, the following query will find all the German press releases and their Italian translations from the year 2021 containing the word *Umwelt* ("environment") that are at least 5000 characters long:

```
SELECT DE_title, DE_content, IT_title, IT_content
FROM corpus
WHERE DE_content LIKE "%Umwelt%"
AND LENGTH(DE_content) > 5000
```

[5] https://docs.python.org/3/library/sqlite3.html
[6] https://docs.python.org/3/tutorial/stdlib.html

Column	Description
id	Automatically incremented unique ID
file_id	Original file ID
date	Release date
DE_title	Title of German document
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

Table 3.1: Description of the table corpus in corpus.db

Column	Description
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

Table 3.2: Description of the table raw in corpus.db

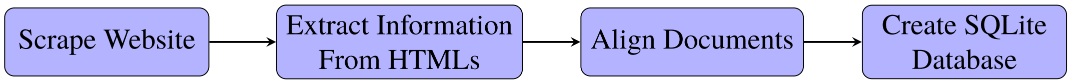

Figure 3.3: Corpus creation pipeline

3.6 Summary

For compiling the corpus, the following steps were taken (see also Figure 3.3):

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

3.6.1 Statistics

The corpus contains 3,536 parallel documents, with a yearly average of 56.8 documents prior to 2009 and a yearly average of 207.8 documents from 2009 onward[7], see also Table 3.3. Table 3.4 breaks down the number of documents per each year and language.

The unaligned corpus contains 2,484,250 German tokens, 2,760,690 Romansh tokens and 2,581,168 Italian tokens[8]. Table 3.5 displays the 20 most frequent tokens for each language in the corpus.

[7]Not including 2022

[8]Including punctuation tokens

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
		Total	**3,536**

Table 3.3: Number of parallel documents per year, as of July 20, 2022.

Year	German	Romansh	Italian
1997	181	17	18
1998	168	153	153
1999	161	130	130
2000	192	167	169
2001	233	159	171
2002	235	157	165
2003	167	110	111
2004	132	97	94
2005	157	134	133
2006	211	173	174
2007	199	147	145
2008	201	168	169
2009	212	175	176
2010	219	183	184
2011	203	167	167
2012	254	207	207
2013	260	219	219
2014	260	218	218
2015	227	183	183
2016	221	190	190
2017	236	207	207
2018	248	221	220
2019	238	216	216
2020	310	284	285
2021	322	294	294
2022	169	153	153
Total	**5616**	**4529**	**4551**

Table 3.4: Number of documents per language and year as of 20 July, 2022.

German		Romansh		Italian	
Type	Count	Type	Count	Type	Count
Graubünden	17859	Grischun	16775	Governo	15592
Regierung	15557	regenza	14683	Grigioni	15337
Kanton	8554	chantun	10928	Cantone	10402
Franken	7400	davart	8917	franchi	6594
Bündner	7375	chantunala	7668	cantonale	6494
Gemeinden	5638	francs	7264	progetto	5966
Quelle	5027	persunas	6285	essere	5918
Gremium	4940	fin	6053	viene	5623
Standeskanzlei	4050	vischnancas	5983	Stato	5476
Amt	3807	project	5917	legge	5381
Chur	3640	lescha	5652	comuni	5198
genehmigt	3533	l'onn	5176	Consiglio	5116
Gemeinde	3324	grond	4954	Organo	4309
Grossen	3264	cussegl	4799	revisione	4162
Tel	3224	scola	4272	Fonte	4114
Jahr	3222	revisiun	4215	federale	4059
betreffend	3211	Funtauna	4106	Gran	3941
rund	3052	grischuna	4042	nonché	3916
Kantons	3013	Gremi	4029	grigionese	3894
wurde	2902	construcziun	3993	protezione	3754

Table 3.5: Twenty most frequent tokens in each language in the corpus, excluding punctuation and stop words. Stop words lists for German and Italian taken from NLTK (Bird, Loper, and Klein 2009)

Chapter 4

Sentence Alignment

4.1 Introduction

The corpus presented in chapter 3 is a raw parallel corpus, that is, it is a corpus of aligned documents without any further processing. In order to use the corpus for tasks such as training a machine translation model, another processing step is needed: sentence alignment (Koehn 2009, p. 55).

A bilingual, sentence-aligned corpus can be useful for a variety of tasks. Bilingual corpora are probably mostly used for training a machine translation model (Gale and Church 1991; Moore 2002; Chen 1993), but they can also be used for building translation memories (Sennrich and Volk 2011) or a for bilingual concordance systems, with the purpose of allowing a user to find out how a given sentence is translated (Moore 2002; Gale and Church 1991), e.g., multilingwis[1] (Graën, Sandoz, and Volk 2017).

4.1.1 Formal definition

Formally, the task of sentence alignment can be described as follows: We have a list of sentences in language e, $e_1, ...e_{n_e}$ and a list of sentences in language f, $f_1, ..., f_{n_f}$. (Note that n_e the number of sentences in language e, is not necessarily identical to n_f the number of sentences in language f.) A sentence alignment S consists of a list of sentence pairs $s_1, ..., s_n$, such that each sentence pair s_i is a pair of sets:

$$s_i = (\{e_{\text{start-e}(i)}, ..., e_{\text{end-e}(i)}\}, \{f_{\text{start-f}(i)}, ..., f_{\text{end-f}(i)}\})$$

(Koehn 2009, p. 56)

This means that each set in this pair of sets can consist of one or more sentences. The number of sentences in each set is referred to as *alignment type*. A 1-to-1 alignment is an alignment where exactly one sentence of language e is aligned to exactly one sentence of

[1] https://pub.cl.uzh.ch/projects/sparcling/multilingwis2.demo/

language f. In a 1-to-2 alignment, one sentence in language e is a aligned to two sentences in language f. There are also 0-to-1 alignments, in which a sentence of language f is not aligned to anything of language e. Sentences may not be left out and each sentence may only occur in one sentence pair (Koehn 2009, p. 57).

4.2 Method Overview

Traditionally, there are three main approaches for solving the problem of sentence alignment: length-based, dictionary- or translation-based and partial similarity-based (Varga et al. 2005).

4.2.1 Length-Based

One early method for sentence alignment is "based on a simple statistical model of character lengths" (Gale and Church 1991). The method, dubbed since as the "Gale & Church method/algorithm", arose out of the need to design a faster, computationally more efficient algorithm than the ones that existed at the time[2].

The Gale & Church method is based on the assumption that longer sentences in language e are usually translated into longer sentences in language f and vice-versa—shorter sentences in one language correspond to shorter sentences in the other language.

The method combines a distance measure based on the lengths of the sentence with a prior probability of the alignment type (1-to-1; 1-to-0 or 0-to-1; 2-to-1 or 1-to-2; 2-to-2) to a probabilistic score. It assigns this score to possible sentence pairs in a dynamic programming framework to find the best (most probable) pairs (Koehn 2009, p. 57).

A program based on this method was tested against a human-made alignment on two pairs of languages: English-German and English-French. The program made a total of 55 errors out of a total of 1,316 alignments (4.2%). By taking the best-scoring 80% of the alignments, the error rate was able to be reduced to 0.7%. The method was also much faster than the algorithms that existed up to that time: It took 20 hours to extract around 890,000 sentence pairs, around 44,500 sentence pairs per hour, which is about 3.5 times faster than previous algorithms (Gale and Church 1991).

4.2.2 Partial Similarity-Based

Another method is similarity-based such as the one presented in Simard and Plamondon 1996. Here, alignment follows two steps (or passes). In the first step, *isolated cognates*

[2]With the algorithms that existed up to that time, it took 10 days to extract 3 million sentence pairs, 12,500 sentences per hour.

are used to mark sort of *anchors* in the texts. The term “cognate” refers here to two word-forms in different languages, whose first four characters are identical. Isolated cognates are cognates with no resembling word forms within a context window. It follows the assumption that two isolated cognates of different languages are parts of segments that are mutual translations and should be aligned with each other. These cognates are used as anchors, and the process is repeated recursively between the anchors, in order to find further isolated cognates within these boundaries, until no more anchor points can be found.

In an intermediate step, segmentation into sentence boundaries takes place and the search space is determined. In other words, based on the anchors found in the first step, it is determined which sentences could be aligned with each other. Only sentence-pairs that are within the same search space boundaries are alignment candidates.

In the second step, the final alignment takes place. Theoretically, any sentence alignment program that can operate within the restricted search space defined in the previous steps can take over the job. In Simard and Plamondon 1996, the authors use a statistical lexical translation model (commonly known as IBM Model 1, see Section 5.2.1), to measure how probable it is to observe one sentence given another sentence, and so find the sentences that are most likely mutual translations.

4.2.3 Translation-Based

Another possibility for aligning sentences is translation-based. Here, the alignment algorithm constructs a statistical word-to-word translation model of the corpus. It then finds the sentence alignment that maximizes the probability of generating the corpus with this translation model. In other words, it aligns sentences that are most likely translations of each other, given the translation model (Chen 1993).

4.2.4 Hybrid models

There are also hybrid sentence-alignment methods, combining several methods.

Moore 2002 presents a method in which sentence lengths are combined with word correspondences to find the best alignments. It works in three steps: First, sentences are aligned using a sentence-length-based model. Then, the sentence pairs with the highest probability, i.e., those that are most likely real correspondences of each other, are used to train a translation model. The translation model is then used to augment the initial alignment, so that the result is length- and translation-based (Moore 2002).

Another hybrid method was presented by Varga et al. 2005. It combines a dictionary- and a length-based method. Here, a sort of a dummy translation of the source text is produced using a translation dictionary which is supplied to the program[3]. The program

[3]Note that this is not a real restriction. See Section 4.4.4

then simply converts each token into its corresponding dictionary translation. After the dummy translation has been created, a similarity score is computed for each sentence pair. The similarity score consists of two components: a score based the number of shared words in the sentence pair (token-based) and a score based on the ratio of character counts between sentences (length-based). The program treats paragraph boundaries (special <p> tokens) as sentences with special scoring. This similarity score of a paragraph-boundary and a real sentence is always minus infinity, which makes sure they never align. This way, paragraph boundaries always align with themselves and can be used as anchors to keep paragraphs mutually aligned (Varga et al. 2005).

4.2.5 Summary

All the methods presented here perform very well on clean, well-structured data in similar languages. Already the Gale & Church algorithm from 1993 achieved a precision of 98% on the Canadian Hansards[4], which Gale and Church acknowledge are easy to align. What seems to have led researchers to develop better sentence alignment algorithms are speed (Chen 1993; Varga et al. 2005) and better performance on noisy data (such as 1-to-many alignments and misrecognized paragraph boundaries (Sennrich and Volk 2010)).

While speed might be considered a mundane issue, when working with noisy data or with a large amount of data, several alignment runs might be required until misalignments can be detected. When the alignment process takes less time, texts that are less suitable for alignment (mixed order of chapters, different prefaces, etc.) can be filtered out earlier, and pre-processing steps such as tokenization and sentence segmentation, which may also influence the alignment quality, can be tested. Tweaking and fine-tuning the model parameters may also require several runs (Varga et al. 2005).

In other words, it may take several attempts until unsuitable texts can be filtered out, the best pre-processing steps are identified, and the best model parameters are found. An algorithm which performs faster has a clear advantage in such cases.

4.3 More Recent methods

While the statistics- and length-based methods described in section 4.2 date back to the 1990's, more recently other methods were suggested.

4.3.1 Bleualign

One of these methods was presented in Sennrich and Volk 2010 and has been dubbed since as Bleualign. It arose as a method addressing the problem of aligning less "easily"

[4] Transcriptions of parliamentary debates which exist in English and in French

alignable corpora. Sentence alignment methods up to that time perform excellent on well-structured corpora with a high language similarity such as the Canadian Hansards or the Europarl[5] which are considered easy to align because they are well-structured—they provide markup information to identify speakers which is useful for creating anchor points and the subsequent alignments (Simard and Plamondon 1996; Sennrich and Volk 2011). However, when aligning pairs of languages which are fundamentally different and/or of less structured texts, the alignment task becomes more difficult (Sennrich and Volk 2010).

Bleualign uses BLEU as a similarity score to find sentence alignments. BLEU, which stands for Bilingual Evaluation Understudy, is a popular automatic metric for evaluating machine translation models. It measures the similarity between two sentences by considering matches of several n-grams[6] [7]. The higher the BLEU score, the higher the similarity between two sentences (Koehn 2009, p. 226).

Although BLEU has been criticized as a measure of translation quality, BLEU scores can be used for deciding whether two sentences are mutual translations: The higher the BLEU score, the more likely it is that two sentences are mutual translations. BLEU scores for two unrelated sentences is usually 0. Instead of aligning sentences of the source and the target language with each other, Bleualign aligns a machine translated version of the target side of the corpus with the source side in order to find the most reliable alignments (Sennrich and Volk 2010).

However, this approach requires an already existing machine translation system with reasonable performance. This problem was addressed in Sennrich and Volk 2011 by suggesting an iterative method for alignment, combining length-based and BLEU score-based methods, which doesn't require an already existing machine translation system. In the first iteration, sentences are aligned using an implementation of the Gale & Church algorithm, then a statistical machine translation (SMT) system is trained on the sentence-aligned corpus. In the following iterations, the corpus (target side) is machine-translated using the SMT system trained in the last iteration and is then aligned to the source side using Bleualign. Then, a new SMT system is trained using the current alignments.

Sennrich and Volk 2011 do not recommend this iterative sentence alignment procedure for all purposes. It should be used mainly where conventional sentence alignment algorithms such as Gale & Church have lower accuracy or where language-specific resources such as dictionaries (needed for hunalign (Varga et al. 2005)) or machine translation systems are unavailable or lacking in quality.

[5]Parliamentary proceedings of the EU Parliament

[6]Sequences of tokens of length n

[7]Usually scores are combined for n-grams of order 1 to 4.

4.3.2 Vecalign

The desire for sentence alignment of even higher quality rose with the insight that, while misaligned sentences have small effect on SMT performance, they do have a crucial effect on neural machine translation (NMT) systems. This is especially true in scenarios with less data for low-resource NMT (Thompson and Koehn 2019).

Vecalign uses a novel method which is based on the similarity of bilingual sentence embeddings. Sentence embeddings are, in a manner similar to word embeddings (see Section 5.3), vector representation of sentences that are learned by and can be extracted from a neural language model. This vector representation is said to represent the meaning of a sentence. The sentence embeddings are obtained from a language model that was trained on multiple languages, thus, the embeddings for all languages share the same vector space. This means that the embeddings are indifferent to the specific input language: They are language agnostic. If two sentences, regardless of their language, are similar, their vector representations will lie close to each other in the vector space. A function that is most often used for measuring vector similarity is the cosine similarity (see Section 5.3.3. In this manner, similar sentences in different languages can be identified and aligned (Artetxe and Schwenk 2019).

4.4 Sentence Alignment Pipeline

I shall now describe the steps I took for extracting sentence pairs out of the corpus I compiled in section 3.

4.4.1 Tool of choice

My tool of choice was hunalign (Varga et al. 2005). It is presented as a software package on GitHub, it is free to use and contrary to the Microsoft program presented by Moore 2002, its license allows corpora produced by it to be freely distributed. It is also well documented, was easy to compile on my system[8] and runs fast (aligning around 100,000 sentences takes about three minutes).

I tried, just for the sake of interest, to use Vecalign on a small portion of my corpus (300 sentences). Veclaign requires that all adjacent sentences be concatenated first (to allow for 1-to-many alignments). Then for each sentence-concatenation, the sentence embeddings have to be obtained from the LASER language model. Only then, can sentence alignment be calculated (Thompson and Koehn 2019).

The process of obtaining the sentence alignment took quite some time—around 10 minutes for 300 sentences—and by quick inspection with the bare eye, the result wasn't

[8] MacBook Air, M1 2020, 8GB RAM, running MacOS Monterey 12.3.1

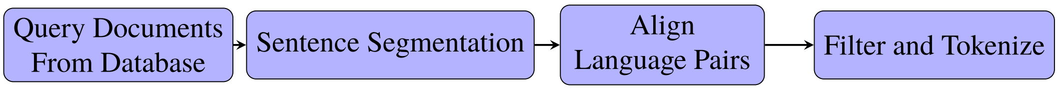

Figure 4.1: Sentence alignment pipeline

better than the one achieved with hunalign, but rather worse. Obviously, this may be due to the fact that Romansh is not one of the languages LASER was trained on. That being said, LASER *has* been said to generalize to unseen languages that are similar to the ones the model was trained on, e.g., Swiss German or West Frisian, which are similar to German and Dutch, respectively (Artetxe and Schwenk 2019) [9].

Since the corpus at hand is well-structured—the documents are pre-aligned, the translations are close translations, paragraphs in the source language correspond to paragraphs in the target language and the press releases are usually not longer than a few sentences—hunalign performed excellently. I didn't create a gold standard for sentence alignment, so automatic evaluation was not possible, but during the task of annotating word alignments for the gold standard of German-Romansh (see Chapter 6), I only had to discard 11 out of 611 sentences due to misalignment. This corresponds to a precision of 98.2% or an error rate of 1.8%.

4.4.2 Pipeline

The scripts responsible for compiling the sentence pairs are under the folder `align_sentences` on my repository on GitHub. The bash script `make_bicorpus.py` is responsible for executing the pipeline.

Figure 4.1 visualizes the steps taken for sentence alignment.

4.4.3 Database Query and Sentence Segmentation

In the first step, all aligned documents are extracted from the corpus and are written to monolingual files, one sentence per line, and one file per year. This is done by querying the SQLite database for all the aligned documents for each year, a task for which the script `exctract_multicorpus.py` is responsible.

Sentence segmentation (also called sentence tokenization) was done using NLTK's Punkt tokenizers (Bird, Loper, and Klein 2009). Since I wasn't able to integrate a sentence tokenizer for Romansh into the pipeline, I used the an NLTK Punkt tokenizer model which was trained on Italian. After instantiating both the German and the Italian models, I extended the list of abbreviations[10] to enhance the performance of the tokenizer and avoid wrong segmentation.

[9] See also https://github.com/facebookresearch/LASER

[10] The abbreviations for Romansh were kindly taken from Lisa Gasner's/Samuel Läubli's GitHub reposi-

In the course of sentence segmentation, paragraphs are retained by converting line breaks into special <p> tokens. These tokens will serve hunalign as anchor points for sentence alignment, cf., Section 4.2.4.

The result is three files for each year, one for each language, containing one sentence per line and <p> tokens marking paragraph borders. Further, to keep the corpus well-structured, the file ID (cf., section 3.3) is included at the beginning of each document. In case there is no mutual file ID, the date is included. The file IDs/dates will be used by `hunalign` as anchor points for keeping the documents aligned, see Listing 4.1 for an example.

Listing 4.1: Excerpt from a file containing sentences for alignment. In order to keep the file structured and increase alignment performance, each document starts with a date and paragraph are boundaries are marked with a special <p> token.

```
2004-01-27
www.gr.ch neu mit Online-Schalter und mit Interessenbindungen des
   Grossen Rats
Ein neues, zentrales Element von www.gr.ch ist der integrierte Behörden
   -Online-Schalter www.ch.ch.
...
Der Online-Schalter wird laufend in Zusammenarbeit zwischen Bund,
   Kantonen und Gemeinden weiterentwickelt und inhaltlich erweitert.
<p>
Parlament: Interessenbindungen öffentlich einsehbar
...
Weiter wurden die Funktionalitäten der Stichwortsuche verbessert, der
   Informationsgehalt im Bereich "Unser Kanton" erweitert ("Produkte
   aus Graubünden", Suchmaschine für Graubünden) sowie der
   Sprachenwechsel zwischen den Inhalten in deutsch, romanisch und
   italienisch vereinfacht.
<p>
Standeskanzlei: Leitbild neu im Internet
...
Zudem verrät www.staka.gr.ch auch, warum ein Picasso und der Begriff "
   Light" ohne weiteres mit der Standeskanzlei Graubünden in
   Zusammenhang gebracht werden können.
<p>
 Die neuen Web-Inhalte finden Sie hier:
 - Online- Schalter
 - Mitglieder
 - Stellvertreter
 - www.staka.gr.ch
<p>
 Gremium: Standeskanzlei Graubünden
 Quelle: dt Standeskanzlei Graubünden
```

tory.

4.4.4 Aligning Language Pairs

As described in Section 4.4.1, my tool of choice for aligning the sentence is hunalign. hunalign can use a bilingual dictionary for alignment, but the existence of such a dictionary is not a real restriction. In the absence of such a dictionary, the program will first fall back to sentence-length information, then automatically build a dictionary based on this alignment, and finally use this automatically-built dictionary for alignment in a second pass[11].

Although inspection with the bare eye revealed excellent precision (from the 611 sentences extracted for annotation of word alignment for the gold standard, only 11 were misalignments) which means the absence of a pre-made dictionary is not obstacle, when aligning the entire corpus, I used the German–Rumantsch Grischun dictionary downloaded from the online dictionary *Pledari Grond*[12] to support hunalign even further.

Files for three language pairs are then created: German–Romansh, German–Italian and Romansh–Italian, one file for each year. The files for each language combination are then concatenated. The result is three files containing all the sentence pairs for each language combination, from 1997 until today.

4.4.5 Filtering and Tokenizing

The press releases often contain sentences that are repeated throughout many of them, such as noting the source of the information at the end of the press release. A very common sentence ending a press release in German is *Quelle: dt Standeskanzlei Graubünden* "Source: German State Chancellory Grisons". Such duplicate sentences are not simply redundant in the corpus, but might also be considered noise in the data. Misaligned sentences and untranslated sentences are also considered noise that can have a negative influence on NMT models (Khayrallah and Koehn 2018). Therefore, duplicates and untranslated sentences should be filtered out, in order to make sure the remaining pairs are of high quality.

The script `filter_bicorpus.py` takes a file generated by hunalign (containing three tab-seperated columns: source–target–score) and produces a tab-separated file containing two columns (source and target) with the filtered sentences, one sentence per line and word-tokenized. The script removes sentences containing E-Mails, URLs or phone numbers, as well as sentences where source and target languages are identical, i.e., untranslated sentences. Sentences in which the difference in character length between source and target is too large (more than three times), for which I then assume misalignment, are also removed.

Word tokenization is important for the next step—word alignment. For the task of

[11] https://github.com/danielvarga/hunalign

[12] https://www.pledarigrond.ch/rumantschgrischun

tokenization, I used NLTK's (Bird, Loper, and Klein 2009) word tokeniziation functions, while applying the German model for German text and the Italian model for Romansh and Italian text. The justification for the latter is that Romansh, in a manner very similar to Italian, uses apostrophes to attach enclitics (articles and pronouns) to neighboring words, which should be separated for word tokenization. An inspection with the bare eye looked precise enough. In the course of annotating the word alignment for the gold standard, I had to correct the tokenization less than 10 times for 600 sentences.

4.5 Results

The resulting final parallel corpus consists of three files containing around 80,000 unique sentence pairs for each of the three language combinations: German–Romansh, German–Italian and Romansh–Italian. Each line in the file contains a sentence pair, separated by a tab character (see Listing 4.2).

Table 4.1 elaborates on the number of sentences, tokens and type for each combination.

Combination	Sentence pairs	Tokens Source	Types Source	Tokens Target	Types Target
German–Romansh	79,613	1,400,313	80,239	1,792,851	42,656
German–Italian	78,186	1,396,933	80,149	1,685,792	48,854
Romansh–Italian	78,101	1,760,424	42,295	1,655,822	48,753

Table 4.1: Parallel corpus in numbers, as of July 20, 2022. "Sentences" are sentence pairs. "Source" refers to the language on the left and "Target" to the language on the right, not necessarily to the actual source language of the translation.

```
Das kantonale Personal und die Volksschullehrerinnen und -lehrer müssen auf einen Teuerungsausgleich verzichten .→Il persunal chantunal e las scolastas ed ils scolasts da las scolas popularas ston desister d' ina gulivaziun da la chareschia .
Mit diesem Lohnopfer leisten sie in Würdigung der angespannten Finanzlage des Kantons und der schwachen Wirtschaftslage einen Beitrag dazu , die Kosten einzudämmen .→Cun quest sacrifizi da salari prestan els , a vista da la situaziun precara da las finanzas chantunalas e da la flaivla economia , ina contribuziun per franar ils custs .
Die Teilrevision des Behindertengesetzes wird auf Anfang 1998 in Kraft gesetzt .→La revisiun parziala da la lescha dals impedids vegn messa en vigur cun l' entschatta da 1998
```

Listing 4.2: Excerpt from the file containing sentence pairs in German–Romansh

Chapter 5

Word Alignment

Before We now reach the core of my thesis, computing word alignments using the novel method "SimAlign" (Jalili Sabet et al. 2020) and evaluating it against two baseline methods, I shall give a short introduction to the topic of word alignment and explain the mechanisms behind statistical word alignment and similarity/word embedding-based word alignment.

5.1 Introduction

Following the success statistical models had in the task of sentence alignment, word alignment was seen as a natural extension of that work. This work had two main goals: offer a valuable resource in bilingual lexicography and develop a system for automatic translation (Brown et al. 1993).

Word alignments are objects indicating for each word in a string in the target language f which word in the source language e it arose from (Brown et al. 1993). In other words, it is a mapping of words in a string of the source language e to the words in a string of the target language f (Koehn 2009, p. 84).

A simple example for an alignment for a pair of sentences from the corpus I compiled are the German sentence *Die Beratungen sind kostenlos* "The consultations are gratuitous" and its Romansh counterpart *Las cussegliaziuns èn gratuitas*.

1	2	3	4
Die	Beratungen	sind	kostenlos
Las	cussegliaziuns	èn	gratuitas
1	2	3	4

Figure 5.1: Example of a word alignment between two sentences in German and Romansh

In this example, each word in German is aligned to exactly one word in Romansh and the words follow exactly the same order, such that the resulting alignment is the set of mappings $\{1 \rightarrow 1, 2 \rightarrow 2, 3 \rightarrow 3, 4 \rightarrow 4\}$. Such alignments, in which each word in the source sentence is aligned to exactly one word in the target sentence, and in which the words follow the same order, are considered simple (Koehn 2009, p. 85).

Things become more complicated when word order differs between languages or when several words in one sentence are mapped to one or several words in the other sentence. The latter gives rise to a variety of alignment types. A word in the target language may be aligned to several words in the source language (1-to-many alignment), or several words in the target language may be aligned to one word in the source language (many-to-1 alignment). Sometimes words in the target have no relation to the source (for instance in case of untranslatable words, or words that were omitted in the translation). In that case, they will be aligned to a special `NULL` token (Koehn 2009, p. 85).

In order to deal with these challenges of different word order and alignments that are not 1-to-1 alignments, Brown et al. 1993 developed their pipeline of translation models, the IBM Models 1-5.

5.2 Overview of Methods

I shall now give a quick explanation of word alignment methods, namely of the IBM Models, and of SimAlign, a similarity-based alignment model that uses word embeddings. Since I am not a mathematician, I will not go into the mathematics of these models. I will rather attempt to explain their *modus operandi* in a more intuitive way, so as to to allow the reader some basic understanding of the mechanics behind the scenes.

5.2.1 IBM Model 1

The IBM models are translation models. They were developed in order to compute the conditional probability of a sentence in the target language f given a sentence in the source langauge e: $P(f|e)$ (Brown et al. 1993). In layman's terms, they compute how likely a given sentence in the target language is a translation of a sentence in the source language. By modeling these probabilities, the models can generate a number of different translations for a sentence. However, there are infinitely many sentences in a language and most sentences occur, even in large corpora, only once. This makes the task of modeling the probability distribution for full sentences hard and not promising. Instead, the problem is broken up into smaller steps: the model models the probability distributions for individual words—it computes how likely a word in one sentence is a translation of a word in that sentence's translation. The IBM Model 1 is therefore based solely on modeling the probability distributions of lexical translations, i.e., of individual words (Koehn 2009, p. 88).

Incomplete Data

There is, however, a problem. We can compute the probability distributions of lexical translations given their counts. That is, by counting how often a word s_i^e in the sentence s_e in language e was translated as a word s_j^f in a sentence s_f in language f, we can compute the desired probability distributions. Take for example a set of German-English sentence pairs. By counting how many times the German word *das* was translated as *the*, how many times it was translated as *that*, etc., we can compute each word's translation probability distribution. With these individual probability distributions we can compute the likelihood of a sentence in language f being a translation of a sentence in language e (Koehn 2009, p. 88). Unfortunately, while sentence alignment is a relatively easy task (at least for well-structured texts), and while sentence aligned parallel corpora are not hard to compile or come by, we do not know which words correspond to which words in the sentence pairs. In other words, we do not know *a priori* how each word in the source sentence was translated, which means we cannot compute the counts for the probability distributions.

This problem, dubbed as a *chicken and egg problem*, is basically the following: If we had word alignments, it wouldn't be a problem to estimate the lexical translation model and compute the probability distributions for words and sentences; And if we had a model, we could easily estimate the most likely correspondences between words in the source and the target sentences. Unfortunately, we have none of the above (Koehn 2009, p. 88).

EM Algorithm

In order to solve the problem of incomplete data, an iterative learning algorithm, the expectation-maximization (EM) algorithm comes into play. The EM algorithm is mathematically intricate. I shall try to explain in simple words the idea behind it.

In the very first iteration, the values of the model parameters are unknown and are initialized with a uniform distribution. This means all words are equally likely translations of each other. Then, in the estimation step, the model is applied to the data to compute the most likely alignments. In the maximization step, the model is learned from the data based on counts collected from it. The algorithm counts co-occurrences of words in the source and the target languages, which are then weighted with the probabilities that were computed in the estimation step. These weighted counts are used to compute again the probabilities in the next estimation step. These two steps, estimation and maximization, are then repeated until convergence—until a global minimum has been reached (Koehn 2009, pp. 88–92; Brown et al. 1993).

In simple words, the model does not know in the beginning which words in the source language correspond to which words in the target language. In the very first iteration, all alignments are equally likely—any word in a sentence in the target language is equally likely a translation of any word in the source language. In order to find the most probable correspondences (or alignments), the model counts how often words are aligned with each other, that is, how often they co-occur in parallel sentences (maximization step). These counts are weighted with the probabilities computed in the previous estimation step to refine the values in the next estimation step. Likely links between words are strengthened, while less likely links are weakened. This goes on until the model converges and the most likely word alignments have been learned by the model.

5.2.2 Higher IBM Models

Without going too much into detail, I will shortly mention the other IBM models, Models 2-5.

Model 1 makes the unrealistic assumption that all connections for each position are equally likely. This means that word order is not modeled by Model 1. Simply put, the word order does not influence the likelihood of word alignments. Therefore, Model 2 *does* depend on word order. It adds an explicit model for alignment based on the absolute positions of the source and the target words (Brown et al. 1993; Koehn 2009, p. 99).

Model 3 adds a probability distribution of the number of words a source word is usually translated to (dubbed *fertility*). It is able to model alignments of types other than 1-to-1 (Koehn 2009, p. 100).

Models 4 and 5 add more complexity and take into account for instance the positions of any other target words that are connected with the same source word (Brown et al. 1993), since words that are next to each other in the source sentence tend to be next to each other in the target sentence (large phrases tend to move together as units) (Koehn 2009, p. 107).

Models 1-4 serve as stepping stones towards the training of Model 5. Model 1 has a simple mathematical form and a one unique local minimum, which means the parameters learned by it do not depend on the starting point[1]. The estimates learned by Model 1 are used to initialize the training of Model 2, those of Model 2 are used to initialize Model 3, and so on, and so forth—each model is initialized from the parameters of the model before it. This way, the estimates arrived at by the end of training of Model 5 do not depend on the initial estimates of the parameters for Model 1 (Brown et al. 1993).

These models have been playing a key role in word alignment tasks and in statistical- and phrase-based machine translation. Put together in a pipeline of models, they serve as the groundwork for Giza++, a toolkit for training word-based translation models. Using these alignments, phrase alignments can be learned in order to train a statistical phrase-based machine translation (Och and Ney 2000; Koehn, Och, and Marcu 2003)

[1]The other models have several minima; this means according to the starting parameters, different minima can be arrived at.

5.3 Word Embeddings

A different approach to word alignment is based on similarity between words, which is in turn computed using word embeddings. But what are word embeddings?

5.3.1 Excursion: Words

Before we discuss word embeddings, I would like to write a few words about words and their meanings.

Words are actually an arbitrary way to split linguistic material into units. What we refer to as words are usually units separated by a whitespace in writing, but the use of whitespaces is arbitrary and inconsistent. There is no real phonetic motivation for splitting units into words. Some single words sound exactly like two other words (*a maze* sounds like *amaze* and *in sight* like *incite*). The words *someone* and *anyone* are written as one word, while *no one* is written as two words, although there is obviously no difference in character between them (Jespersen 1924, pp. 92–95).

For the sake of simplicity, I will stick to the term *word*, referring to any linguistic unit, made up of one or several morphemes (or words), divided in written form by whitespaces from its neighboring units.

Meaning of Words

The question of describing the meanings of words is an entire field: semantics. But already in his posthumously published work *Cours de linguistique générale* ("Course in General Linguistics") from 1916, the Swiss linguist and semiotician, Fredinand de Sassure, came to an important conclusion. Linguistic elements receive their value only by being arranged in a sequence, which de Saussure calls *syntagm*: "A term in the syntagm acquires its value only because it stands in opposition to everything that precedes or follows it, or to both." (Saussure 1959, p. 123)

Additionally, each term in the syntagm, in the sequence of terms, has associative (or *paradigmatic*) relations. These relations reside in the memory of the speakers. For instance the German word *zudrehen* "close something by turning" unconciously calls to mind related words, such as other words beginning with *zu-*: *zumachen* "close", *zumauern* "wall something up", *zuklappen* "close something shut". But also words with the verb *drehen*: *aufdrehen* "turn open", *verdrehen* "twist, contort", etc. etc. (Saussure 1959, pp. 122–127)[2].

Each term in the syntagm stands in opposition not only to the preceding and following parts in the syntagm, but also to terms in the paradigm, which are called to mind by the

[2]Examples are my own.

associative series. The meaning, or rather value of words, is a result of an intersection of two axes—the syntagmatic, the horizontal axis, and the paradigmatic axis, the vertical axis.

Take, for instance, the sentence *I am drinking coffee*. The word *coffee* gets its **syntagmatic** value from the perceding word *drinking*, which stands in **paradigmatic** opposition to other words (*plant*, *grow*) which would give *coffee* a different meaning. We know that by *coffee* a hot-drink is meant, because it follows the verb *drink*. In the sentence *I grow coffee* it would mean a plant or a tree, in *I bought one pound of coffee* it would mean beans, and in *coffee ice-cream* it would describe a flavor.

The Austrian-British philosopher, Ludwig Wittgenstein, summed up the meaning of the word *meaning* (German *Bedeutung*) in two sentences in his Philosophical Investigations, no. 43:

> *Man kann für eine große Klasse von Fällen der Benützung des Wortes »Bedeutung« – wenn auch nicht für alle Fälle seiner Benützung – dieses Wort so erklären: Die Bedeutung eines Wortes ist sein Gebrauch in der Sprache.*[3] [4]

5.3.2 Word Embeddings

These ideas, which were further developed by linguists in the 1950's, namely that a word can be defined by its environment or distribution, i.e., by its set of contexts in which it occurs and its grammatical environments, is the inspiration for what is called vector semantics. The idea of vector semantics is to represent a word as a point in some n-dimensional vector space. These vectors are called *embeddings*. There are different ways and versions of word embeddings, but in each case the values of the vectors are based in some way on counts of neighboring words (Jurafsky and Martin 2019, pp. 98–99).

Neural Language Models

One version of word embeddings comes from neural language models. Language modeling is the task of assigning probabilities to a sequence of words, that is, modeling how likely it is that a sequence of words in a language would be uttered/written by a speaker of that language (Koehn 2009, p. 181). In practice, the task of a language model is predicting upcoming words from prior word context (Jurafsky and Martin 2019, p. 137).

In a neural language model, the modeling is done using a neural network. Without going too much into detail, a neural network is a complex non-linear function. It is made

[3]For a large class of cases of the use of the word *meaning*—and maybe for all of its use cases—one could explain the word as follows: The meaning of a word is its use in the language.

[4]https://www.wittgensteinproject.org/w/index.php?title=Philosophische_Untersuchungen#43

up of layers, which are vectors, and weights, which are matrices. The numbers (a vector) from each layer are passed on to the next layer by multiplying it with the weights (a matrix) between the layers using matrix multiplication. The vector resulting from this matrix multiplication (usually passed through some non-linear activation function), is the next layer in the neural network. The output of a neural network can be a single value, as in the cases of a binary classification task, in which the output is either 0 or 1, but it can also be a vector representing some probability distribution.

In the course of the training of a neural language model, i.e., while the neural network learns the probability distributions for words given its neighboring words, the parameters for the weights are learned. The weights connecting the input layer with the first hidden layer are our said word embeddings. When inputting a word into the network (in form of a one-hot vector), we can get its vector representation, i.e., its embedding, from the so-called embedding layer. Since this representation is conditioned on context, similar words should have similar embeddings (Koehn 2020, pp. 104–105).

Neural Embeddings

There are different ways for learning word embeddings. One of the most popular methods are *word2vec* (actually made up of two different methods) and *GloVE*. These methods are simpler than neural language models (Jurafsky and Martin 2019, p. 111); their main goal is to learn high quality word vector representations, not to generate language.

Sub-words

Due to computational limitations, neural language models usually have a fixed vocabulary size. This means that even if we had some hypothetical corpus which contains all the words in a language, the model will still not be able to "learn" all these words. Some words will remain out-of-vocabulary. There are different ways for dealing with this limitation in vocabulary size, i.e., with rare words. One way is to split words into small sub-word units. There are different algorithms for splitting words. mBERT uses an algorithm called WordPiece (Y. Wu et al. 2016; Devlin et al. 2018) and XLM-R uses BPE (Conneau et al. 2020; Sennrich, Haddow, and Birch 2016b).

5.3.3 Word Similarity

If words are represented by vectors, we need a measure for taking two such vectors and determining how similar they are. The most common similarity metric is the **cosine similarity**—measuring the angle between the vectors.

Again, without going into too much mathematical details, using the dot product for measuring similarity, i.e., multiplying the vectors with each other, favors long vectors.

Long vectors are vectors with high values in each dimension, which represents the frequency of words. This means more frequent words would have higher values, but we are interested in measuring the similarity between words regardless of their frequency. To solve this problem, we need to **normalize the dot product** by dividing it by the lengths of the vectors. Thus, the cosine similarity metric between two vectors $\mathbf{v}$ and $\mathbf{w}$ can be computed as:

$$\text{cosine}(\mathbf{v}, \mathbf{w}) = \frac{\mathbf{v} \cdot \mathbf{w}}{|\mathbf{v}||\mathbf{w}|} = \frac{\sum_{i=1}^{N} v_i w_i}{\sqrt{\sum_{i=1}^{N} v_i^2}\sqrt{\sum_{i=1}^{N} w_i^2}} \tag{5.1}$$

With $\sum_{i=1}^{N} v_i w_i$ being the dot product of the vectors $\mathbf{v}$ and $\mathbf{w}$, and $\sqrt{\sum_{i=1}^{N} v_i^2}$ and $\sqrt{\sum_{i=1}^{N} w_i^2}$ being the lengths of the vectors $\mathbf{v}$ and $\mathbf{w}$, respectively (Jurafsky and Martin 2019, pp. 103–104).

The cosine similarity returns a value between -1 and 1. The highest similarity is 1: the vectors are parallel and pointing in the same direction. If it is 0, the angle between the vectors is a 90° angle. The lowest similarity is -1: the vectors point in opposite directions.

5.3.4 Multilingual Word Embeddings

There are also methods for computing multilingual word embeddings. Multilingual word embeddings are word embeddings for words in different languages that share the same vector space. This can be achieved by learning word embeddings for each language separately on monolingual data, and then map these embeddings to a shared vector space (Artetxe, Labaka, and Agirre 2018). Multilingual word embeddings can also be extracted from a multilingual language model (Jalili Sabet et al. 2020).

The idea behind multilingual word embeddings is that two equivalent words in different languages should have a similar distribution, thus their vector representations should also be similar (Artetxe, Labaka, and Agirre 2018).

5.3.5 Summary

Word embeddings are vector representations of words learned by a neural language model or by a more simple embeddings model. These vectors' dimensions usually range between 100 and 1000 dimensions. Similar words (words that appear in the same context) have similar word embeddings. To measure word similarity, we measure the similarity between their embeddings using the cosine similarity. Multilingual word embeddings are word embeddings for words in different languages sharing the same vector space. Similar words in different languages should have similar embeddings.

		1 Ich	2 liebe	3 ja	4 Äpfel
1	I	0.9	0.2	0	0.2
2	love	0.1	0.9	0	0.1
3	apples	0.1	0.1	0	0.9

Figure 5.2: Similarity matrix $S \in [0,1]^{l_e \times l_x}$, filled with values between 0 and 1 corresponding to the similarity measure between the embeddings of the words. The values are fictive.

		1 Ich	2 liebe	3 ja	4 Äpfel
1	I	1	0	0	0
2	love	0	1	0	0
3	apples	0	0	0	1

Figure 5.3: Alignment matrix $A \in \{0,1\}^{l_e \times l_f}$ extracted from the similarity matrix S. The two most similar words in row i and column j of S will receive a score of 1; the rest 0.

5.4 Similarity-Based Word Alignment

If similar words in different languages have similar embeddings, these embeddings can be leveraged in order to find word alignments using a similarity matrix, without the need for parallel data. This is the idea that forms the basis of SimAlign (Jalili Sabet et al. 2020).

5.4.1 Method

SimAlign takes two parallel sentences s_e and s_f of lengths l_e and l_f in languages e and f. For this sentence pair a *similarity matrix* is defined as $S \in [0,1]^{l_e \times l_f}$. It is a matrix the size of the lengths of the sentences. Each cell in the matrix will be filled with a value between 0 and 1, returned from a function measuring similarity between the embeddings of two words. This means that for each combination of two words from sentence s_e and sentence s_f, their similarity measure is filled into the corresponding cell in the matrix (Figure 5.2). From this similarity matrix S, a binary alignment matrix $A \in \{0,1\}^{l_e \times l_f}$ is extracted. The cell A_{ij} in the alignment matrix A will be filled with 1 (which means i and j will be aligned) if the word s_i^e in the sentence s_e is the most similar to the word s_j^f in the sentence s_f and vice versa (Figure 5.3).

That is, a cell A_{ij} in the matrix A is set to 1 if:

$$(i = \arg\max_l S_{l,j}) \land (j = \arg\max_l S_{i,l})$$

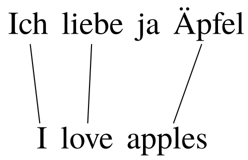

Figure 5.4: The resulting word alignment

If all entries in a row i or a column j of S are 0 (as is the case in column 3 of Figure 5.2), A_{ij} will be set to 0. The resulting alignment can be seen in Figure 5.4.

This basic method is referred to in Jalili Sabet et al. 2020 as **Argmax**. Mutual argmaxes can be rare, which is why for many sentences Argmax only identifies few alignments. To remedy this, Argmax is applied iteratively in a method called **Itermax**. In each iteration, the model focuses on still unaligned pairs and tries to align them. Further, if the similarity with an already aligned word is very high, the model can add another alignment edge. This allows for one word to be aligned to multiple other words, i.e., create 1-to-many alignments.

Argmax finds a local optimum and Itermax is a greedy algorithm. There is a third alignment method, called **Match**, which finds global optima. The alignments generated with the Match method are inherently bidirectional (the source is aligned to the target and the target is aligned to the source)[5].

For the task of word alignment, SimAlign can use multilingual embeddings which were learned in advance from monolingual data and then mapped to a shared vector space. SimAlign can also use, out-of-the-box, the embeddings from two multilingual language models: mBERT, which is a version of BERT (Devlin et al. 2018) trained on 104 languages[6], and XLM-RoBERTa base, trained on 100 languages (Conneau et al. 2020).

However, none of these models has "seen" Romansh, i.e., Romansh is not part of the training data for these models. But multilingual models were shown to generalize even for unseen languages. mBERT, for instance, achieves reasonable results out-of-the-box (without further training) on unseen languages in a variety of tasks such as named entity recognition (NER) and part of speech (POS) tagging (Pires, Schlinger, and Garrette 2019). There is therefore good reason to expect that SimAlign would also work for aligning words in sentence pairs with Romansh.

5.4.2 Summary

By measuring the similarity between multilingual word embeddings, word alignments for sentence pairs in the languages the models were pre-trained on can be computed. Multilingual embeddings can be learned from monolingual data, and thus word alignment

[5]Jalili Sabet et al. 2020 don't elaborate on the relevance of the notion of source and target sentences.

[6]`https://github.com/google-research/bert/blob/master/multilingual.md`

can be computed even in low-resource scenarios, i.e., in scenarios where parallel data is scarce, which makes similarity-based word alignment a competitive method against statistical methods.

Traditional statistical methods such as the IBM Models (Brown et al. 1993) and their implementations, such as GIZA++ (Koehn, Och, and Marcu 2003) or fast_align (Dyer, Chahuneau, and Smith 2013) require a large amount of parallel data to perform well. The quality of the alignments deteriorates quickly when the size of data diminishes[7].

In experiments done by Jalili Sabet et al. 2020, their similarity-based word alignment method, when using embeddings extracted from mBERT or XLM-R, outperforms any state-of-the-art statistical method for the languages Czech, German, French and Hindi, paired with English. However, all of these languages were included in mBERT's and XLM-R's training data. Jalili Sabet et al. 2020 emphasize the advantage of their method being high performance also in the case of little parallel data.

In the following two chapters I will describe the creation of a gold standard (Chapter 6) in order to answer my research question and test **whether SimAlign performs just as well on data unseen by said language models, specifically for the language pair German-Romansh** (Chapter 7).

[7]In Och and Ney 2000, the alignment error rate (AER) for aligning words in 1.5M sentence pairs is 9.4%. When aligning words in only 50,000 sentences, the AER goes up to 15.6% (see Table 4 in Och and Ney 2000)

Chapter 6

Gold Standard

6.1 Introduction

In the previous chapter, I discussed SimAlign, a method for computing word alignments based on measuring the similarity between multilingual word embeddings. The clear advantage of this method is that it does not rely on the existence of parallel data— The multilingual word embeddings can be learned from monolingual data. Jalili Sabet et al. 2020 evaluated their method on language pairs which were all part of the training data for the language models in use (mBERT and XLM-R). I shall now proceed to test how well SimAlign performs on the language pair German-Romansh, under the consideration that Romansh is not part of the training data for these language models, i.e., it is an unseen language.

In order to measure the quality of words alignments, a model's performance is measured on a test set, dubbed gold standard, which is created by human annotators. For the gold standard to be of good quality and consistent with itself, annotators have to follow strict guidelines. These guidelines address issues of ambiguity in word alignments. (Koehn 2009, p. 115).

Some problematic cases that might occur are function words[1] that have no clear equivalent in the other language. Koehn 2009 gives as an example the German-English sentence pair: *John wohnt hier nicht* and *John does not live here*. What German word should the English word *does* be aligned to? Three different choices can be made:

1. The word should remain unaligned since it has no clear equivalent in German.
2. The word *does* is connected with *live*; it holds information about number (singular) and tense (present tense), which, in German, is contained in one word: *wohnt*. Thus, it should be aligned to *wohnt*, together with *live*.

[1]Function words form a closed class of words (a fixed set of words with virtually no new additions), they occur frequently and often have structuring uses in grammar. Pronouns, prepositions and conjunctions like *of*, *it*, *and*, or *you* are function words (Jurafsky and Martin 2019, p. 144).

3. *does* is part of the negation; without it, the sentence would not contain this word. Therefore, *does* should be aligned with *nicht* (the German negation).

There are several possibilities, all of them arguable, none of them plain wrong. This illustrates the need for clear guidelines.

6.2 Sure and Possible Alignments

An approach for solving problematic cases is the distinction between "Sure" and "Possible" alignments (Och and Ney 2000), which are also sometimes referred as "fuzzy alignments" (Clematide et al. 2018). Generally, these labels allow to distinguish between ambiguous and unambiguous links. Ambiguous links are labeled Possible and unambiguous links are labeled Sure (Lambert et al. 2005). The Possible label was conceived to be used especially for aligning words within idiomatic expressions, free translations and missing function words (Och and Ney 2000). This distinction also has an impact on the way the evaluation metrics are computed (see Section 7.1).

There seems to be no clear global definition about which alignments should be considered umabiguous and thus marked as Sure, and which should be considered ambiguous and marked as Possible. For some created gold standards, no distinciton between Sure and Possible alignments was made at all (Clematide et al. 2018). In another case, annotators were asked to first label all alignments as Sure and then refine their alignments with confidence labels (Holmqvist and Ahrenberg 2011). And in yet anoter instance, two annotators used only Sure links. Their annotations were then combined; all 1-to-1 alignments both annotators agreed upon (i.e., the intersection of their annotations) were makred as Sure and all other alignments were marked as Possible (Steingrímsson, Loftsson, and Way 2021). Different annotation schemes use Sure and Possible alignment in different ways.

6.3 Gold standard for German-Romansh

As explained before, in order to measure the performance of the different models, the similarity-based model (SimAlign) and the stastitical models (fast_align and eflomal), on the language pair German-Romansh, a gold standard is needed.

Since no such gold standard exists, I took upon myself to create one. Although I am not a speaker of Romansh, my experience as a trained linguist, as well as my knowledge in related languages (Latin, Italian, French), allows me to confidently tackle this task. Additionally, whenever I was in doubt, I referred to the online dictionary Pledari Grond[2], which also offers a grammar overview.

[2]https://www.pledarigrond.ch/rumantschgrischun

6.3.1 Annotation tool

I used the tool AlignMan which was originally programmed for creating the gold standard for English-Icelandic (Steingrímsson, Loftsson, and Way 2021). It is quite easy to use and its code is readable. I also had to make some small changes to the code. For instance, the sentences to be aligned, while loaded into the database, were read in opposite order, such that the source language became the target language and vice versa. I fixed this issue, so that source (German) and target (Romansh) languages stay the same accross all applications.

As mentioned in Section 6.2, the annotation scheme used by Steingrímsson, Loftsson, and Way 2021 does not allow labeling of links with Sure and Possible. Instead, AlignMan treats the union of 1-to-1 alignments made by two annotators as Sure alignments and all other alignments as Possible. This means, each annotator is expected to only annotate Sure alignments. This also applied to myself while annotating the German-Romansh gold standard: I only annotated Sure alignments.

6.3.2 Guidelines

As mentioned before, clear guidelines need to be defined for creating the gold standard in order to ensure quality and consistency. I shall now proceed to describe the guidelines I used for my annotation of the word alignments for the gold standard.

A motto often cited for annotating word alignments is "Align as small segments as possible, and as long segments as necessary." (Vronis and Langlais 2000, cited in Ahrenberg 2007) A variation of this is found in Clematide et al. 2018: "As few words as possible and as many words as necessary that carry the same meaning should be aligned," referring to Lambert et al. 2005. This motto guided me throughout the annotation task and it especially comes to mind in Principle II below.

In the following sections I will list some general principles as well as more specific principles involving German and Romansh.

6.3.3 General priniciples

Principle I. Use only Sure alignments: Since the annotation tool I was using does not provide the use of confidence labels (cf. Section 6.3.1: Annotation tool), I only aligned words which would be considered Sure alignments, i.e., they are unambiguous (cf. Section 6.2).

Principle II. Prefer 1-to-1 alignments over 1-to-many alignments or n-to-many alignments: Since all alignments are seen as Sure alignments, 1-to-many alignments should be avoided, unless a single word in the source sentence lexically corresponds to several words in the target sentence. This means alignments of phrases should be avoided. This is also due to the fact that we are testing models for automatic word alignment, and not phrase alignment.

Words that are repeated in one language, but not in the other, should only be linked once, leaving the repetition unaligned.

Principle III. Lexical alignments should always be preferred over all other alignments (part of speech (POS) alignments or morphosyntactical alignments). This means alignments should describe first and foremost lexical correspondences, i.e., both words have the same lexical meaning (but not necessarily share the same grammatical function or the same POS). Only words that are translations of each other also outside of the specific context of the sentence pair at hand should be aligned. This is in line with Clematide et al. 2018. In cases of paraphrasing during translations, words should remain unaligned.

6.3.4 Examples

I will now give some examples to illustrate the above principles.

Compound words

Compounding is the formation of new lexemes by adjoining two or more lexemes (Bauer 1988). In German, compounds are productive and prominent means of word formation in German (Clematide et al. 2018). In a sample of 4,500 types examined by Clematide et al. 2018, 80% of German nouns were compounds. Romansh, in comparison, uses prepositions (usually *da*) for linking nouns, with one noun modifying the other (Tscharner and Denoth 2022). Some other prepositions used for linking words are *cunter* and *per*. [3] In other cases, German compounds might be translated to Romansh using an adjective + noun, e.g., German *Gastkanton* was translated to *chantun ospitant* "hosting canton". See Table 6.1 for more examples.

German compounds will be aligned to their equivalent lexical words, but not to function words, resulting in a 1-to-many alignment: *Webseite ~ pagina [d'] internet*, *Gebäudeversicherung ~ Assicuranza [d'] edifizis*. This is also inline with principles I, II and III in Clematide et al. 2018. See Figure 6.1 for alignment examples.

German preterite vs. Romansh perfect

In the corpus at hand, two tenses are used in German for referring to past events: the preterite and the perfect. The German preterite is a synthetic verb form, i.e., it is made up

[3] Typologically, this is inline with other Romance languages such as French, which uses prepositions (*de*, *en* and *à*) for linking two nouns, e.g., *une robe de soie* "a silk dress" (Price 2008, p. 510).

Figure 6.1: Aligning German compounds to a Romansh noun phrases

German	Romansh	
Beratungsstelle	*post* ***da*** *cussegliaziun*	"consultation point"
Gebäudeversicherung	*Assicuranza* ***d'****edifizis*	"building insurance"
Webseite	*pagina* ***d'****internet*	"web site"
Kindermasken	*mascrinas* ***per*** *uffants*	"children masks"
Brandversicherung	*assicuranza* ***cunter*** *fieu*	"fire insurance"
Gastkanton	*chantun ospitant*	"hosting canton"

Table 6.1: Translation examples of German compounds into Romansh

of a single conjugated form. Some examples are *nahm* (infinitive *nehmen* "take") or *wurde* (infinitive *werden* "become"). The German perfect is an analytic construction made up of an auxiliary verb (*haben* "have" or *sein* "be") and the past participle, e.g., *Die Präsidentenkonferenz* ***hat*** *nun* ***entschieden*** "the presidential conference has decided".

In contrast to German, Romansh only has one tense referring to past events: the perfect. It is an analytic construction made, in a similar fashion as in German, of an auxiliary *habere* "have" for transitive verbs or *esse* "be" for intransitive verbs and the past participle (Bossong 1998, p. 189). The German sentence given above (*Die Präsidentenkonferenz hat nun entschieden*) was translated as *La conferenza da las presidentas e dals presidents* ***ha*** *usse* ***decidi***. *ha* is the auxiliary and *decidi* is the past participle. This poses no real problem since we can link the German auxiliary to the Romansh auxiliary and the German participle to the Romansh participle.

However, a German preterite is always translated using the Romansh perfect. For example, in the sentence *Der Kanton Graubünden war letzsmals 2003 Gastkanton* "The last time the Canton of Grisons was a host canton was in 2003" the verb *war* "was" is translated as *è stà*. This theoretically results in a 1-to-2 link. However, since Romansh *è* only carries grammatical information of tense and number, but no real lexical information, it

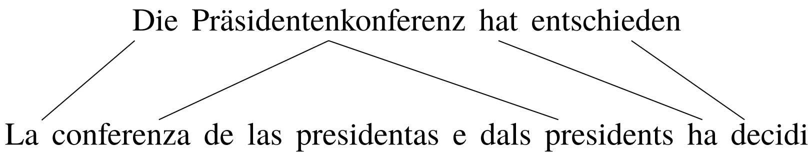

Figure 6.2: Aligning German perfect to Romansh perfect

Der Kanton Graubünden war letztmals 2003 Gastkanton .

Il chantun Grischun è stà l’ ultima giada l’ onn 2003 chantun ospitant .

Figure 6.3: Alignment of German preterite to Romansh perfect

should remain unaligned.

The German perfect should be aligned to the Romansh perfect using a 1-to-1 alignment; auxiliary to auxiliary and participle to participle. **The German preterite should also be aligned using a 1-to-1 alignment to the Romansh participle, leaving the Romansh auxiliary unaligned and avoiding a 1-to-2 alignment.** (cf. Principle II: prefer 1-to-1 alignments)

German present participle

German present participles (known in German as *Partizip I*) are translated to Romansh using relative clauses. Moreover, adjectives (and participles in the function of adjectives), can be nominalized, meaning they become the head of a noun phrase and there is no need for an actual noun. A good example for that in the corpus is the German noun phrase *nichtarbeitslose Stellensuchende* (cf. Ex. 1), which was translated as a noun phrase with a relative clause: *persunas che tschertgan ine plazza che n’èn betg dischoccupadas* “persons who look for a job (and) who are not unemployed”.

(1) nicht-arbeit-s-los-e Stellen-such-end-e
not-work-GEN-less-PL job-search-PRES.PART-PL
“People looking for jobs who are not unemployed”

nichtarbeitslose Stellensuchende

persunas che tschertgan ina plazza che n’ èn betg dischoccupadas .

Figure 6.4: Aligning a German present participle to a Romansh relative clause

In this case, these two phrases should not be aligned as phrases, but only the content words which lexically correspond to each other: *nichtarbeitslose* ~ *betg dischoccupadas*; *Stellensuchende* ~ *tschertgan [ina] plazza*. Figure 6.4 illustrates this.

Double negation

Negation in Romansh is constructed using two particles: *na* and *betg* to negate verbs or *nagin-* to negate nouns. Since we prefer 1-to-1 alignments (Principle II), the German negations *nicht* (for verbs) and *kein-* (for nouns) should be aligned only to the second

Romansh particle (*betg/nagin-*), leaving Romansh *na* unaligned. This is also linguistically motivated: in certain cases, *na* can be omitted (Caduff, Caprez, and Darms 2008, section 285).

Articles and Prepositions

German articles inflect in case, which expresses some syntactic relations involving nouns. Romansh often uses preopsitions for expressing the same relations. Take the German genitive case in *Zustimmung der Person* "the person's agreement" which is translated as *consentiment da la persuna*. I align the German article in genitive *der* with the Romansh preposition *da*, leaving *la* unaligned. Except for my preference for 1-to-1 alignments, the motivation for this is that it is the preposition *da* that expresses the genitival relations between the nouns.

Separable verbs

Separable verbs are verbs in front of which affixes (mostly prepositions) are placed. These affixes delimit and modify the verb's meaning (Dreyer and Schmitt 2009, p. 47). Since both the verb and the affix form together the meaning of the word and are conceptually inseparable, both of them should be aligned to the corresponding Romansh verb, resulting in a 2-to-1 alignment.

6.4 Flaws

I shall now discuss the quality of my gold standard and some of its flaws.

The most obvious flaw is the fact that I created the gold standard alone, without a second annotator. With more than one annotator, more elaborate annotating schemes can be used in order to ensure higher quality, consistency and harmony. For instance the annotators' agreement can be measured using the so-called inner-annotator agreement (Holmqvist and Ahrenberg 2011). Further, the intersection of the annotators' Sure alignment can be used to build the final Sure alignments set and the reunion of the annotators' Possible alignments can be used to create the final Possible alignments set (Mihalcea and Pedersen 2003). A third annotator can also revise and resolve conflicts between two annotators (Mihalcea and Pedersen 2003). When several annotators work on the same task, they can also discuss conflicts and resolve them using a majority vote (Melamed 1998). All of these possible schemes cannot be realized in the case of a single annotator, which was my case.

Another flaw are the missing confidence labels (Sure and Possible), which may influence the evaluation scores (see Section 7.4.1: General Problems with Evaluation). There are however precedents for gold standards without Possible links, using only Sure links (Clematide et al. 2018; Mihalcea and Pedersen 2003). It is therefore arguable.

6.5 Statistics

The 600 sentence pairs of the gold standard contain 6743 German tokens and 9158 Romansh tokens. The gold standard contains 6962 edges, 6275 of them are 1-to-1 alignments.

Unfortunately, I did not keep tabs on time during annotation, but I would estimate that at a rate of around 60 sentences per hour, annotation took around 10 hours (not including setting up the alignment program and defining the annotation guidelines).

Chapter 7

Results

After having created a gold standard (see Chapter 6) for evaluating the quality of the alignments, I compared the alignments computed by SimAlign with the alignments computed by two baseline systems. I shall now proceed to present the results of these experiments.

7.1 Evaluation Metrics

To evaluate the quality of word alignment, four measures are used. The first three—precision, recall and F-measure—are traditional measures in information retrieval (Mihalcea and Pedersen 2003).

Precision is the percentage of items that the system retrieved, which are indeed positive. It answers the question "how many of the items marked as positive by the system are in fact positive?" and is defined as Precision $= \frac{\text{TP}}{\text{TP+FP}}$, with TP being "true positives" and FP being "false positives" (Jurafsky and Martin 2019, p. 67).

Recall is the percentage of true positives retrieved by the system out of all positives. It answers the question "how many of all the true positives were actually found by the system?" and is defined as Recall $= \frac{\text{TP}}{\text{TP+FN}}$, with TP being "true positives" and FN being "false negatives" (Jurafsky and Martin 2019, p. 67).

F-measure is a score that incorporates precision and recall. The fourth measurement for evaluating word alignment, alignment error rate (AER), was introduced by Och and Ney 2000.

For computing the evaluation scores of the word alignments, I used a script made available on GitHub[1]by the creators of SimAlign (Jalili Sabet et al. 2020). The script uses a definition of precision, recall and AER which stems from Och and Ney 2000 and was later used by many others (Mihalcea and Pedersen 2003; Och and Ney 2003; Östling and Tiedemann 2016; Jalili Sabet et al. 2020). Precision, recall, F-measure and AER are defined as follows:

[1] https://github.com/cisnlp/simalign/blob/master/scripts/calc_align_score.py

$$\text{Recall} = \frac{|A \cap S|}{|S|}, \quad \text{Precision} = \frac{|A \cap P|}{|A|}, \quad F_1 = 2\frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

$$\text{AER} = 1 - \frac{|A \cap S| + |A \cap P|}{|A| + |S|}$$

With A being the set of alignments generated by the model, S being the set of Sure alignments and P the set of Possible alignments, and $S \subseteq P$, meaning the set of possible alignment P contains also all of the Sure alignments (Och and Ney 2000).

I will later discuss shortly some of the problems I see in these evaluation schemes (Section 7.4.1).

7.2 Baseline Systems

I chose two baseline systems: fast_align (Dyer, Chahuneau, and Smith 2013) and eflomal (Östling and Tiedemann 2016). Both have established themselves as well performing models and were used as baseline models in previous works (Östling and Tiedemann 2016; Jalili Sabet et al. 2020; Steingrímsson, Loftsson, and Way 2021)

7.2.1 fast_align

fast_align is a re-parameterization of the IBM Model 2 which overcomes two problems posed by IBM Models 1 and 2: IBM Model 1 assumes all word orders are equally likely and Model 2 is "vastly overparameterized, making it prone to degenerate behavior on account of overfitting." (Dyer, Chahuneau, and Smith 2013)

fast_align overcomes these problems by implementing a log-linear parameterization. It is ten times faster than IBM Model 4 and outperforms it (Dyer, Chahuneau, and Smith 2013). It has become a popular competitor to Giza++, serves as a baseline system in other works (Östling and Tiedemann 2016; Jalili Sabet et al. 2020), and is even recommended by Philipp Koehn as an alternative to Giza++[2]:

> Another alternative to GIZA++ is fast_align from Dyer et al. It runs much faster, and may even give better results, especially for language pairs without much large-scale reordering. (Koehn 2022, p. 115)

fast_align is extremely fast—computing the word alignments for the around 80,000 sentence pairs takes around 50 seconds on my system[3]. It is well documented and is

[2]For computing the word alignments for Moses SMT, a software package for training statistical machine translation models

[3]MacBook Air (M1, 2020), 8 GB RAM, running macOS Monterey 12.3.1

extremely easy to compile and to operate. All of this makes fast_align a most attractive system to use as a baseline system.

7.2.2 eflomal

eflomal (a.k.a. efmaral[4]) is a system for word alignment using a Bayesian model with Markov Chain Monte Carlo inference (instead of the usual maximum likelihood estimation used in traditional applications of the IBM models for inference, i.e., updating the probabilities). Its performance surpasses fast_align and is on par with Giza++ (Östling and Tiedemann 2016).

7.2.3 Performance

Statistical word alignment models rely heavily on a minimal amount of parallel data before they reach a threshold of good performance. In order to be fair in the evaluation of the baseline systems (fast_align and eflomal) I word-aligned all of the sentence pairs (79,548) with the addition of the 600 annotated sentences from the gold standard (total of 80,148 sentence pairs). I then extracted the alignments of the gold standard for the evaluation.

The performance of the two baseline models on different dataset sizes is presented in Table 7.1. The relation between quality and dataset size is striking.

Compared to results reported in other papers, the results achieved by the models can be considered good. For eflomal, an AER of 0.106 was achieved for English-Swedish (692,662 sentences) and an AER of 0.279 for English-Romanian (48,641 sentences) (Table 2 in Östling and Tiedemann 2016). Trained on 50,000 sentence pairs of German-French, Giza achieves an AER of 0.156; trained on 100,000 an AER of 0.125 is achieved (Table 5 in Och and Ney 2000). The AER of 0.148 achieved for German-Romansh using eflomnal is within this range.

The results are further discussed in Section 7.4.

7.3 SimAlign

I word-aligned the 600 sentences from the gold standard (see Chapter 6) several times using different parameters. I tested the two multilingual embeddings that SimAlign works with out-of-the-box: mBERT[5] and XLM-R (Conneau et al. 2020). mBERT only provides embeddings on a subword level (called WordPiece), while XLM-R works either on the word or the subword level (BPE) (Jalili Sabet et al. 2020) (see also Section 5.3.2).

[4] eflomal is a more memory efficient version of efmaral. See https://github.com/robertostling/efmaral

[5] https://github.com/google-research/bert/blob/master/multilingual.md

Method	Dataset Size	Precision	Recall	F_1	AER
fast_align	80,148	**0.622**	**0.782**	**0.693**	**0.307**
	50k	0.62	0.775	0.689	0.311
	25k	0.603	0.754	0.67	0.33
	10k	0.581	0.727	0.646	0.354
	5k	0.564	0.709	0.628	0.372
	600	0.515	0.644	0.572	0.427
eflomal	80,148	**0.827**	**0.877**	**0.851**	**0.148**
	50k	0.828	0.86	0.844	0.156
	25k	0.812	0.836	0.824	0.176
	10k	0.798	0.805	0.801	0.199
	5k	0.776	0.78	0.778	0.222
	600	0.707	0.724	0.715	0.284

Table 7.1: Word alignment quality of the baseline models, tested on different dataset sizes. Best result per method in bold. "Dataset Size" refers to the number of sentence pairs. The full dataset size is the number of sentence pairs extracted at the time of the experiments (79,548) plus the 600 annotated sentence pairs from the gold standard.

For each embedding and word/subword-level combination, alignments are produced according to each of the three methods (Argmax, Itermax and Match) presented by Jalili Sabet et al. 2020 (see also Section 5.4.1).

7.3.1 Performance

Table 7.2 and Figure 7.1 show the evaluation of performance for word alignments computed with SimAlign with the various methods. For each embedding layer (mBERT and XLM-R), the best score in each column is marked in bold. Generally, the mBERT embeddings perform better. Argmax has the best precision (0.894), which means only 10.6% of the alignments are wrong. However, it has recall measure of only 0.622, which means 37.8% of the alignments are missing. Match has the lowest precision (0.795) but the highest recall (0.767), which makes it the best compromise between precision and recall and it thus has the lowest AER.

These results are reasonable and within the range of reported results for other language pairs using SimAlign. SimAlign's AER ranges between 0.06 for English-French, and 0.39 for English-Hindi. For English-Romanian an AER of 0.29 was achieved, and for English-German an AER of 0.19 (Table 2 in Jalili Sabet et al. 2020). This puts the minimal AER of 0.19 achieved for German-Romansh in a reasonable place within this range.

	Embedding	Level	Method	Percision	Recall	F_1	AER
SimAlign	mBERT	Subword	Argmax	**0.894**	0.622	0.734	0.266
			Itermax	0.832	0.731	0.778	0.222
			Match	0.795	**0.767**	**0.781**	**0.219**
	XLM-R	Word	Argmax	**0.848**	0.399	0.543	0.457
			Itermax	0.767	0.504	0.608	0.391
			Match	0.67	0.647	**0.658**	**0.342**
		Subword	Argmax	0.773	0.488	0.598	0.402
			Itermax	0.671	0.595	0.631	0.369
			Match	0.558	**0.719**	0.628	0.372

Table 7.2: Word alignment quality using SimAlign, with different embeddings and word/sub-word level. Best result per embedding type in bold.

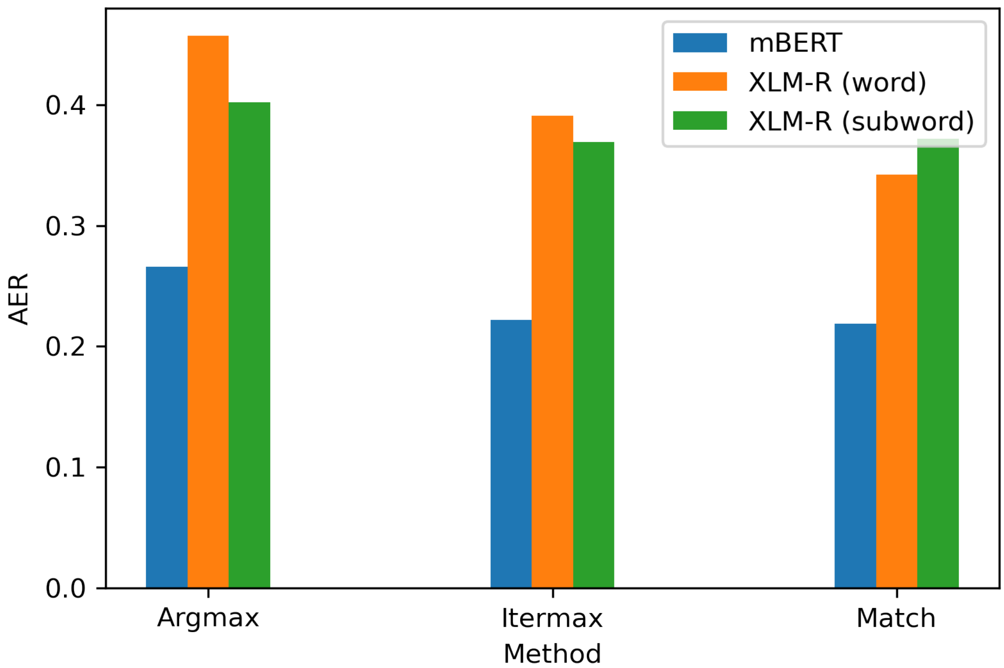

Figure 7.1: Comparison of alignment error rate (AER) (lower is better) for different methods and embeddings using SimAlign

Method	Precision	Recall	F_1	AER
fast_align	0.622	0.782	0.693	0.307
eflomal	**0.827**	**0.877**	**0.851**	**0.148**
SimAlign: mBERT-subword	0.795	0.767	0.781	0.219

Table 7.3: Comparison of the best performance of each of the three methods. The best value in each column is in bold.

7.4 Discussion

Comparing the best performance of SimAlign against the best performance of the baseline systems, SimAlign outperforms fast_align, but is outperformed by eflomal.

Nonetheless, I believe that these results are good news. SimAlign uses embeddings from language models which have never seen Romansh, a scenario which is also referred to as "zero-shot". Despite this fact, the performance is excellent. SimAlign's recall is on par with that of fast_align and its precision is higher than that of fast_align by 17.3 percentage points (27.8%). Also, in the hypothetical case in which we only had the 600 annotated sentences to compute word alignment, SimAlign would have outperformed eflomal as well with an AER of 0.219 (SimAlign) against an AER of 0.284 (eflomal) (cf. Table 7.1).

Further, SimAlign's performance on the language pair German-Romansh (AER of 0.19) does not fall from the performance of SimAlign on English-German sentence pairs (AER of 0.19, Table 2 in Jalili Sabet et al. 2020). This means that the performance in a zero-shot setting with mBERT embeddings for German-Romansh is virtually as good as the performance for a pair of seen languages.

7.4.1 General Problems with Evaluation

It should also be mentioned that each word alignment gold standard has different annotation guidelines and might be more preferable or biased towards one model or the other. For instance, a gold standard which prefers 1-to-1 alignments will reward a model which generates little or no 1-to-many alignments. At the same time, it will penalize the precision measurement of a model that generates 1-to-many alignments, even though these alignments might be correct.

Handling Sure and Possible alignments in a different way in each gold standard might also affect the performance evaluation. Not using Possible alignments will lead to a lower precision value, since it will have lower values for the union of the generated alignments and the possible alignments $|A \cap P|$ (the nominator of the precision measure, see Section 7.1). This will negatively affect precision and will penalize a model that performs better than expected. Labeling many of the alignments as Possible alignments instead of Sure will keep $|S|$ (the denominator of the recall measure) small and thus lead to favorable

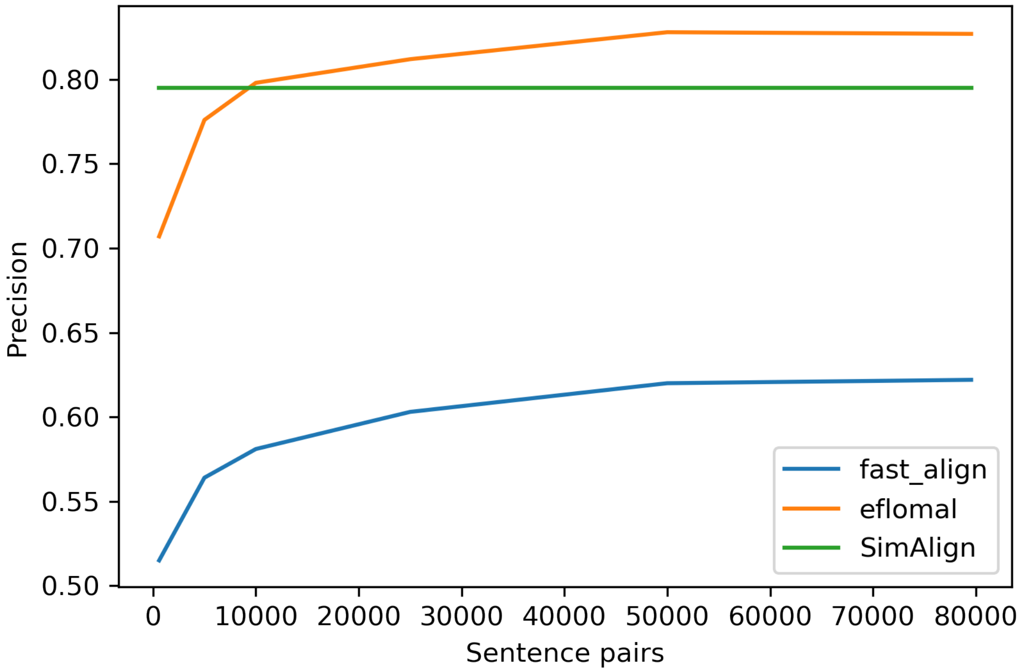

Figure 7.2: Comparing precision between the systems for different dataset sizes.

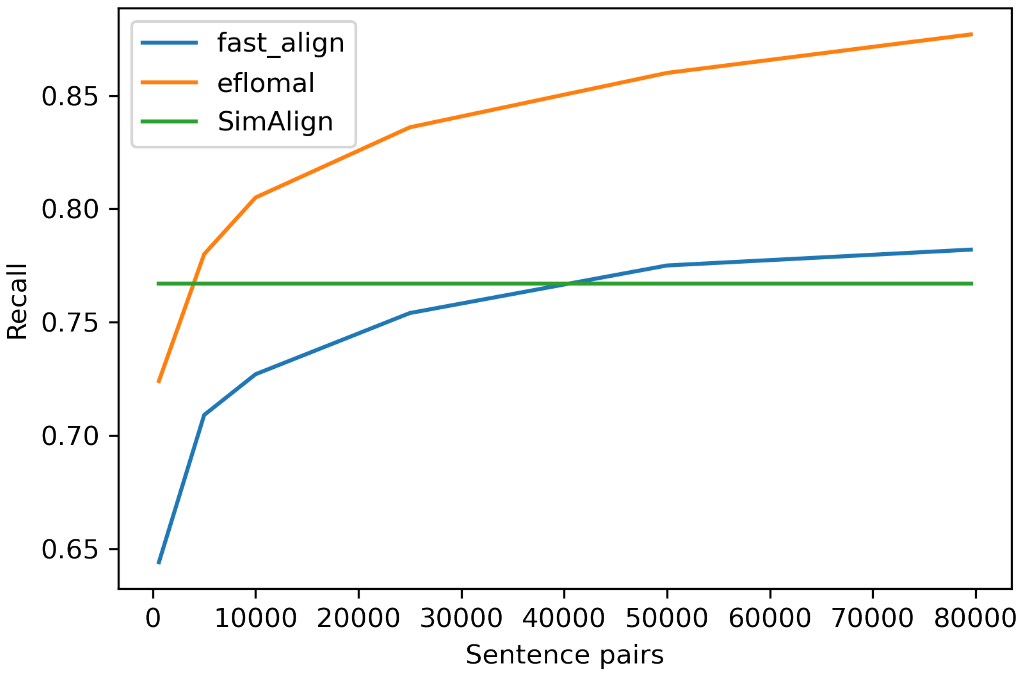

Figure 7.3: Comparing recall between the systems for different dataset sizes.

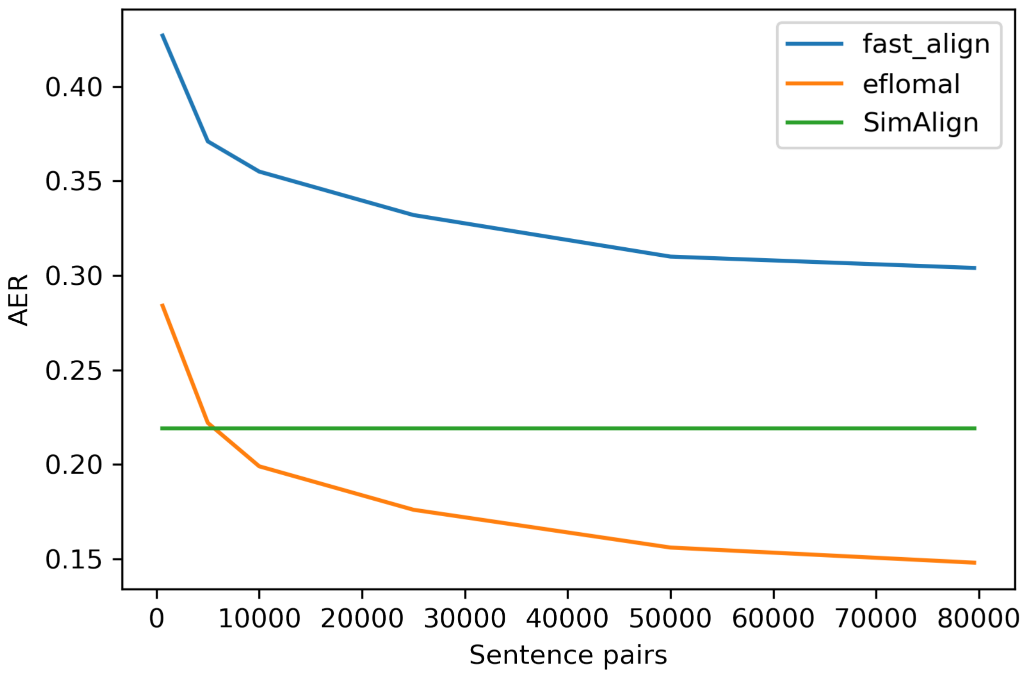

Figure 7.4: Comparing AER between the systems for different dataset sizes.

recall.

Problems with the Gold Standard for German-Romansh

As already explained in Section 6.4, the gold standard I created is not perfect (no second annotator, no Possible alignments). In my annotation guidelines, I preferred 1-to-1 alignments (see Section 6.3.3) and used no Possible label for labeling alignments that might still be correct. Theoretically, not using Possible alignments may explain fast_align's low precision. In theory, it is possible that fast_align generates *correct* 1-to-many alignments which I ignored in my annotations. In that case, we should solely concentrate on recall, which is not affected by Possible alignments. If we were indeed to ignore the other measurements, the difference between fast_align (recall 0.782) and SimAlign (recall 0.767) would be 1.5 percentage points, a difference of 2%, in favor of fast_align.

7.5 Explanation Attempt

Multilingual models such as mBERT show good performance in what is called "cross-lingual zero-shot transfer". It is a scenario in which a pre-trained model is fine-tuned (training taking place after the initial pre-training) on a task, e.g., POS tagging, on one language; the model then carries out this task on a different language (target language) for which it wasn't trained (Deshpande, Talukdar, and Narasimhan 2022). Such models also perform well in a variety of tasks such as POS tagging or NER on **unseen languages** (languages which were not covered by the pre-trained model) such as Faroese, Maltese or Swiss German (Muller et al. 2020).

There is a lack of consensus as to what properties of a language favor performance in such scenarios, i.e., it is not entirely clear *when* zero-shot transfer works. Some suggest sub-word overlap is crucial for good performance (S. Wu and Dredze 2019), while others show that transfer also works well between languages written in different scripts when they are typologically similar[6], meaning sub-word overlap is not a necessary condition (Pires, Schlinger, and Garrette 2019). It was, however, shown that transliterating languages from unseen scripts leads to large gains in performance (Muller et al. 2020).

Deshpande, Talukdar, and Narasimhan 2022 show that zero-shot transfer is possible for different scripts with similar word order, and that the lack of both, on the other hand, hurts performance.

Deshpande, Talukdar, and Narasimhan 2022 also show that zero-shot performance is correlated with alignment between word embeddings, i.e., to what extent the embeddings of different languages share the same geometric shape and are aligned across the same axes: When multilingual word embeddings are learned, the embeddings of the different languages have to be aligned to each other, such that they share similar geometrical shapes and are aligned across the same axes, in order for vectors of similar words across different languages to be next to each other in the vector space (Koehn 2020, pp. 220–223). See Figure 7.5.

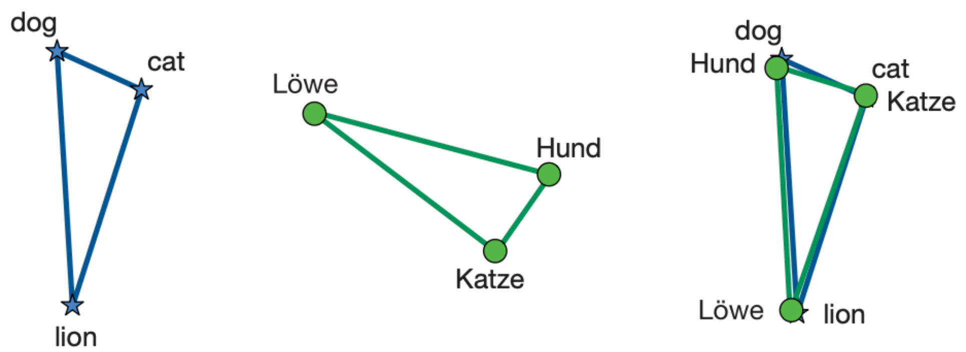

Figure 7.5: Matching up the geometric shape of embedding spaces of words in English and German. Taken from Koehn 2020, p. 223

.

However, in our case, we are not dealing with transfer learning, but simply with the

[6]mBERT fine-tuned for POS tagging in Urdu (Arabic script) achieved 91% accuracy on Hindi (Devanagari script) (Pires, Schlinger, and Garrette 2019). Both languages are mutually intelligible and are considered variants of a single language—Hindustani (The Editors of Encyclopaedia Britannica 2018).

leverage of embeddings for measuring word similarity.

Since multilingual models process tokens at the sub-word level, they work in an open vocabulary setting and can process any language, even languages that aren't part of the pretraining data (providing the character set is part of the pretraining data) (Muller et al. 2020).

According to the mBERT's performance on unseen languages, Muller et al. 2020 put these unseen languages into three categories: Easy, Intermediate and Hard. Muller et al. 2020 ascribe the differences in mBERT's performance on these languages to two things: close relatedness to languages used during pretraining; and the unseen languages using the same script as those closely related languages which were seen during pretraining.

Since Romansh shares a high similarity, not only in script, but also typologically, with other Romance, as well as other European languages[7], which are a major part of the training data for mBERT, it should not be surprising that similarity-based word alignment using word embeddings from mBERT works well.

7.6 Summary

I evaluated the performance of two statistical baseline models (fast_align and eflomal) as well as the performance of SimAlign, a similarity-based word alignment model, on the language pair German-Romansh. SimAlign computed the word similarity using multilingual word embeddings from two language models: mBERT and XLM-R. Neither of the models had seen Romansh during training, i.e., we are dealing with a zero-shot setting. The evaluation was done using a gold standard of 600 annotated sentence pairs in German-Romansh, which I had created myself (see Chapter 6). SimAlign outperformed fast_align, but not eflomal (see Table 7.3).

SimAlign's performance, although worse the eflomal's performance, is on par with that of fast_align and is generally promising. It shows that mBERT's embeddings can be used in a zero-shot setting (Romansh was not part of the training data; mBERT has never seen Romansh before) for the task of word alignment and may give future students and/or researchers the impulse to test the performance of mBERT (or other multilingual models) on Romansh in other tasks, such as information extraction, question answering, sentiment analysis, POS tagging etc.

For a discussion of the differences between the systems in some specific cases, see Appendix B.

[7]European languages from different languages families (Germanic, Romance, Slavic) were shown to display high similarity to each other and to form a so-called *Sprachbund*, dubbed Standard Average European (Haspelmath 2001).

Chapter 8

Concluding Words

8.1 Goals

The goals of this work were twofold:

- Enlarge the amount of digital resources that are available for the Romansh language;
- Evaluate a novel, similarity-based word alignment method, which uses word embeddings, on the language pair German-Romansh.

8.2 Corpus Compliation

In order to achieve both goals, I first had to collect data. I chose to collect the press releases published by the *Standeskanzlei* of the canton of Graubünden from 1997 until today. These press releases have been released in the three official languages of the canton: German, Italian and Romansh. I aligned the press releases (henceforth *documents*) using URL matching when possible, or reverted to a simple heuristic (three releases from the same day in three different languages are mutual translations). The documents (aligned and not aligned), are saved both as JSON files and in a SQLite database; both allow fast and simple queries.

I proceeded to align the sentences using hunalign (Varga et al. 2005), a fast length- and dictionary-based method for aligning sentences. After filtering noise (duplicates and misalignments), as well as sentences containing only phone numbers, URLs or email addresses, I was able to extract around 80,000 unique sentence pairs for each language combination (German-Romansh, German-Italian, Romansh-Italian).

I will be glad to provide the corpus that I collected, as well as the aligned sentence pairs, to other students for further research and experimentation[1].

[1]In case you would like to use this corpus, please consult the copyright notice on `https://www.gr.ch/de/Seiten/Impressum.aspx` before publicly releasing it or parts thereof.

8.3 Gold Standard

In order to evaluate word alignment systems, a gold standard is needed (Koehn 2009, p. 115). In the context of word alignment, a gold standard is a collection of sentence pairs manually annotated for word correspondences. Since there is no gold standard for German-Romansh, I annotated word correspondences in 600 sentences (see Chapter 6). I will gladly provide my annotations to other students for further experiments and research, as well as for second annotation.

8.4 Evaluation

I compared the performance of statistical word alignment methods—fast_align (Dyer, Chahuneau, and Smith 2013) and eflomal (Östling and Tiedemann 2016)—with the novel similarity- and embeddings-based method SimAlign (Jalili Sabet et al. 2020). SimAlign's performance is on par with fast_align, but was outperformed by eflomal. This still shows that SimAlign is a viable method for computing word alignments for German-Romansh. Considering the fact that the multilingual embeddings used by SimAlign (mBERT) do not contain embeddings for Romansh (a.k.a. zero-shot setting), I believe that these results are very promising.

8.5 Future

The corpus I collected might be used by future students in a variety of ways. One way that comes to mind is training a neural machine translation model using the $\sim 80,000$ sentence pairs I extracted and testing a variety of methods for enriching using monolingual data, such as back-translation (an automatic translation of the monolingual target text into the source language) (Sennrich, Haddow, and Birch 2016a). See also R. Wang et al. 2021.

Another possibility would be to fine-tune or extend mBERT with Romansh data. Enlarging the vocabulary of mBERT to accomodate an unseen language and then continue training the model on this language was shown to significantly improve performance in NER tasks for that lanauge compared to a zero-shot setting (Z. Wang et al. 2020).

It would also be desirable that a future student would repeat my annotations of the 600 sentences as a second annotator. This would make the gold standard more sensible, reliable and acceptable, and would introduce a set of Possible alignments to it (see Section 6.4).

Glossary

Graubünden The Canton of Grisons. 1, 6, 11, 60

HTML Hypertext Markup Language. A language containing display instructions for web browsers and the format in which web pages are usually saved . 12

JSON JavaScript Object Notation. A format for organizing data in a hierarchical form. 13

Standeskanzlei State Chancellery of Grisons. 11, 60

URL Uniform Resource Locator. A reference to an internet resource, a web address. 12

Acronyms

AER alignment error rate. 41, 50, 52, 53, 54, 55, 57, 65

EM expectation-maximization. 33

gen genitive. 47

HTML Hypertext Markup Language. 12, 13, 17

JSON JavaScript Object Notation. 13, 14, 15, 17, 60, 62, 75

NER named entity recognition. 2, 40, 58

NLP natural language processing. 2, 9

NMT neural machine translation. 26, 29

part participle. 47

pl plural. 47

POS part of speech. 2, 9, 40, 45, 57, 58, 59

pres present. 47

SMT statistical machine translation. 25, 26

URL Uniform Resource Locator. 29, 60

List of Tables

List of Figures

List of Listings

Bibliography

Ahrenberg, Lars (2007). *LinES 1.0 Annotation: Format, Contents and Guidelines*. Tech. rep. Linköping University. URL: https://www.ida.liu.se/~larah03/transmap/Corpus/guidelines.pdf.

Artetxe, Mikel, Gorka Labaka, and Eneko Agirre (July 2018). "A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings." In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, pp. 789–798. DOI: 10.18653/v1/P18-1073. URL: https://aclanthology.org/P18-1073.

Artetxe, Mikel and Holger Schwenk (Sept. 2019). "Massively Multilingual Sentence Embeddings for Zero-Shot Cross-Lingual Transfer and Beyond." In: *Transactions of the Association for Computational Linguistics* 7, pp. 597–610. ISSN: 2307-387X. DOI: 10.1162/tacl_a_00288. eprint: https://direct.mit.edu/tacl/article-pdf/doi/10.1162/tacl_a_00288/1923278/tacl_a_00288.pdf. URL: https://doi.org/10.1162/tacl%5C_a%5C_00288.

Bauer, Laurie (1988). *Introducing Linguistic Morphology*. Edinburgh University Press. ISBN: 0-855224-582-3.

Bender, Emily M. (2019). *The #BenderRule: On Naming the Languages We Study and Why It Matters*. Accessed: 10 August 2022. URL: https://thegradient.pub/the-benderrule-on-naming-the-languages-we-study-and-why-it-matters/.

Bird, Steven, Edward Loper, and Ewan Klein (2009). *Natural Language Processing with Python*. O'Reilly Media Inc.

Bossong, Georg (1998). *Die Romanischen Sprachen: Eine vergleichende Einführung*. Hamburg: Helmut Buske Verlag. ISBN: 978-3-87548-518-9.

Brown, Peter F. et al. (1993). "The Mathematics of Statistical Machine Translation: Parameter Estimation." In: *Computational Linguistics* 19.2, pp. 263–311. URL: https://aclanthology.org/J93-2003.

Bundesamt für Statistik (2020). *Hauptsprachen in der Schweiz - 2020*. URL: https://www.bfs.admin.ch/bfs/de/home/statistiken/bevoelkerung/sprachen-religionen/sprachen.assetdetail.21344032.html (visited on 06/17/2022).

Caduff, Renzo, Uorschla N. Caprez, and Georges Darms (2008). *Grammatica per l'instrucziun dal rumantsch grischun*. Freiburg/Fribourg/Friburg: Seminari da rumantsch da l'Universitad da Friburg. URL: https://doc.rero.ch/record/9712/files/Gramminstr.pdf.

Cathomas, Bernard (2012). "Geschichte und Gegenwart des Rätoromanischen in Graubünden und im Rheintal." In: ed. by Gerhard Wanner and Georg Jäger. Chur: Desertina. Chap. Sprachen fallen nicht vom Himmel. Zur Sprachplanung in der Rätoromania, pp. 125–147.

Chen, Stanley F. (June 1993). "Aligning Sentences in Bilingual Corpora Using Lexical Information." In: *31st Annual Meeting of the Association for Computational Linguistics*. Columbus, Ohio, USA: Association for Computational Linguistics, pp. 9–16. DOI: 10.3115/981574.981576. URL: https://aclanthology.org/P93-1002.

Clematide, Simon et al. (2018). "A multilingual gold standard for translation spotting of German compounds and their corresponding multiword units in English, French, Italian and Spanish." In: *Multiword Units in Machine Translation and Translation Technology*. Ed. by Ruslan Mitkov et al. John Benjamins, pp. 125–145. DOI: https://doi.org/10.1075/cilt.341.

Conneau, Alexis et al. (July 2020). "Unsupervised Cross-lingual Representation Learning at Scale." In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 8440–8451. DOI: 10.18653/v1/2020.acl-main.747. URL: https://aclanthology.org/2020.acl-main.747.

Dazzi, Anna-Alice (2012). "Geschichte und Gegenwart des Rätoromanischen in Graubünden und im Rheintal." In: ed. by Gerhard Wanner and Georg Jäger. Chur: Desertina. Chap. Die verschiedenen Aktivitäten der Lia Rumantsche zur Erhaltung und Förderung des Rätoromanischen, pp. 117–124.

Deshpande, Ameet, Partha Talukdar, and Karthik Narasimhan (July 2022). "When is BERT Multilingual? Isolating Crucial Ingredients for Cross-lingual Transfer." In: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Seattle, United States: Association for Computational Linguistics, pp. 3610–3623. DOI: 10.18653/v1/2022.naacl-main.264. URL: https://aclanthology.org/2022.naacl-main.264.

Devlin, Jacob et al. (2018). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. DOI: 10.48550/ARXIV.1810.04805. URL: https://arxiv.org/abs/1810.04805.

Dreyer, Hilke and Richard Schmitt (2009). *Lehr- und Übungsbuch der detuschen Grammatik. Die Gelbe aktuell.* Hueber Verlag. ISBN: 978-3-19-307255-9.

Dyer, Chris, Victor Chahuneau, and Noah A. Smith (June 2013). "A Simple, Fast, and Effective Reparameterization of IBM Model 2." In: *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics:*

Human Language Technologies. Atlanta, Georgia: Association for Computational Linguistics, pp. 644–648. URL: https://aclanthology.org/N13-1073.

Gale, William A. and Kenneth W. Church (June 1991). "A Program for Aligning Sentences in Bilingual Corpora." In: *29th Annual Meeting of the Association for Computational Linguistics*. Berkeley, California, USA: Association for Computational Linguistics, pp. 177–184. DOI: 10.3115/981344.981367. URL: https://aclanthology.org/P91-1023.

Graën, Johannes, Dominique Sandoz, and Martin Volk (May 2017). "Multilingwis2 – Explore Your Parallel Corpus." In: *Proceedings of the 21st Nordic Conference on Computational Linguistics, NoDaLiDa*. Linköping Electronic Conference Proceedings 131. Linköping University Electronic Press, Linköpings universitet, pp. 247–250. ISBN: 978-91-7685-601-7. URL: https://doi.org/10.5167/uzh-137129.

Haiman, John and Paola Benincà (1992). *The Rhaeto-Romance Languages*. London and New York: Routledge. ISBN: 0-415-04194-5.

Haspelmath, Martin (2001). "The European linguistic area: Standard Average European." In: *Language Typology and Language Universals* 20.2, pp. 1492–1510. DOI: 10.1515/9783110171549.2.14.1492.

Hedderich, Michael A. et al. (June 2021). "A Survey on Recent Approaches for Natural Language Processing in Low-Resource Scenarios." In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, pp. 2545–2568. DOI: 10.18653/v1/2021.naacl-main.201. URL: https://aclanthology.org/2021.naacl-main.201.

Holmqvist, Maria and Lars Ahrenberg (May 2011). "A Gold Standard for English-Swedish Word Alignment." In: *Proceedings of the 18th Nordic Conference of Computational Linguistics (NODALIDA 2011)*. Riga, Latvia: Northern European Association for Language Technology (NEALT), pp. 106–113. URL: https://aclanthology.org/W11-4615.

Jalili Sabet, Masoud et al. (Nov. 2020). "SimAlign: High Quality Word Alignments Without Parallel Training Data Using Static and Contextualized Embeddings." In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, pp. 1627–1643. DOI: 10.18653/v1/2020.findings-emnlp.147. URL: https://aclanthology.org/2020.findings-emnlp.147.

Jespersen, Otto (1924). *The Philosophy of Grammar*. New York: W.W. Norton & Company Inc. ISBN: 978-0-393-00307-9.

Jurafsky, Daniel and James H. Martin (2019). *Speech and Language Processing. An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Third Edition Draft. URL: https://web.stanford.edu/~jurafsky/slp3/.

Khayrallah, Huda and Philipp Koehn (July 2018). "On the Impact of Various Types of Noise on Neural Machine Translation." In: *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*. Melbourne, Australia: Association for Computational Linguistics, pp. 74–83. DOI: 10.18653/v1/W18-2709. URL: https://aclanthology.org/W18-2709.

Koehn, Philipp (2009). *Statistical Machine Translation*. Cambridge University Press.

— (2020). *Neural Machine Translation*. Cambridge University Press. DOI: 10.1017/9781108608480.

— (Apr. 2022). *Moses. Statistical Machine Translation System. User Manual and Code Guide*. URL: http://www2.statmt.org/moses/manual/manual.pdf.

Koehn, Philipp, Franz Josef Och, and Daniel Marcu (2003). "Statistical Phrase-Based Translation." In: *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*. NAACL '03. Edmonton, Canada: Association for Computational Linguistics, pp. 48–54. DOI: 10.3115/1073445.1073462. URL: https://doi.org/10.3115/1073445.1073462.

Kofler, Michael (2019). *Python. Der Grundkurs*. 1st ed. Bonn: Rheinwerk Verlag. ISBN: 978-3-8362-6679-6.

Lambert, Patrik et al. (2005). "Guidelines for Word Alignment Evaluation and Manual Alignment." In: *Language Resource and Evalution* 39, pp. 267–285. DOI: 10.1007/s10579-005-4822-5.

Liver, Ricarda (1999). *Rätoromanisch: Eine Einführung in das Bündnerromanische*. Tübingen: Narr.

Melamed, I. Dan (1998). "Annotation Style Guide for the Blinker Project." In: *CoRR* cmp-lg/9805004. URL: http://arxiv.org/abs/cmp-lg/9805004.

Mihalcea, Rada and Ted Pedersen (2003). "An Evaluation Exercise for Word Alignment." In: *Proceedings of the HLT-NAACL 2003 Workshop on Building and Using Parallel Texts: Data Driven Machine Translation and Beyond*, pp. 1–10. URL: https://aclanthology.org/W03-0301.

Moore, Bob (Oct. 2002). "Fast and Accurate Sentence Alignment of Bilingual Corpora." In: Springer-Verlag. URL: https://www.microsoft.com/en-us/research/publication/fast-and-accurate-sentence-alignment-of-bilingual-corpora/.

Muller, Benjamin et al. (2020). *When Being Unseen from mBERT is just the Beginning: Handling New Languages With Multilingual Language Models*. DOI: 10.48550/ARXIV.2010.12858. URL: https://arxiv.org/abs/2010.12858.

Müller, Mathias, Annette Rios, and Rico Sennrich (Oct. 2020). "Domain Robustness in Neural Machine Translation." In: *Proceedings of the 14th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*. Virtual:

Association for Machine Translation in the Americas, pp. 151–164. URL: https://aclanthology.org/2020.amta-research.14.
Och, Franz Josef and Hermann Ney (Oct. 2000). "Improved Statistical Alignment Models." In: *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*. Hong Kong: Association for Computational Linguistics, pp. 440–447. DOI: 10.3115/1075218.1075274. URL: https://aclanthology.org/P00-1056.
— (2003). "A Systematic Comparison of Various Statistical Alignment Models." In: *Computational Linguistics* 29.1, pp. 19–51. DOI: 10.1162/089120103321337421. URL: https://aclanthology.org/J03-1002.
Östling, Robert and Jörg Tiedemann (Oct. 2016). "Efficient word alignment with Markov Chain Monte Carlo." In: *Prague Bulletin of Mathematical Linguistics* 106, pp. 125–146. URL: http://ufal.mff.cuni.cz/pbml/106/art-ostling-tiedemann.pdf.
Pires, Telmo, Eva Schlinger, and Dan Garrette (July 2019). "How Multilingual is Multilingual BERT?" In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 4996–5001. DOI: 10.18653/v1/P19-1493. URL: https://aclanthology.org/P19-1493.
Price, Glanville (2008). *A Comprehensive French Grammar*. Blackwell Publishing. ISBN: 978-1-4051-5385-0.
Saussure, Ferdinand de (1959). *Course in General Linguistics*. Ed. by Charles Bally and Albert Sechehaye. Trans. by Wade Baskin. New York: Philosophical Library. URL: https://ia902704.us.archive.org/35/items/courseingenerall00saus/courseingenerall00saus.pdf.
Scherrer, Yves and Bruno Cartoni (May 2012). "The Trilingual ALLEGRA Corpus: Presentation and Possible Use for Lexicon Induction." In: *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*. Istanbul, Turkey: European Language Resources Association (ELRA), pp. 2890–2896. URL: http://www.lrec-conf.org/proceedings/lrec2012/pdf/685_Paper.pdf.
Schmid, Heinrich (1982). *Richtlinien für die Gestaltung einer gesamtbündnerromanischen Schriftsprache RUMANTSCH GRISCHUN*. Lia Rumantscha. Chur.
Sennrich, Rico, Barry Haddow, and Alexandra Birch (Aug. 2016a). "Improving Neural Machine Translation Models with Monolingual Data." In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, pp. 86–96. DOI: 10.18653/v1/P16-1009. URL: https://aclanthology.org/P16-1009.
— (Aug. 2016b). "Neural Machine Translation of Rare Words with Subword Units." In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Lin-

guistics, pp. 1715–1725. DOI: 10.18653/v1/P16-1162. URL: https://aclanthology.org/P16-1162.

Sennrich, Rico and Martin Volk (Oct. 2010). “MT-based Sentence Alignment for OCR-generated Parallel Texts.” In: *Proceedings of the 9th Conference of the Association for Machine Translation in the Americas: Research Papers*. Denver, Colorado, USA: Association for Machine Translation in the Americas. URL: https://aclanthology.org/2010.amta-papers.14.

— (May 2011). “Iterative, MT-based Sentence Alignment of Parallel Texts.” In: *Proceedings of the 18th Nordic Conference of Computational Linguistics (NODALIDA 2011)*. Riga, Latvia: Northern European Association for Language Technology (NEALT), pp. 175–182. URL: https://aclanthology.org/W11-4624.

Simard, Michel and Pierre Plamondon (Oct. 1996). “Bilingual sentence alignment: balancing robustness and accuracy.” In: *Conference of the Association for Machine Translation in the Americas*. Montreal, Canada. URL: https://aclanthology.org/1996.amta-1.14.

Standeskanzlei Graubünden (2022). *State Chancellery of Grisons*. URL: https://www.gr.ch/EN/institutions/administration/staka/Seiten/Home.aspx (visited on 06/29/2022).

Steingrímsson, Steinþór, Hrafn Loftsson, and Andy Way (2021). “CombAlign: a Tool for Obtaining High-Quality Word Alignments.” In: *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*. Reykjavik, Iceland (Online): Linköping University Electronic Press, Sweden, pp. 64–73. URL: https://aclanthology.org/2021.nodalida-main.7.

TextShuttle AG (June 2022). *Automatisierte Übersetzung für Rätoromanisch*. Accessed: 10 August 2022. URL: https://www.textshuttle.ai/news-articles/ratoromanisch.

The Editors of Encyclopaedia Britannica (Nov. 2018). *Hindustani Language*. Accessed: 12 August 2022. URL: https://www.britannica.com/topic/Hindustani-language.

Thompson, Brian and Philipp Koehn (Nov. 2019). “Vecalign: Improved Sentence Alignment in Linear Time and Space.” In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pp. 1342–1348. DOI: 10.18653/v1/D19-1136. URL: https://aclanthology.org/D19-1136.

Tscharner, Gion and Duri Denoth (2022). *Grammatikteil des Vallader / Grammatica valladar*. URL: http://www.udg.ch/dicziunari/files/grammatica_vallader.pdf (visited on 06/07/2022).

Valär, Rico Franc (2012). “Geschichte und Gegenwart des Rätoromanischen in Graubünden und im Rheintal.” In: ed. by Gerhard Wanner and Georg Jäger. Chur: Desertina.

Chap. Wie die Anerkennung des Rätoromanischen die Schweiz einte. Einige Hintergründe zur Volksabstimmung vom 20. Februar 1938, pp. 101–116.

Varga, D. et al. (2005). "Parallel corpora for medium density languages." In: *Proceedings of the RANLP 2005*, pp. 590–596.

Volk, Martin and Simon Clematide (Oct. 2014). "Detecting Code-Switching in a Multilingual Alpine Heritage Corpus." In: *Proceedings of the First Workshop on Computational Approaches to Code Switching*. Doha, Qatar: Association for Computational Linguistics, pp. 24–33. DOI: 10.3115/v1/W14-3903. URL: https://aclanthology.org/W14-3903.

Vronis, Jean and Philippe Langlais (2000). *Evaluation of parallel text alignment systems - The ARCADE project*.

Wang, Rui et al. (2021). *A Survey on Low-Resource Neural Machine Translation*. DOI: 10.48550/ARXIV.2107.04239. URL: https://arxiv.org/abs/2107.04239.

Wang, Zihan et al. (Nov. 2020). "Extending Multilingual BERT to Low-Resource Languages." In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, pp. 2649–2656. DOI: 10.18653/v1/2020.findings-emnlp.240. URL: https://aclanthology.org/2020.findings-emnlp.240.

Weibel, Manuela (2014). "Aufbau paralleler Korpora und Implementierung eines wortalignierten Suchsystems fuür Deutsch – Rumantsch Grischun." MA thesis. University of Zurich.

Wikipedia contributors (2022). *ISO 8601 — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=ISO_8601&oldid=1095673391. Accessed: 30 June 2022.

Wu, Shijie and Mark Dredze (Nov. 2019). "Beto, Bentz, Becas: The Surprising Cross-Lingual Effectiveness of BERT." In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pp. 833–844. DOI: 10.18653/v1/D19-1077. URL: https://aclanthology.org/D19-1077.

Wu, Yonghui et al. (2016). *Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation*. DOI: 10.48550/ARXIV.1609.08144.

Appendix A

JSON examples

Below are examples for the JSON files contatining the press releases. Listing A.1 is an example for a JSON file containing the press releases prior to alignment. Each entry is a single document. Listing A.2 is an example for a JSON file containing the press releases after alignment. Each entry contains three documents which are mutual translations. See also Chapter 3.


```json
{
    "2": {
        "id": "17811",
        "orig_file": "../html/2008/2008_17811_DE.html",
        "lang": "DE",
        "title": "Stiftung für Innovation, Entwicklung und Forschung Graubünden nimmt ihre Tätigkeit auf",
        "date": "31.01.2008",
        "content": "Die im Dezember 2007 gegründete Stiftung für Innovation, Entwicklung  und Forschung Graubünden hat ihre Tätigkeit im Januar 2008 aufgenommen.  ..."
    },
    "3": {
        "id": "17812",
        "orig_file": "../html/2008/2008_17812_IT.html",
        "lang": "IT",
        "title": "La Commissione preparatoria del Gran Consiglio accoglie con favore l'aggregazione dei Comuni di Feldis, Scheid, Trans e Tomils nel Comune di Tomils",
        "date": "04.07.2008",
        "content": "Dopo lunghi e intensi lavori preparatori delle autorità dei Comuni  interessati, il 13
```

```
        ↪ dicembre 2007 gli aventi diritto di voto di tutti
        ↪ e  quattro i Comuni di Feldis/Veulden, Scheid,
        ↪ Trans e Tumegl/Tomils hanno  accolto a larga
        ↪ maggioranza la convenzione sulla nuova
        ↪ aggregazione nel  Comune di Tomils.  ..."
    },
    "4": {
        "id": "17813",
        "orig_file": "../html/2008/2008_17813_RM.html",
        "lang": "RM",
        "title": "La cumissiun predeliberanta dal cussegl grond
        ↪ beneventa la fusiun da las vischnancas da Veulden,
        ↪ da Sched, da Tràn e da Tumegl a la vischnanca da
        ↪ Tumegl",
        "date": "04.07.2008",
        "content": "Suenter lavurs preliminaras intensivas che
        ↪ las autoritads da las  vischnancas pertutgadas han
        ↪ prestà durant divers onns han las votantas  ed ils
        ↪ votants da tut las quatter vischnancas da Veulden,
        ↪ da Sched, da  Tràn e da Tumegl acceptà ils 13 da
        ↪ december 2007 cun gronda maioritad en  tut las
        ↪ vischnancas la cunvegna da fusiun a la nova
        ↪ vischnanca da  Tumegl.  ..."
    }
}
```

Listing A.1: Example for a JSON file containing the press releases extracted from the HTML files.

```
{
    "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 A.2: Example for a JSON file containing aligned documents

Appendix B

Alignment Examples

I would like to shortly compare the alignments computed by eflomal and SimAlign (mBERT, subword level, Itermax method, cf., Sections 5.4.1 and 7.3) with my annotations from the gold standard, especially regarding some of the examples I mentioned in Chapter 6: Gold Standard for handling ambiguous cases. I will consider the following cases: German compounds, German preterite and perfect vs. Romansh perfect and Romansh double negation. Please refer to Section 6.3.4: Gold Standard–Examples for more details.

In all of the examples below, **filled green squares** are the **gold standard**, **circles** are alignments produced by **SimAlign** and **boxes** are alignments produced **eflomal**.

The plots were created using a script provided on GitHub[1] accompanying SimAlign (Jalili Sabet et al. 2020).

B.1 Compounds

First, I would like to see how eflomal and SimAlign deal with aligning German compounds.

eflomal seems to be doing a better job creating 1-to-many alignments for compounds. In Figure B.1, eflomal aligns the German word *Fachhochschule* (*technical college*) correctly to Romansh *Scola auta spzialisada*, whereas SimAlign only aligns it to *Scola* ("school"). However, both models correctly align German *Ostschweizer* ("eastern Swiss") to Romansh *Svizra Oreintala*.

Figure B.2 shows a similar case. The German compound *Grundversorgungsauftrag* ("basic services mission") is aligned by eflomal to two words in Romansh: *incumbensa* and *provediment*. But it leaves *basa* wrongly unaligned. The compound *Nationalstrassen* ("national roads") is correctly aligned to *vias naziunalas* by eflomal. SimAlign again only aligns the first word of the corresponding Romansh words: *incumbensa* and *vias*, respectively.

[1] https://github.com/cisnlp/simalign/blob/master/scripts/visualize.py

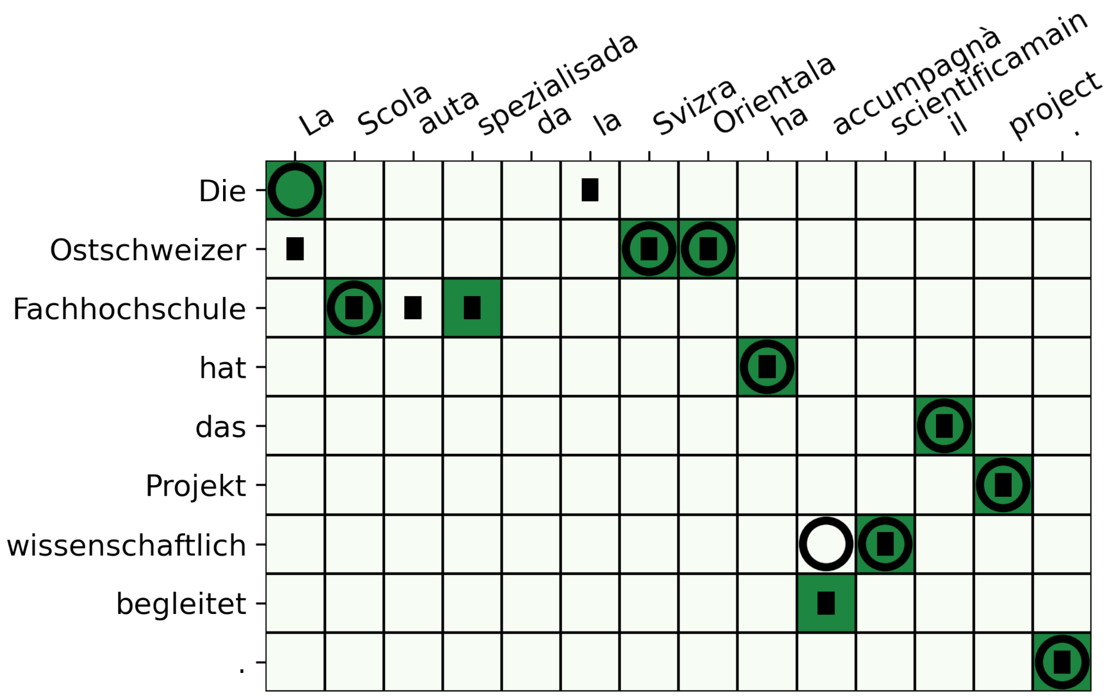

Figure B.1: Word alignment example for the case of perfect tense in German and Romansh

In yet another case (Figure B.3), both models succeed in creating a 1-to-2 alignment by aligning the German word *Leitbild* ("role model") to Romansh *model directiv*. However, eflomal fails to align German *departmentsübergreifend* ("inter-departmental") to Romansh *interdepartamental*, although this would have been a 1-to-1 alignment. I am assuming that this is due to this word appearing only once in the entire corpus. SimAlign succeeds here, probably due to these words (or parts of them) having appeared enough times in the monolingual training data of mBERT. The German compound *Aufgabenfeld* ("field of duties") is aligned by SimAlign only to the first word again: *champ* ("field"). eflomal fails here completely.

To summarize, it seems eflomal generally does a better job creating 1-to-many alignments for German compounds. However, a much larger sample size would be needed to reach definite conclusions.

B.2 Perfect–Perfect

Figure B.4 shows an example for aligning the German perfect with the Romansh perfect. The German and the Romansh auxiliaries *hat* and *ha* should be aligned to each other, as well as the German and the Romansh participles *verabschiedet* and *deliberà*. SimAlign's alignment are in accord with the gold standard, while eflomal aligned Romansh *deliberà* to German *hat*, leaving German *verabschiedet* unaligned. However, in another case (Figure

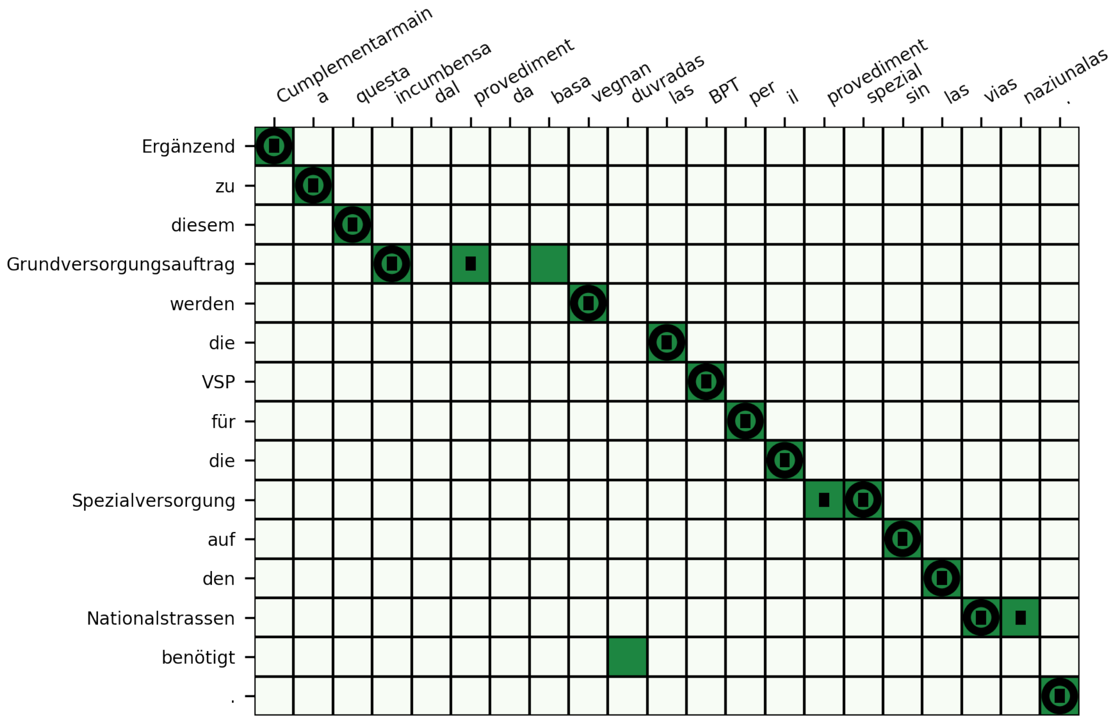

Figure B.2: Word alignment example with compounds.

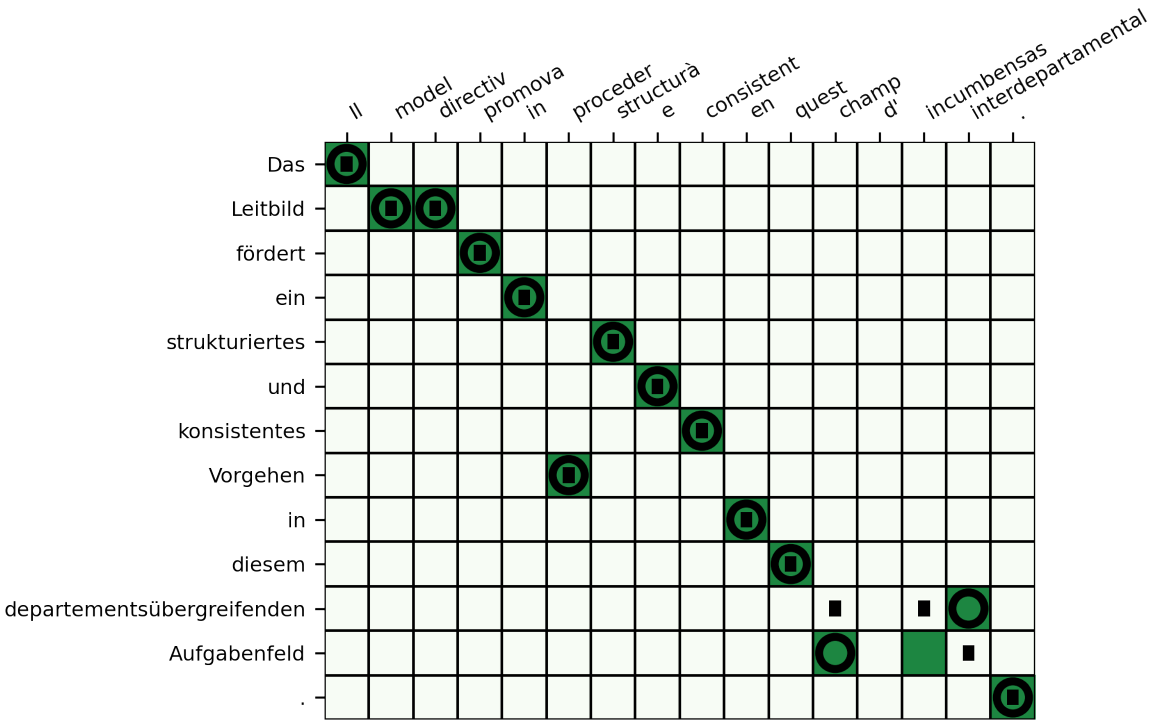

Figure B.3: Word alignment example with compounds.

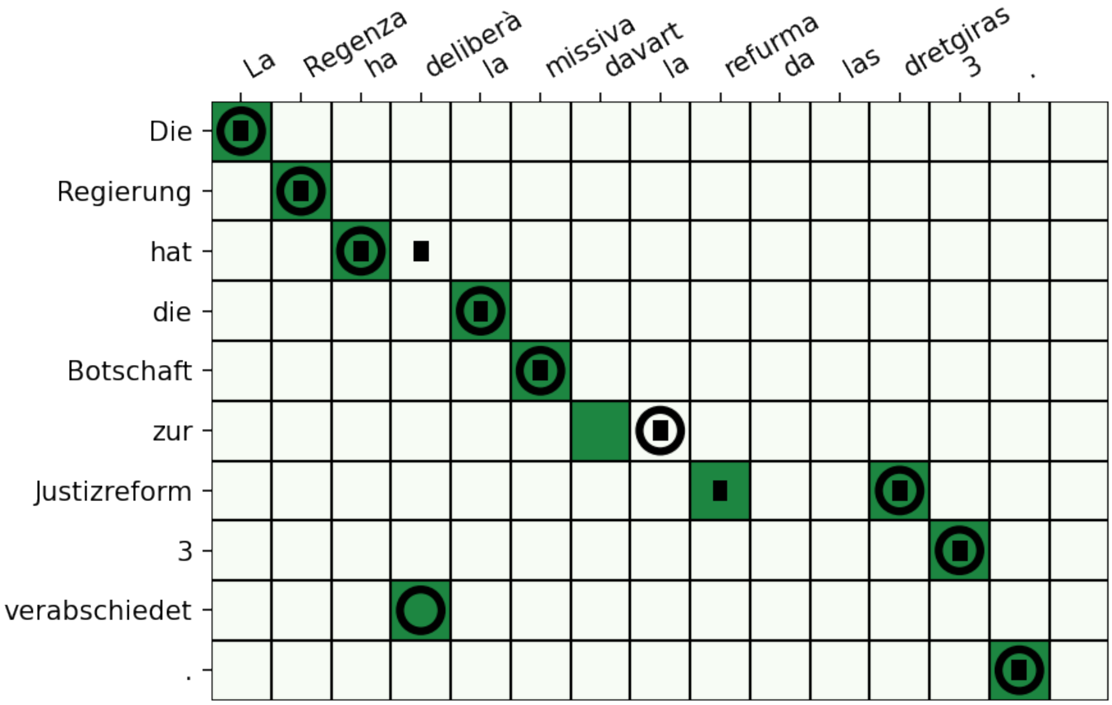

Figure B.4: Word alignment example for the case of perfect tense in German and Romansh

B.1), eflomal correctly aligned the German participle to the Romansh participle, whereas SimAlign didn't. It would be interesting to test this on a larger scale and see which system is more consistent regarding this.

B.3 German Preterite–Romansh Perfect

In the matter of aligning the German preterite with Romansh perfect, eflomal creates a 1-to-2 alignment, connecting both the auxiliary *han* and the participle *visità* to the German preterite *besichtigten* (Example B.5), an alignment which is not even acceptable, but also desirable, but which I chose to avoid in my annotations due to my preference of 1-to-1 alignments. However, in a different case (Example B.6), eflomal failed to align the participle, which is lexically the more important part, and left it unaligned . SimAlign successfully aligns the German preterite to the Romansh participle in the first case, but fails as well in the second case. In the case of preterite–perfect, there is no clear advantage of any of the models over the other.

Example B.7 presents an even more challenging case. Here we are dealing with a separable German verb in preterite *nahm* ... *auf* ("start, open") , which is translated to the Romansh perfect *ha* ... *avert* ("has ... opened"). The gold standard stipulates that German *nahm* ... *auf* should be aligned to Romansh *avert*, leaving the Romansh auxiliary *ha* unaligned. However, both models align German *nahm* to Romansh *ha*. SimAlign leaves

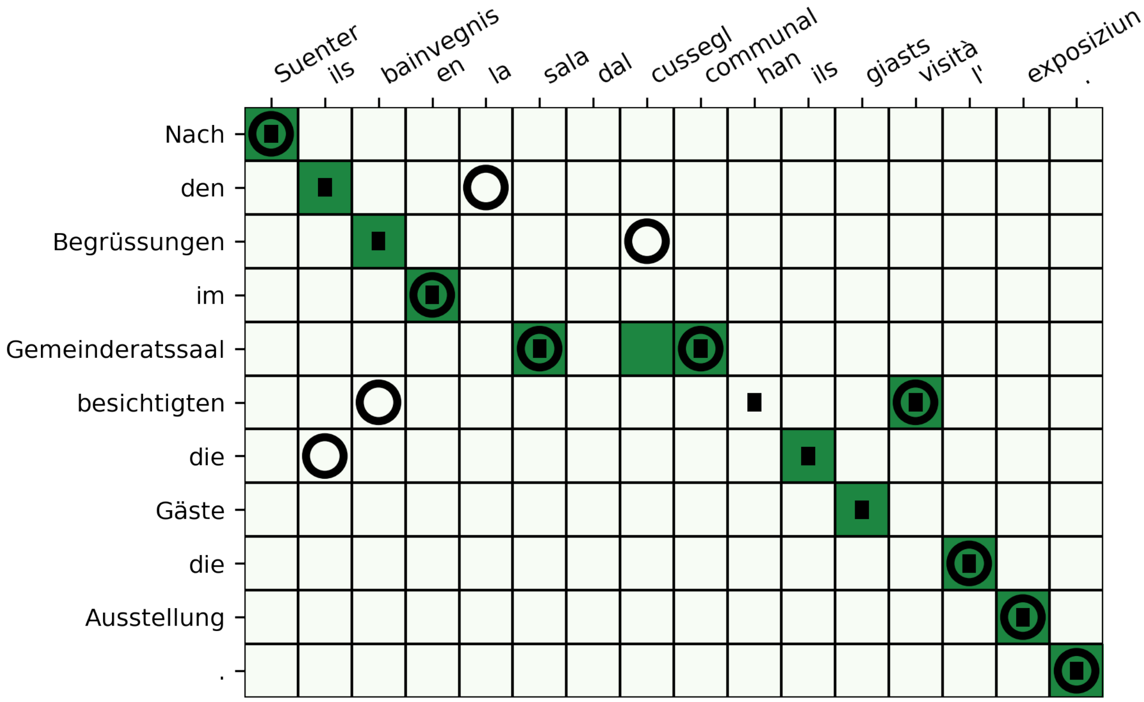

Figure B.5: Word alignment example for the case of German preterite

avert completely unaligned; eflomal aligns *avert* to *Gebäudeversicherung*, which is wrong.

B.4 Double Negation

I picked two random cases with negation, which are expressed by the words *na* ... *betg* in Romansh. In both cases, (Examples B.8 and B.9), eflomal aligns *betg* to the German negation *nicht*, which is correct, but also aligns *na* to the German finite verb, which is wrong. SimAlign fails in both cases to align any of the negating words to each other.

B.5 Differing Word Order

It seems that both models perform well also when word order differs between German and Romansh. In Example B.10, SimAlign has a recalls and precision of 100%, but eflomal is not far behind, missing only one alignment, namely the past participle (see also Section B.2).

In Example B.11, both models deal well with the differing word order, although eflomal's recall is higher. Here, eflomal aligns German *möglichst* ("as much as possible") to Romansh *tant sco pussaivel*, *correctly* creating a 1-to-many alignment. elfomal's precision is punished here due to my gold standard not having Possible alignments for this case of 1-to-many alignment (see also Section 7.4.1).

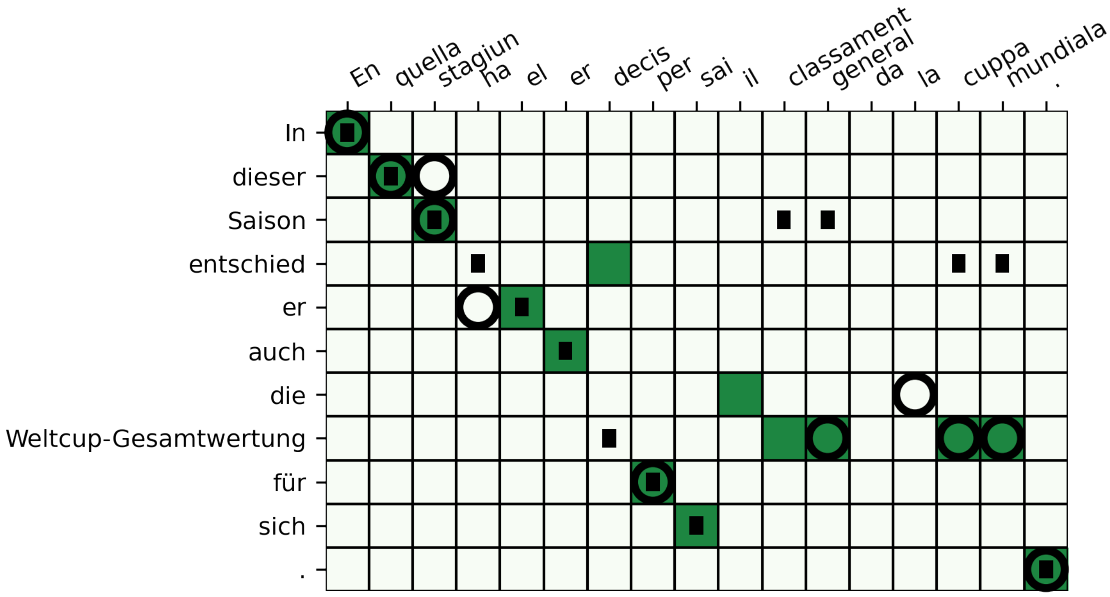

Figure B.6: Word alignment example for the case of German preterite

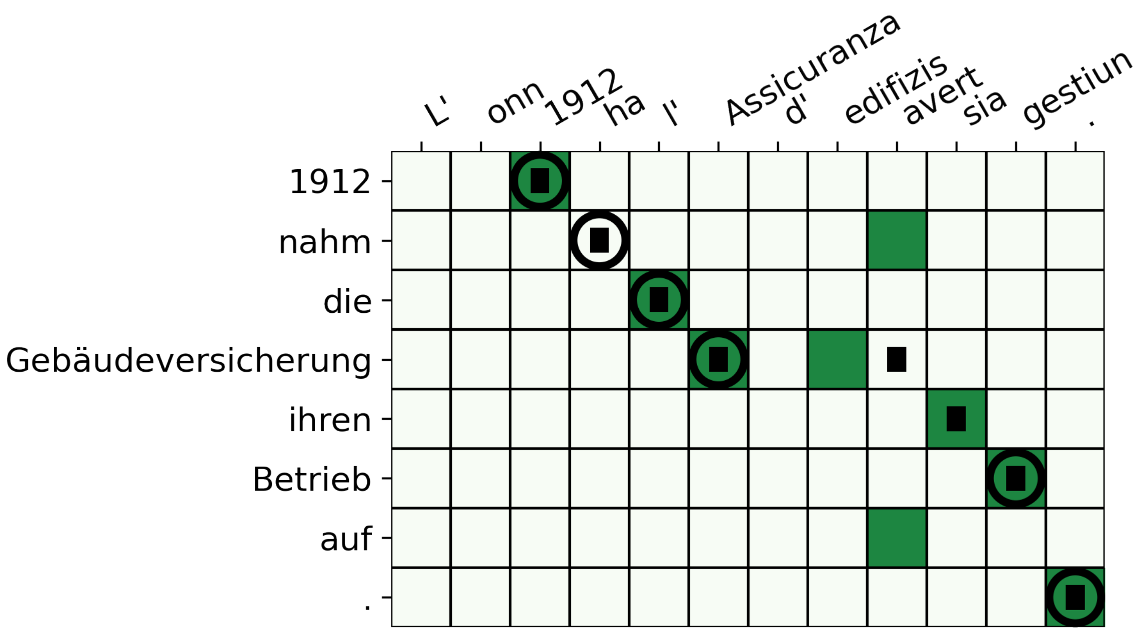

Figure B.7: Word alignment example with a German separable verb in preterite

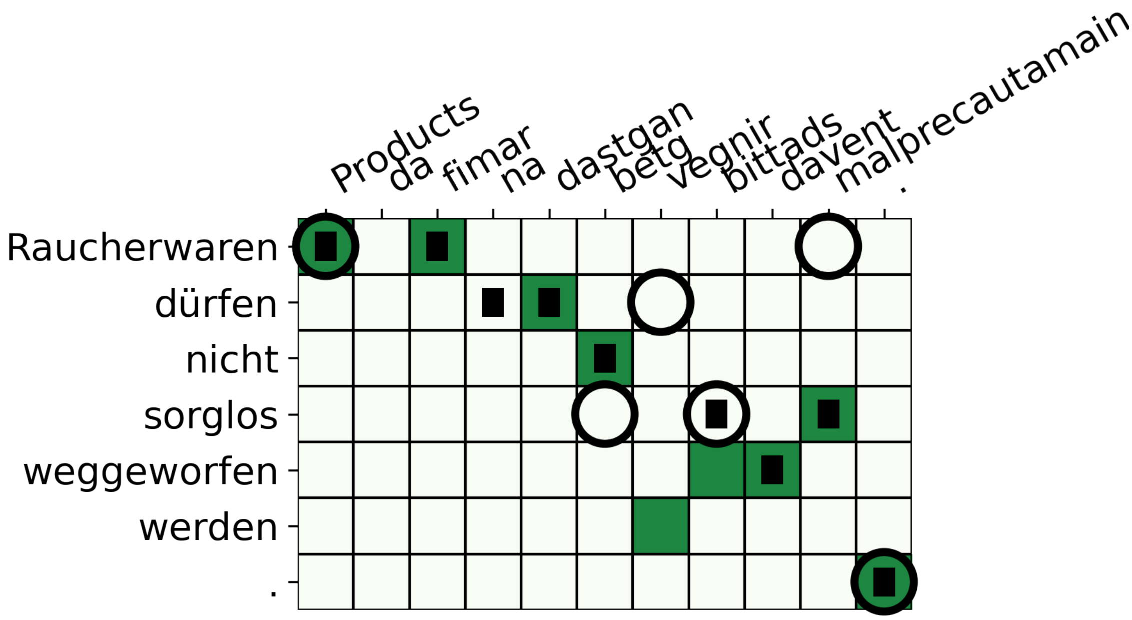

Figure B.8: Word alignment example with Romansh double negation (*na* ... *betg*)

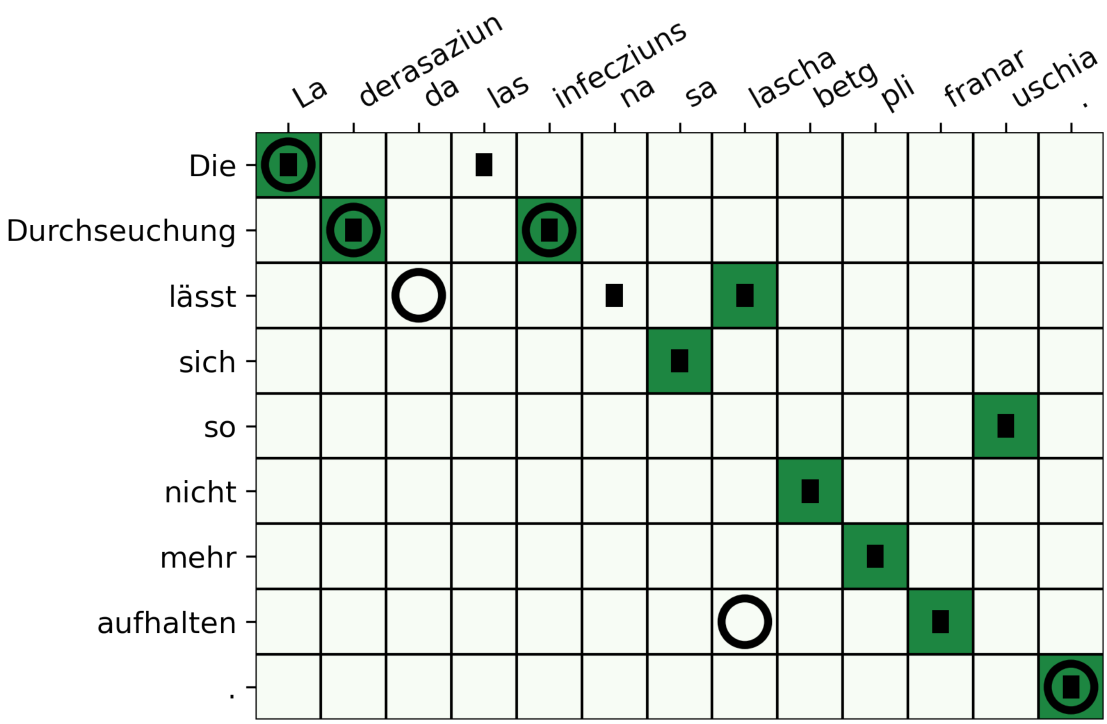

Figure B.9: Word alignment example with Romansh double negation (*na* ... *betg*)

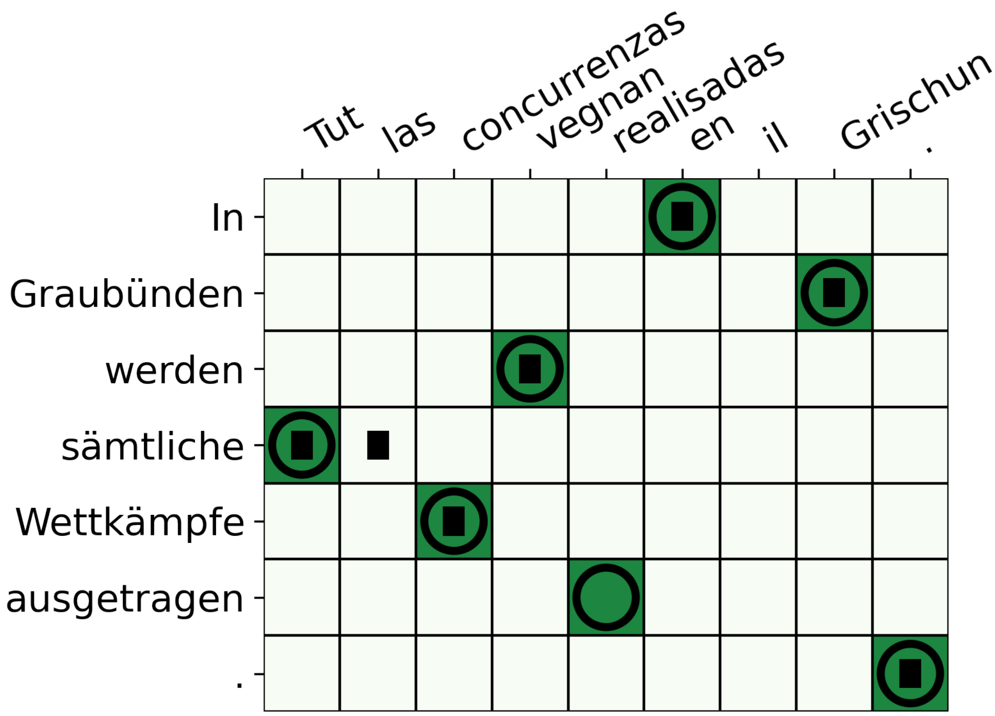

Figure B.10: Word alignment example with differing word order

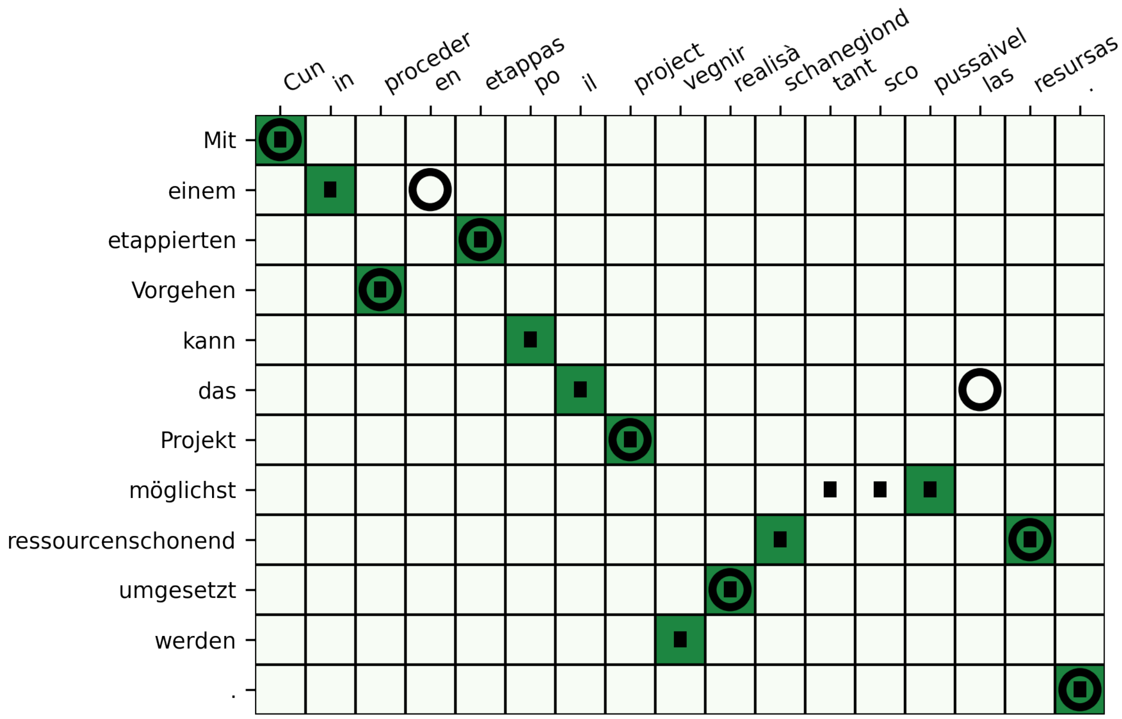

Figure B.11: Word alignment example for a long sentence with differing word order

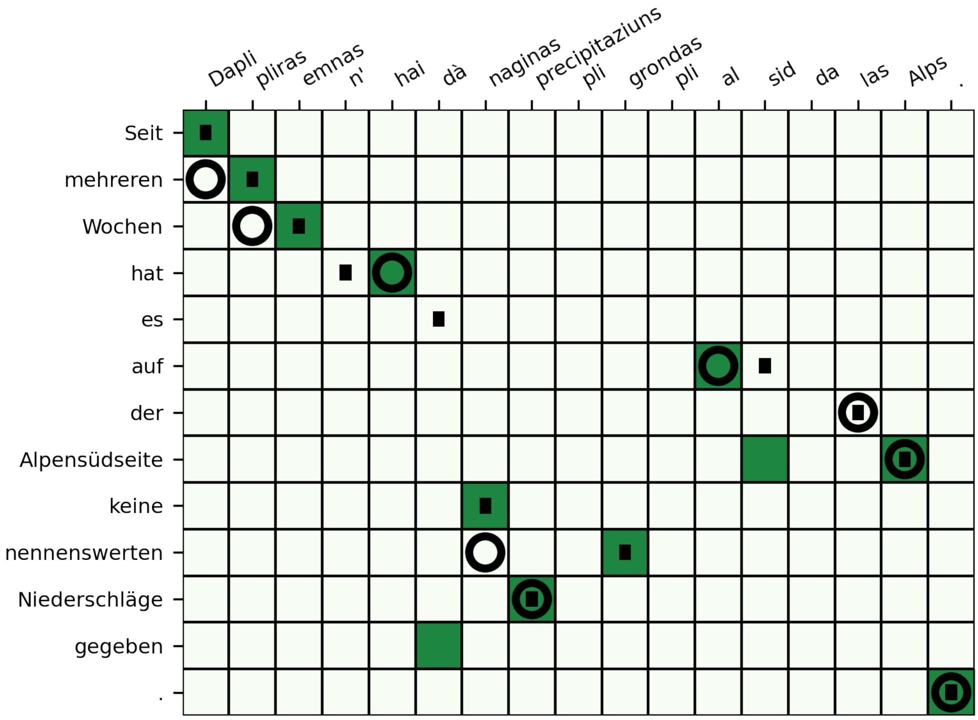

Figure B.12: Word alignment example for a long sentence with differing word order

B.6 Summary

I reviewed the differences between eflomal and SimAlign in some specific cases. It generally seems that both models perform quite well when German and Romansh follow the same word order and when the sentences mostly contain 1-to-1 alignments. German compounds seem to be aligned better by eflomal than by SimAlign. Differing word order is more challenging, but is manageable by both models. However, the combination of 1-to-many alignments and differing word order seems to be quite challenging for both models.

Appendix C

Aligning Romansh to Italian

Due to the nature of my research question, I virtually ignored in the course of this work the issue of word alignments using embeddings (i.e., SimAlign) between Romansh and Italian. Therefore, I would like to curtly attend this issue in this appendix part.

Romansh and Italian share many similarities. Both of them are Romance languages and some researchers even consider Romansh to be a part of the Italian dialect continuum (see Section 2.1).

Since 1-to-many alignments and differing word order are more challenging to model than 1-to-1 alignments and similar or identical word order—word order or 1-to-many alignments are not modeled by IBM Model 1, but only by higher models (Brown et al. 1993)—one might expect that it should be easier to word-align languages that are more similar in structure, word order and grammar. That is, word-aligning Romansh to Italian should be easier than aligning Romansh to German due to the higher similarity between the former languages. Further, when dealing with unseen languages, as in the case of Romansh, multilingual language models have been shown to favor language similarity and vocabulary overlaps (Pires, Schlinger, and Garrette 2019). All this gives rise to the assumption that word alignment for Romansh–Italian might perform better.

I randomly hand-picked a few examples[1] and compared SimAlign's performance on the pairs Romansh-Italian and Romansh-German in order to unempircally[2] test this notion.

The plots in this part were generated using SimAlign's demo website[3].

[1]The only precondition was that the sentences be short; Visualization for longer sentences leaves something to be desired.

[2]Obviously, a gold standard for Romansh-Italian would be needed.

[3]https://simalign.cis.lmu.de

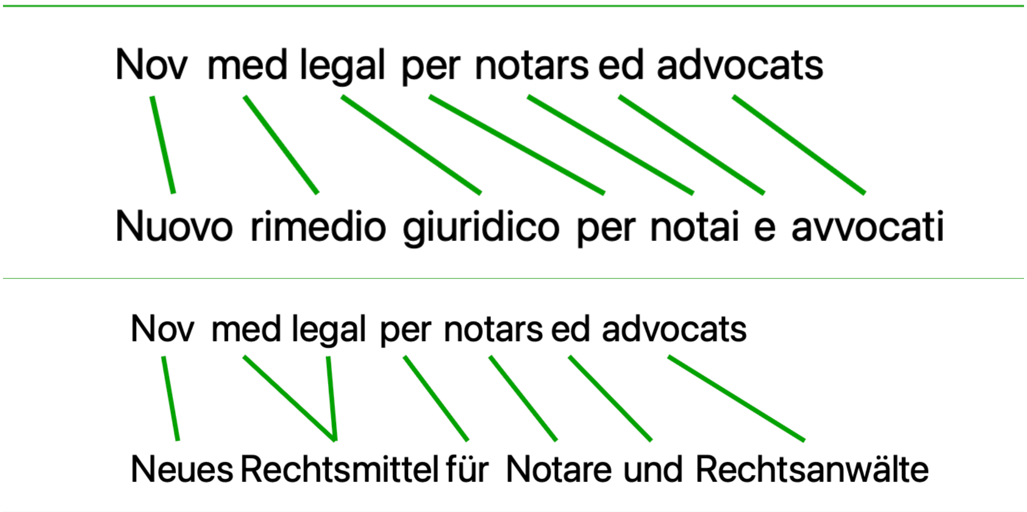

Figure C.1: Word alignment example Romansh–Italian and Romansh–German

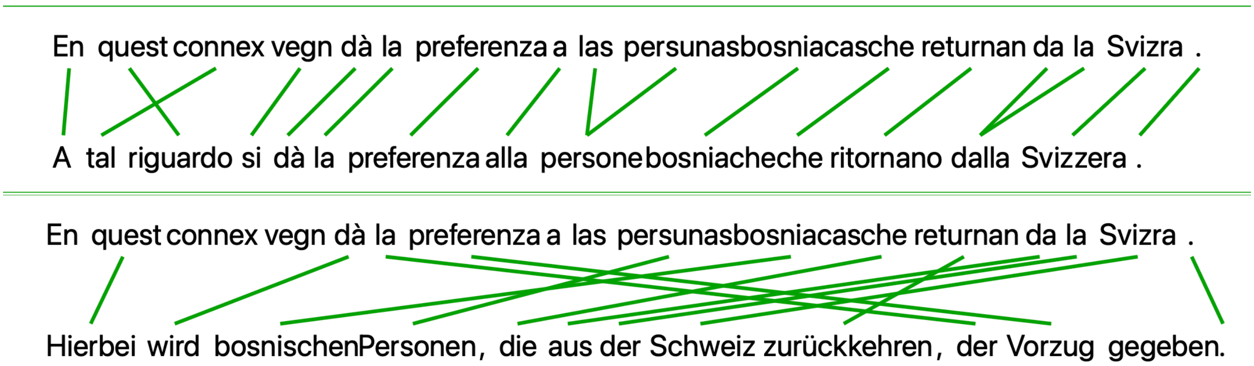

Figure C.2: Word alignment example Romansh–Italian and Romansh–German

C.1 Examples

Figure C.1 is an example for a word alignment that works perfectly both with Italian and with German. In Figure C.2[4], word alignment works well with Italian and German exactly for the same Romansh words, and it is exactly the same words where SimAlign fails: Romansh *en quest connex* ("in this context/matter") is not aligned correctly, neither in German nor in Italian. The same applies for Romansh *vegn* (literally "come", but here part of the passive construction), which is misaligned both times. This is also the case in Figure C.3. The same words are aligned correctly with German and with Italian, but in both cases Romansh *chantun* ("canton") remains unaligned.

In Figure C.4 word alignment with German is even better than with Italian. Here, every alignment is correct, whereas in the Italian example, Romansh *schilar* ("tackle") is not aligned to Italian *affronatare*, which should have been the case.

Finally, Figure C.5 is an example for many misalignments. In the German example, SimAlign succeeds in aligning Romansh *la derasaziuna da infecziuns* to German *die*

[4] Apologies for the somewhat unreadable edges in Romansh–German

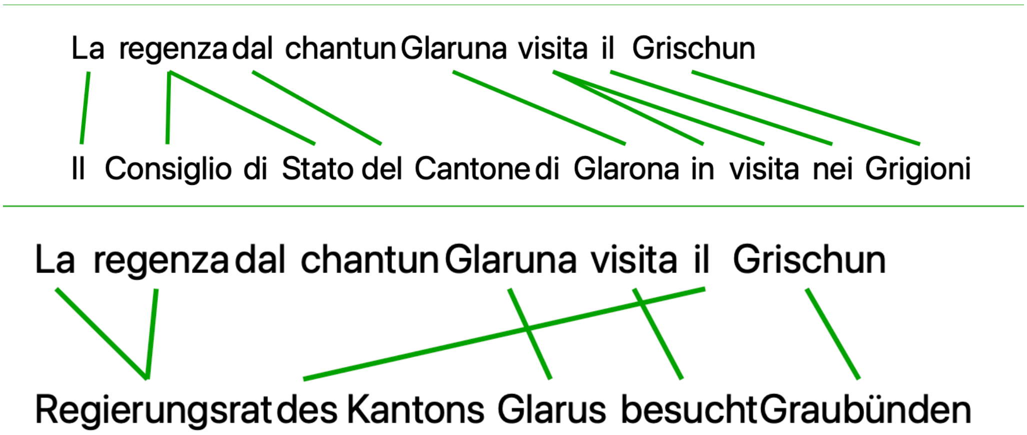

Figure C.3: Word alignment example Romansh–Italian and Romansh–German

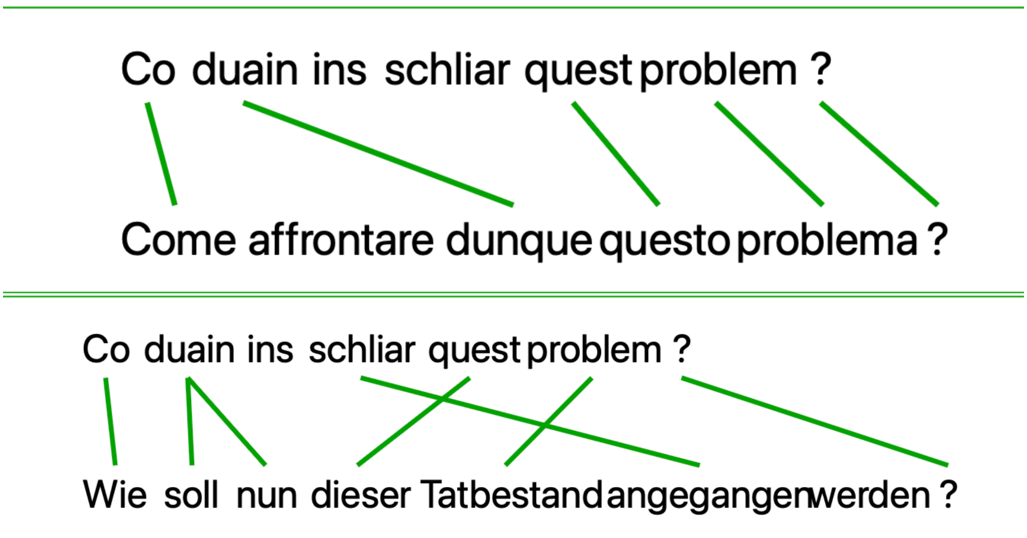

Figure C.4: Word alignment example Romansh–Italian and Romansh–German

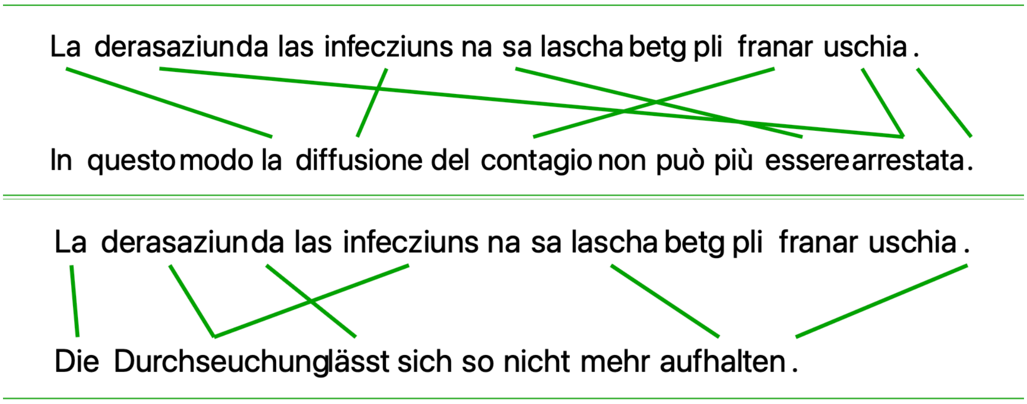

Figure C.5: Word alignment example Romansh–Italian and Romansh–German

Durchseuchung, but the rest of the alignments are wrong. The Italian example is completely misaligned.

C.2 Summary

From observing these very few hand-picked cases, SimAlign doesn't seem to perform better when aligning Romansh to Italian. This is in spite of the higher similarity between Romansh and Italian, compared with German.

One possible explanation for this is that what mostly influences performance is the quality of the embeddings. If the Romansh word is similar enough to any of the words (or subwords) in the language model, alignment will work, regardless of the target language. Take for example Figure C.1. Here, all of the Romansh words are reminiscent of other seen languages and alignment works perfectly. However, in the case of Figure C.3, a suitable embedding for the Romansh word *chantun* apparently cannot be looked-up, hence the word remains unaligned in both cases.