Universität Zürich[UZH]

Masterarbeit
zur Erlangung des akademischen Grades
Master of Arts
der Philosophischen Fakultät der Universität Zürich

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

Tested on the Case of German–Romansh

Verfasser: Eyal Dolev
Matrikel-Nr: (xx-xxx-xxx)

Referent: Prof. Dr. Martin Volk
Institut für Computerlinguistik

Abgabedatum: 01.01.2023

Abstract

This is the place to put the English version of the abstract.

Acknowledgements

I would like to thank...

Contents

Chapter 1

Introduction

1.1 Motivation

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

> 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*) 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 1997, the majority of all press releases published by the Canton Graubünden were released in these three langauges. This existence of such parallel documents in three langauges lends itself to the collection and the compliation of a trilingual parallel corpus.

[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`

Of special interest is here the Romansh language, which, having such a low number of speakers, may be considered a "low resource language".

1.2 Research Question and Goals

1.2.1 Research Questions

Jalili Sabet et al. 2020 were able to show that their algorithm for word alignment, which is similarity based and uses word embeddings to compute similarity, outperforms all the statistical baseline models.

But not only that the model outperforms the existing stastical models, its biggest advantage as propogated by Jalili Sabet et al. 2020 is that it requires no training data. Statistical models will only reach a threshold of good performance with enough training data (Jalili Sabet et al. 2020; Och and Ney 2000). With word embeddings, words in just one single sentence can be aligned with high precision, without the need of a large set of sentence pairs for training a word alignment model. However, all of this works persuming we already have trained a multingual language model, whose learned embeddings we can leverage for this task. There exist some language models that were trained on multi-lingual 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).

Multilingual language models were shown to perform also well on unseen languages, dubbed as "zero-shot setting". Although the LASER model was pretrained on 93 languges, it obtained strong results for sentence embeddings in 112 languages (Artetxe and Schwenk 2019). It was also shown that mBERT performs well 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, then, 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. Especially since vocabulary 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 also has a large portion of Romance-based vocabulary.

1.2.2 Goals

My goals for this thesis are twofold:

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

- Test whether similarity based word alignment using multilingual word embeddings will perform on par with statistical word alignment models on the uneen language 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 ??), then describe how I collected the data and aligned the documents (Chapter ??) and how I further aligned the sentences to create sentence pairs (Chapter ??). In Chapter ?? I will shortly explain the mechanism behind word alignment methods. Finally, I will explain how and according to which guidelines I created the gold standard (Chapter ??) and show the results my experiments comparing different word aligning systems (Chapter ??).

Throughout this work, I went to effort to not become too technical in details, always writing to an imaginary fellow Linguistics student, such that this work, if it ever falls in the hands a future student, will be comprhensible and readable. I hope that it will be read by and inspire future students, the way I was inspired by master's theses written 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 builds a third of the resulting corpus, but is 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 chracterizing phenomena in a number of Romance dialects spoken in parts of Switzerland and Italian (but without a geographical continuum) and named this group of dialects "Ladino". Since 1883, influenced by Theodor Gartner's publication *Raetoromanishce Grammatik* on this group of dialects, this name (German *Rätoromanisch*, English "Rhaeto-Romance") became associated with them.

Rhaeto-Romance is spoken in three separated areas 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 sociolinguistical-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 where language researchers were fascinated by the regularity of sound changes and common historical sound changes were used as the main means to group languages and dialects together. He 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, but were seperated by the Germanic incursions in the years CE 250-800 (Bossong 1998, p. 174; Haiman and Benincà

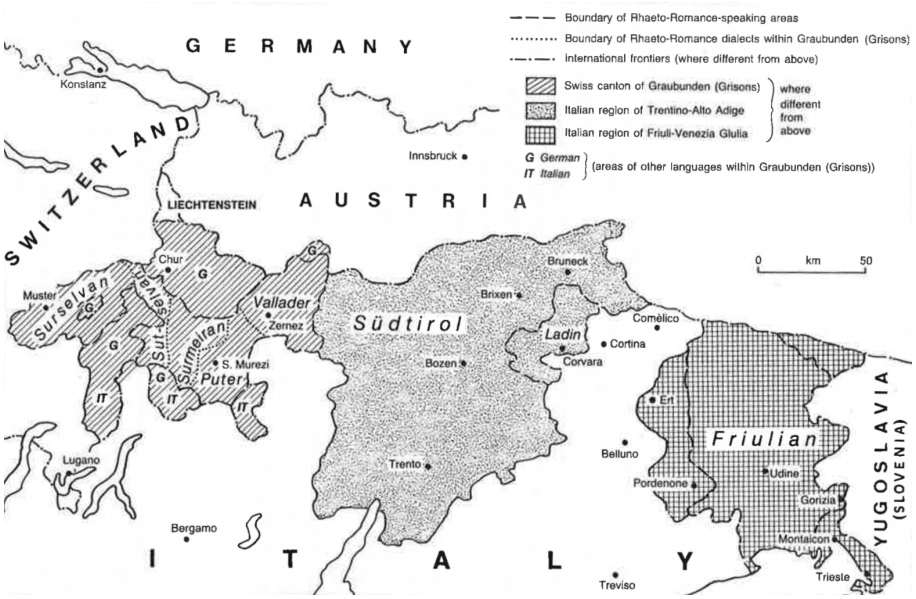


Figure 2.1: Distribution of Rhaeto-Romance, taken from Haiman and Benincà 1992, p. 2

1992, p. 11).

An opposing group of reserchers believes that the three Rhaeto-Romance dialects show decisive features common with their respective neighboring Italian dialects. They should therfore be classified as Italian and 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 Roamansh 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, Sutselvam, Suermeran, Puter, Vallader), with 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 pretensions to "Italian" territories in Switzerland (referred to as the Italian irredentism) (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 just 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 organisation 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, being responsible for developing a language standard and 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 endeavours 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 interrgional and intelligible by speakers of all dialects. The main motivation for planning an interregional 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 Romansch 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 Properties

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 of 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 dialect the word for "key" is *clav*, hence, this is also the Rumantsch Grischun word for "key". In case the dialects do not agree, it uses the word form common to the majority of dialects in a sort of "majority vote". That way, it never prefers one dialect over the other throughout.

Clarity and transperence also play a major role. This means that forms which exhibit stem alternations, for instance between singular and plural, are abandoned in favor of the simpler, more regular form. Further, phenomenons that are specific to just one dialect are left outside, 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 writings and 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 1997, which build up this corpus, in three languages: German, Italian and Romansh using the Rumantsch Grischun standard.

Chapter 3

Compiling the Corpus

3.1 Introduction

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

3.2 Collecting the Data

At first, I contacted the *Standeskanzlei* ("State Chancellery of Grisons") which is the "the general administrative authority for questions of office, coordination and liaison with the cantonal parliament ('Grosser Rat'), government and cantonal administration" (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, we are not talking about private or commercial data, but about public translation work financed with taxpayers' money.

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

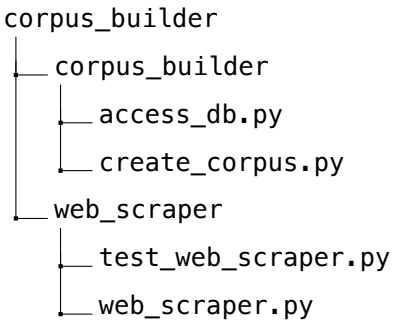


Figure 3.1: Directory tree of `corpus_builder`

3.3 Web Scraping

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

Web Scraper

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

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

By deafult, the script will download the press releases for the entire year range (1997 to the current year) and in all three languages. This can be limited by using the following optional arguments:

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

```
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 3.2: Directory scheme for saving the HTML files

```
--lang de,it
```

3.4 Building the Corpus

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

3.4.1 HTML Parsing

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

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

```
1 {
2     "0": {
3         "id": "12924",
4         "orig_file": "html/1997/1997_12924_DE.html",
5         "lang": "DE",
6         "title": "25 Jahre Arge Alp: Graubünden feiert tüchtig mit",
7         "date": "31.12.1997",
8         "content": "Die Arge Alp feiert heuer ihr 25-Jahre-Jubiläum. Aus diesem
              ↪ Grund finden vom 27. September bis 12. Oktober 1997 die
```

```
            ↪ Festwochen des Alpenraums in Telfs-Mösern, Tirol, statt. ..."
    },
    "1": {
        "id": "12926",
        "orig_file": "html/1997/1997_12926_DE.html",
        "lang": "DE",
        "title": "Kanton will auch personelles Engagement bei der Bündner
            ↪ Kraftwerke AG verstärken",
        "date": "31.12.1997",
        "content": "Nachdem der Kanton Graubünden letzten Herbst die
            ↪ Aktienmehrheit der Bündner Kraftwerke AG übernommen hat, will er
            ↪ nun auch seine Vertretung im Verwaltungsrat stärken...."
    },
    "2": {
        "id": "12927",
        "orig_file": "html/1997/1997_12927_DE.html",
        "lang": "DE",
        "title": "Graubünden trifft präventive Massnahmen zur Bekämpfung der
            ↪ illegalen Einwanderung",
        "date": "31.12.1997",
        "content": "Die Fremdenpolizei des Kantons Graubünden trifft im
            ↪ Einvernehmen mit dem kantonalen Sozialamt, dem Amt für
            ↪ Zivilschutz sowie der Kantonspolizei Graubünden Massnahmen, um
            ↪ die illegale Einwanderung in den Südtälern des Kantons Graubünden
            ↪ zu bekämpfen...."
    },
```

Listing 3.1: Example for a JSON file

3.4.2 Document Alignment

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

Linked vs. unlinked

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

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

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

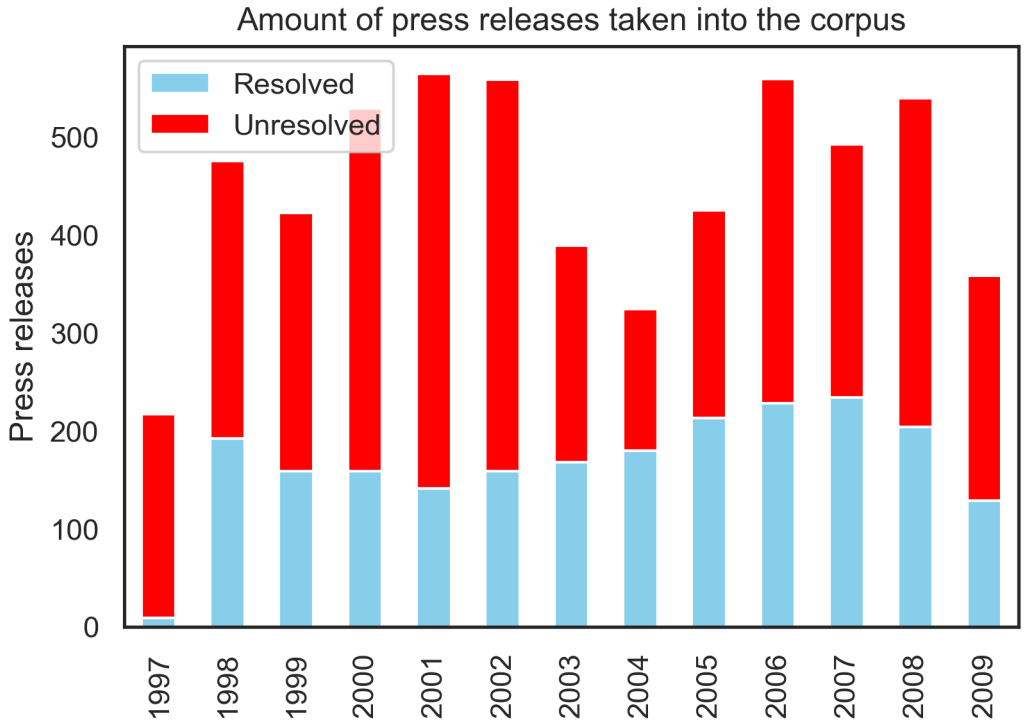


Figure 3.3: Portion of automatically aligned press releases up to 2009

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

Aligned corpus

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

The script `create_corpus.py` deals with this task. Using the Python library Pandas, the JSON files are read into a Dataframe. For linked releases, all the unique ID's are taken,

and then for each ID the three languages are collected and saved into a new row. The dates are converted from their original format (DD.MM.YY) to an ISO 8601 format (YYYY-MM-DD) (Wikipedia contributors 2022) for better compatibility and easier processing later.

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

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

```
{
    "0": {
        "id": "2010010501",
        "date": "2010-01-05",
        "DE_title": "Neues Online-Angebot für das Bündner Rechtsbuch",
        "DE_content": " Das im Internet verfügbare Bündner Rechtsbuch ist neu
            ↪ gestaltet worden und enthält neue Funktionalitäten. ...",
        "IT_title": "Nuova offerta online per la Collezione sistematica del
            ↪ diritto cantonale grigionese",
        "IT_content": " La Collezione sistematica del diritto cantonale
            ↪ grigionese disponibile in internet è stata ristrutturata e
            ↪ contiene nuove funzioni. ... ",
        "RM_title": "Nova purschida d'internet per il cudesch da dretg grischun",
        "RM_content": " Il cudesch da dretg grischun che stat a disposiziun en
            ↪ l'internet ha survegnì in nov concept e novas funcziuns. ... "
    },
    "1": {
        "id": "2010010502",
        "date": "2010-01-05",
        "DE_title": "Staupe bei Füchsen und Dachsen im Puschlav",
        "DE_content": " Nachdem sich im Verlaufe des letzten Herbstes die
            ↪ Staupe-Krankheit bei Wildtieren in Nord- und Mittelbünden
            ↪ verbreitete, sind im Laufe der letzten Wochen nun auch im
            ↪ Puschlav bei Füchsen und Dachsen Infektionen mit dem Staupevirus
            ↪ nachgewiesen worden. ... ",
        "IT_title": "Volpi e tassi affetti da cimurro in Valposchiavo",
        "IT_content": " Dopo che nel corso dell'autunno il cimurro si è diffuso
            ↪ tra gli animali selvatici del Grigioni settentrionale e centrale,
            ↪ nelle ultime settimane la presenza del virus è stata rilevata
            ↪ anche tra volpi e tassi della Valposchiavo. ...  ",
        "RM_title": "Pesta dals chauns tar vulps e tar tass en il Puschlav",
```

```
        "RM_content": " Suenter che la pesta da chauns è sa derasada tar la
            ↪ selvaschina dal Grischun dal nord e central en il decurs da
            ↪ l'atun passà, èn vegnidas cumprovadas en il decurs da las ultimas
            ↪ emnas ussa er infecziuns cun il virus da questa malsogna tar
            ↪ vulps e tar tass en il Puschlav. ... "
    },
    "2": {
        "id": "2010010801",
        "date": "2010-01-08",
        "DE_title": "Projekt Sicherheitsfunknetz POLYCOM Graubünden mit
            ↪ Vertragsunterzeichnung offiziell gestartet",
        "DE_content": " Die Vorsteherin des Departements für Justiz, Sicherheit
            ↪ und Gesundheit, Regierungsrätin Barbara Janom Steiner, und der
            ↪ Chef des Grenzwachtkorps, Jürg Noth, haben heute in Chur eine
            ↪ Vereinbarung zur Realisierung des Sicherheitsfunknetzes POLYCOM
            ↪ im Kanton unterzeichnet. ... ",
        "IT_title": "Avviato ufficialmente con la sottoscrizione del contratto
            ↪ il progetto di rete radio di sicurezza POLYCOM Grigioni",
        "IT_content": " La Consigliera di Stato Barbara Janom Steiner,
            ↪ direttrice del Dipartimento di giustizia, sicurezza e sanità, e
            ↪ il capo del Corpo delle guardie di confine, Jürg Noth, hanno
            ↪ sottoscritto oggi a Coira un accordo per la realizzazione nel
            ↪ Cantone della rete radio di sicurezza POLYCOM. ... ",
        "RM_title": "Il project per la rait radiofonica da segirezza POLYCOM dal
            ↪ Grischun è vegnì lantschà uffizialmain cun suttascriver il
            ↪ contract",
        "RM_content": " La scheffa dal departament da giustia, segirezza e
            ↪ sanadad, cussegliera guvernativa Barbara Janom Steiner, ed il
            ↪ schef dal corp da guardias da cunfin, Jürg Noth, han suttascrit
            ↪ oz a Cuira ina cunvegna per realisar la rait radiofonica da
            ↪ segirezza POLYCOM en il chantun. ... "
    },
}
```

Listing 3.2: Example for a JSON file containing aligned documents

3.5 Manual alignment of unlinked documents

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

to a CSV file, one for each year.

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

3.6 SQLite database

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

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

3.7 Summary

For compiling the corpus, the following steps were taken:

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

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

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

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

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

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

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

  - DE_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

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

3.7.1 Statistics

The corpus contains 3,536 parallel documents.

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

German		Romansh		Italian	
Type	Count	Type	Count	Type	Count
.	122176	da	199663	di	110946
,	88048	la	111803	.	93732
der	82288	.	98181	,	80509
und	74298	,	59456	e	62295
die	69741	e	56493	la	40558
in	33423	il	50155	il	38844
für	31464	per	46475	per	38100
des	27467	las	45869	del	36805
den	27037	en	42061	della	31309
:	26154	dal	40407	a	31164
Die	25310	a	30284	in	30506
von	25205	ils	22554	dei	25525
im	20684	:	21442	:	21407
Graubünden	17859	cun	20114	un	20410
zu	17709	che	19931	i	17909
werden	17258	La	18494	è	17525
mit	16621	ina	18413	le	17051
Regierung	15557	è	18053	che	15986
auf	15276	Grischun	16775	Governo	15592
das	14370	ed	16030	Grigioni	15337

Table 3.3: Twenty most common tokens in each language in the corpus.

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. Probably the most important task bilngual corpora are used for nowadays is for training a machine translation model (Gale and Church 1991; Moore 2002; Chen 1993), but other tasks it can be used for are building translation memories (Sennrich and Volk 2011) or a for a bilingual concordance system with the purpose of allowing a user to find out how a given translation is translated (Moore 2002; Gale and Church 1991).

4.1.1 Formal definition

Formally, the task 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 1–1 alignment is an alignment where exactly one sentence of language e is aligned to exactly one sentence of language f. In a 1–2 alignemnt, one sentence in lanauge e is a aligned to two sentences

in langauge f. There are also 0–1 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 the one described in Gale and Church 1991 which is "based on a simple satistical model of character lengths" (Gale and Church 1991). The method arose out of the need to design a faster, computationally more efficient algorithm than the ones that existed at the time[1].

The Gale & Church method uses the fact 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 probility of the alignment type (1–1; 1–0 or 0–1; 2–1 or 1–2; 2–2) to a probabilistic score. It assigns this score to possible sentence pairs in a dynamic programming framework to find the best (most probable) pairs. 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 could 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, around 3.5 times faster than previous algorithms.

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* are used to mark sort of *anchors* in the texts. The term *cognate* refers here to two word-forms of different languages whose first four characters are identical. Isolated cognates are cognates

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

with no resmebling 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 until no more anchor points can be found.

In an intermediate step, segmentation into sentence boundaries takes place and the search space is determined, i.e., 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), to measure how probable it is to observe one sentence given another sentence and so find the sentences that are most probably 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.

Another hybrid method was presented in 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 supplied to the program. The program 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. The 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.

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[2], 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, misalignments can be detected faster and filtering of texts that are less suitable for alignment (mixed order of chapters, different prefaces, etc.) can be carried out earlier. Pre-processing (tokenization, sentence segmentation) may also influence the alignment quality. Tweaking and fine-tuning these parameters may also require several runs (Varga et al. 2005).

In other words, sentence alignment for a big corpus often requires several passes or runs until misalignments due to less suitable texts or faulty tokenization and sentence segmentation are revealed. An algorithm that performs faster has an obvious advantage here.

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 cited since as Bleualign. It rose as a method addressing to the problem of aligning less "easily" alignable corpora. Sentence alignment methods up to that time perform excellent on well-strctured corpora with a high language similarity such as the Canadian Hansards which are considered easy to align or the Europarl[3] because they are well-structured—they provide markup information to identify speakers which is useful for creating anchor points and the subsequent alignment (Simard and Plamondon 1996; Sennrich and Volk 2011). However, when

[2] transcriptions of parliamentary debates which exist in English and in French

[3] parliamentary proceedings of the EU Parliament

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[4] [5]. 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 probable two sentences are mutual translations. BLEU scores for two unrelated sentences is usually 0 (Sennrich and Volk 2010). 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.

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 an SMT (statistical machine translation) 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.

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 have a crucial effect on neural MT (NMT) systems. This is especially true in scenarios with less data for low-resource MT (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 (cf. sec-

[4] sequences of tokens of length n

[5] usually scores are combined for n-grams of order 1–4

tion TODO), 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 reside in the same vector space. This means, the embeddings are general to the input language; they are language agnostic. If two sentences 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. In this manner, similar sentences in different languages can be identified and aligned (Artetxe and Schwenk 2019).

4.4 Sentence alignment pipeline

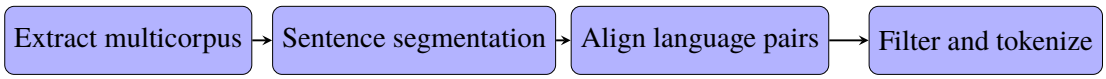


Figure 4.1: Sentence alignment pipeline

I shall now describe the pipeline I used for retrieving sentence pairs for 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 (MacBook Air M1, 2020 running MacOS Monterey 12.3.1) and runs fast (aligning the entire corpus takes around three minutes).

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

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 better than that gained 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[6].

[6] `https://github.com/facebookresearch/LASER`

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 sentneces—`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, I merely had to discard 11 out of 600 sentences due to misalignment. This corresponds to a precision of 98.2%.

4.4.2 Pipeline

In the first step, all aligned documents are extracted from the corpus and are written to monolingual files, one sentenc per line, and one file per year. This is done by querying the SQLite database for all the aligned documents for each year seperately.

4.4.3 Sentence segmentation

Sentence segmentation (also called sentence tokenization) was done using NLTK's Punkt tokenizers. Since I wasn't able to integrate a sentence tokenizer for Romansh into the pipeline, I used the NLTK's Punkt tokenizer model which was trained on Italian. After instantiating both the German and the Italian models, I extended the list of abbrevations[7] to enhance the performance of the tokenizer and avoid wrong segmentation.

In the course of sentence segmentation, paragraphs are retained by converting linebreaks into a special `<p>` token. These tokens will serve `hunalign` as anchor points for sentence alignment.

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 (see section 3.3 Web Scraper) is included at the beginning of each document. In case there is no mutual file ID, the date is included. The file ID/date will be used by `hunalign` as anchor points for keeping the documents and the paragraphs aligned, see listing 4.1.

Listing 4.1: 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 `<t>` token.

```
1 2004-01-27
2 www.gr.ch neu mit Online-Schalter und mit Interessenbindungen des Grossen Rats
3 Ein neues, zentrales Element von www.gr.ch ist der integrierte Behörden-Online-Schalter
      www.ch.ch.
4 ...
```

[7] The abbreviations for Romansh were taken from Samuel Läubli's/Lisa Gasner GitHub repository

```
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
```

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[8].

Although inspection with the bare eye revealed excellent precision (from the 600 sentences extracted for word alignment 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*[9] 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.

[8] `https://github.com/danielvarga/hunalign`

[9] `https://www.pledarigrond.ch/rumantschgrischun`

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 are also considered noise in the data which might negatively influence a machine translation model trained on this corpus. Therefore, the sentences are filtered for duplicates, as well as according to some other heuristics, 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-seperated file containing two columns (source and target) with the filtered corpus, 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 or where the sentence length ratio between source and target is too large, meaning the sentences are unlikely mutual translations.

Word tokenization is important for the next step—word alignment. For the task of tokenization, I used NLTK's 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, I had to correct the tokenization less than 10 times out of 600 sentences.

4.5 Results

The resulting final parallel corpus consists of three files containing around 80,000 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 (cf., listing 4.2). Table 4.1 elaborates on the number of sentences, tokens and type for each combination.

```
1 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 .
2 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 .
```

Combination	Sentences	Tokens Source	Types Source	Tokens Target	Types Target
German–Romansh	79,548	1,399,382	80,239	1,791,511	42,570
German–Italian	78,108	1,396,933	80,239	1,684,152	48,787
Romansh–Italian	78,030	1,758,448	42,235	1,654,165	48,680

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

```
3 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: An excerpt from the file containing sentence pairs in German–Romansh

Chapter 5

Word Alignment

We now reach the core of my thesis, computing word alignments using the novel method SimAlign suggested by Jalili Sabet et al. 2020 and evaluating it against a baseline method. In the following pages, I shall give a short introduction on the topic of word alignment and explain the mechanisms behind statistical word alignment and similarity 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*.

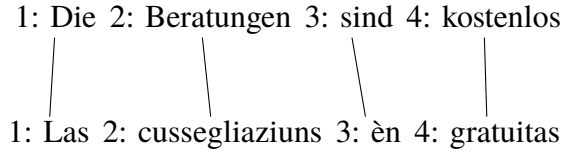


Figure 5.1: Example of a word alignment between two sentences in German and Romansh

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`, an alignment model based on 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 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. For example, 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 of 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. 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 languages. 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 algorithm (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 unkown and are initialized with a uniform distribution. This means all words are equally likely to be 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 estimation step. These two steps, estimation and maximization, are then repeated until convergence—until a global minimum is reached and the probabilities computed stop changing (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 details, 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, 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 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 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). Further, 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) (my examples).

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 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 one, 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 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.*[2] [3]

5.3.2 Word Embeddings

These ideas, which were further developed by linguists in the 1950's, 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 inspriation 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).

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 spaker 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).

Without going too much into details, a neural network is a complex non-linear function. It is made 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 means of matrix multiplication with the weights between those layers. The vector resulting from this matrix multiplication (plus usually some non-linear activation function), is the next layer in the neural net. The output of a neural network can be for instance 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.

[2] 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.

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

In the course of the training of a language neural model, i.e., while the neural net 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 net (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).

There are different ways of 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.

5.3.3 Word Similarity

If words are represented by vectors, we need a measure for taking two such vectors and measuring 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, that is, vectors with high values in each dimension, which represents the frequency of words. This means more frequent words have higher values, but we want to measure 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 $\sqrt{\sum_{i=1}^{N} v_i^2}$ being the length of the vector $\mathbf{v}$.

(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 vec-

tor 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). They 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 which share the same vector space. Similar words of different languages should have similar embeddings.

5.4 Similarity Based Word Alignment

With multilingual word embeddings, similar words in different languages should have similar embeddings. This similarity between embeddings in different languages can be leveraged in order to find word alignments using a similarity matrix, without the need for parallel data, which 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 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).

		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 numbers 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.

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

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

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 remedey 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. 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 alignment.

Argmax finds a local optimum and Itermax is a greedy algorithm. There is a third alignemnt 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

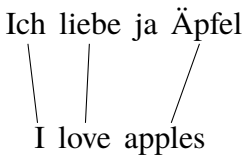


Figure 5.4: The resulting word alignment

the target is aligned to the source).

For the task for word alignment, SimAlign can use multilingual embeddings which were learned in advanced from monolingual data and then mapped to a shared vector spaced. 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[4], and XLM-RoBERTa base, trained on 100 languages (Conneau et al. 2020).

5.4.2 Summary

By measuring the similarity between multilingual word embeddings, word alignments for sentence pairs can be computed.

Multilingual embeddings can be learned from monolingual data, and thus word alignment 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 data to perform well. The quality of the alignments deteriorates quickly when the size of data diminishes[5].

.

Similarity based word alignment using embeddings extracted from mBERT or XLM-R outperforms any state-of-the-art statistical method for the language pairs Czech, German, French and Hindi, paired with English. But all of these languages are part of mBERT's and XLM-R's training data. Jalili Sabet et al. 2020 emphasize the advantage of their method performing well also on a small amount of parallel data. But will SimAlign perform just as well for data unseen by said language models?

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

[5] In Och and Ney 2000, the Average Error Rate (AER) for aligning 1.5M sentence pairs is 9.4%. When aligning 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 alignment based on measuring similarity between multilingual word embeddings. The clear advantage of this method is that it does not rely on the existence of large amounts 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). In the course of this work, I was able to extract 79,109 sentence pairs for German-Romansh. I shall now proceed to test how well SimAlign performs on this language pair, considering the fact the Romansh wasn't included in the training data for these models.

In order to measure the quality of words alignments, a model's performance is measured on a test set which is a gold standard 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 (TODO) that have no clear equivalent in the other language. Koehn 2009 gives as an example the German-English sentence pair: *John wohnt hier nicht 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 to clear equivalent in German.

2. The word *does* is connected with *live*; it contains the number and tense information which is in German contained in one word *wohnt*, so it should be aligned to *wohnt*, together with *live*.

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).

6.2 Sure and Possible Alignments

An approach for solving problematic cases is the distinction between *sure* (s) and *possible* (p) 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 (more on that later).

There seems to be no clear global definition about which alignments should be considered as umabiguous and marked as *sure* and which should be considered ambiguous marked as *possible*. For some created gold standards, no distinciton between *sure* and *possible* alignments was made (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). In the creation of the English-Icelandic gold standard in Steingrímsson, Loftsson, and Way 2021, annotators used only *sure* links. Their annotations were then combined, with all 1-1 alignments both annotators agreed upon (i.e., the intersection of their annotations) makred as *sure* and differences all other alignments made by either one or both were marked as *possible* (Steingrímsson, Loftsson, and Way 2021).

6.3 Evaluation Metrics

TODO: move to results/evaluation part Four types of measures have become standard for evaluating word alignment. Three of them – precision, recall and F-measure – are well known in Information Retrieval metrics Mihalcea and Pedersen 2003. The fourth, alignment error rate (AER) one was introduced by Och and Ney 2000.

6.4 Gold standard for German-Romansh

In order to measure the performance of both models, the embedding based model (SimAlign) and the stastitical model (fast_align), 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, which also offers a grammar overview. (TODO: add more grammar references)

6.4.1 Annotation tool

I used the tool *AlignMan* which was originally programmed for creating the gold standard for English-Icelandic by 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 above, the tool does not allow labeling of links with *Sure* and *Possible*. Instead, AlignMan treats the union of 1-1 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, which also applied to me while annotating the German-Romansh gold standard.

6.4.2 Guidelines

As mentioned above, 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.

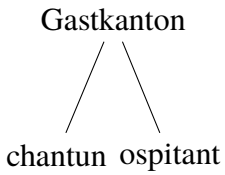
A motto cited often for annotating word alignments is "Align as small segments as possible, and as long segments as necessary" (Vronis and Langlais 2000, cited in Ahrenberg 2007). A variation of this is found in Clematide et al. 2018: "as few words as possible and as many words as necessary that carry the same meaning should be aligned.", referring to Lambert et al. 2005.

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

6.4.3 General priniciples

Principle I. Use only *Sure* alignments. Since the annotating tool I was using does not provide the use of confidence labels (cf. section 6.4.1), I only aligned words which would be considered *Sure* alignments, i.e., they are unambiguous (cf. section 6.2).

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



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 alignments or morphosyntactical alignments). This means alignments should describe first and foremost lexical correspondences, i.e., they have the same lexical meaning (but not necessarily share the same grammatical function or the same part-of-speech). 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 (TODO: example?)

- only sure alignments
- prefer 1-1 alignments over 1-n alignments
- align words, not phrases
- only align words that are translations of each other also outside of context
- POS doesn't matter: German often prefers a nominal style, Romansh prefers a verbal style – expect some noun-verb alignments.

6.4.4 Examples

I will now supply 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). Other prepositions that can be found for linking words are *cunter* and *per*. [1] In other cases, German compounds might be translated to Romansh using an adjective +

[1] 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)[510].

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

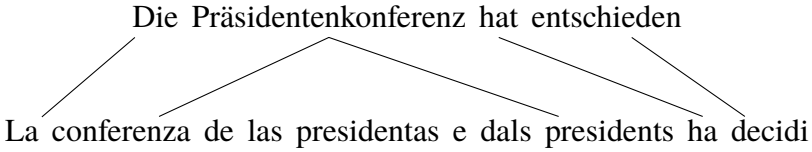


Figure 6.1: Aligning German perfect to Romansh perfect

noun, e.g., German *Gastkanton* was translated to *chantun ospitant* "hosting canton". See table 6.1 for examples.

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

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 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 conference has decided".

In contranst 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 ex-

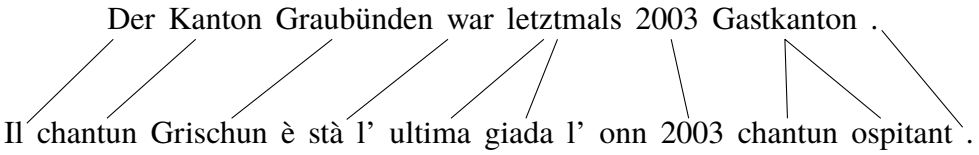


Figure 6.2: Alignment of German preterite to Romansh perfect

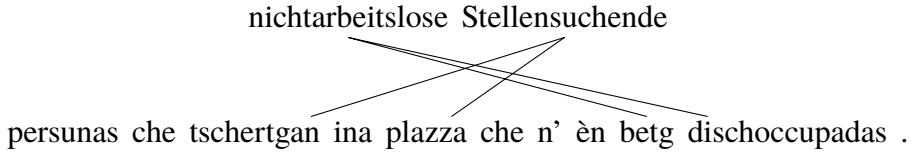


Figure 6.3: Aligning German present participles to Romansh relative clauses

ample, 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 1-2 link. However, since the verb *è* here only carries grammatical information of tense and number, but no real lexical information, it should reamin unaligned.

The German perfect should be aligned to the Romansh perfect using a 1-1 alignment; auxiliary to auxiliary and participle to participle. **The German preterite should also be aligned using a 1-1 alignment to the Romansh participle, leaving the auxiliary unaligned and avoiding a 1-2 alignment.**

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 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"

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*.

Double negation

Negation in Romansh is built using two particles: *na* and *betg* to negate verbs or *nagin-* to negate nouns. Since we prefer 1-1 alignments, 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. Granted, this is also in favor of the SimAlign output, but it is also linguistically motivated: when negating the imperative form, *na* can be omitted required TODO:cite Grammatica per l'instrucziun dal rumantsch grischun.

Articles and prepositions

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

Separable verbs

German uses many verbs to which an adverb or a preopisition is affixed in order to delimit the verb's meaning (or sometimes completely change its meaning). In such cases, both the verb and its affix should be aligned to the corresponding Romansh verb, resulting in a 2-1 alignment.

6.5 Flaws

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

The most obvious flaw is the fact that I created the gold standard alone. With more than one annotator, more intricate 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 *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 annotaors 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 my case.

Another flaw is the missing confidence labels (*Sure* and *Possible*), which may influence the evaluation scores. Doing without *Possible* links and using only *Sure* links is however precedented (Clematide et al. 2018; Mihalcea and Pedersen 2003) and hence defensible.

In order to test my own consistency, I have re-annotated the first 100 sentences in the sample. TODO: results

Despite of the flaws mentioned, I am certain that gold standard is of high quality and consistency, due to the fact that I was also the one to define the 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 a baseline system. I shall now proceed to present the results of the experiment.

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 $= \frac{TP}{TP+FP}$, where TP refers to "true positives" and FP to "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 $= \frac{TP}{TP+FN}$, where TP refers to "true positives" and FN to "false negatives" (Jurafsky and Martin 2019, p. 67).

F-measure is a score that incorporates precision and recall. The fourth measurement, Average 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.

I will later discuss shortly the problems of evaluation (see 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, 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 took around 50 seconds. It is well documented and is extremely easy to compile and to operate. All of this makes `fast_align` a most attractive system to use as a baseline system.

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

Method	Dataset Size	Precision	Recall	F_1	AER
fast_align	79,548	**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	79,548	**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: Evaluation metrics for word alignments with the baseline models for different dataset sizes. "Dataset Size" refers to the number of sentence pairs.

7.2.2 eflomal

eflomal (a.k.a. efmaral[3]) 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

Since statistical word alignment models heavily rely on a minimal amount of data and 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) and then extracted the alignments for the 600 annotated sentences.

The results are shown in Table 7.1.

7.3 SimAlign

I word-aligned the 600 sentences for which I created a 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[4] and XLM-R(Conneau et al. 2020). mBERT only

[3]eflomal is a more memory efficient version of efmaral. Cf. `https://github.com/robertostling/efmaral`

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

	Embedding	Level	Method	Percision	Recall	F_1	AER
SimAlign	mBert	BPE	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**
		BPE	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: Evaluation metrics for word alignments using SimAlign, with different embeddings and word/sub-word level. Best result per embedding type in bold.

works on a subword level (BPE), while XLM-R works either on the word or the subword level.

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 shows the evaluation metrics for word alignments computed with SimAlign with the different 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.

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 these results are promising 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 fast_align and its precision is 27% higher than that of fast_align. Also, in the hypothetical case that we only had the 600 annotated sentences to compute word

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-BPE	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.

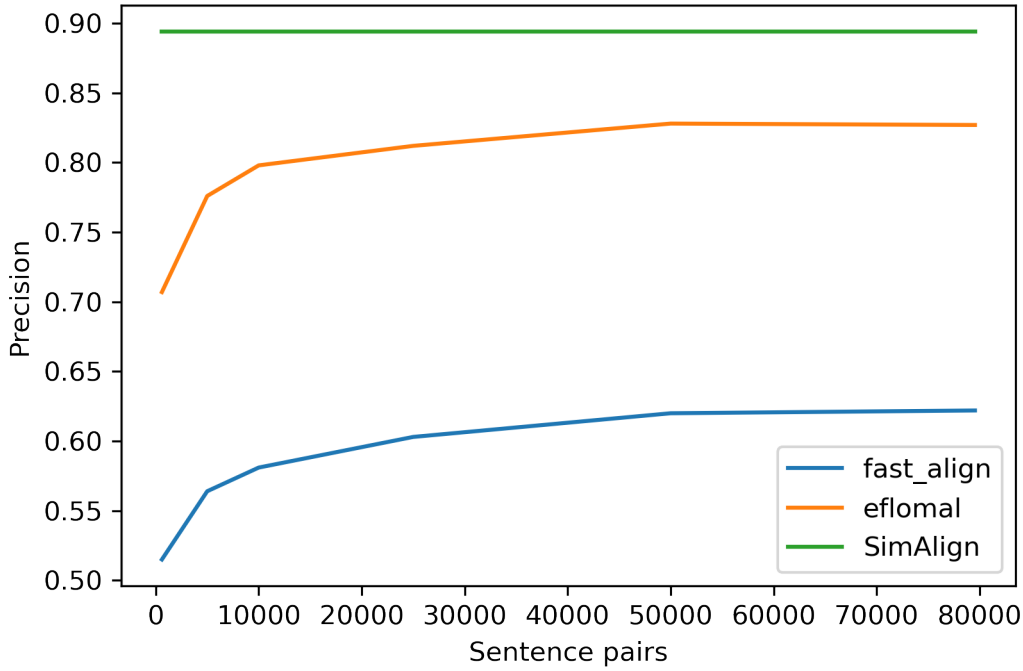


Figure 7.1: Comparing precision between the systems for different dataset sizes.

alignment, SimAlign would have outperformed eflomal as well with an AER of 0.284 (SimAlign) against an AER of 0.219 (eflomal) (cf. Table 7.1).

Further, SimAlign's performance on the language pair German-Romansh (AER of 0.219) doesn't fall from the performance of SimAlign on English-German sentence pairs (AER of 0.21), as presented in Table 2 in Jalili Sabet et al. 2020. This means performance in a zero-shot setting with mBERT embeddings for German-Romansh is 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 preci-

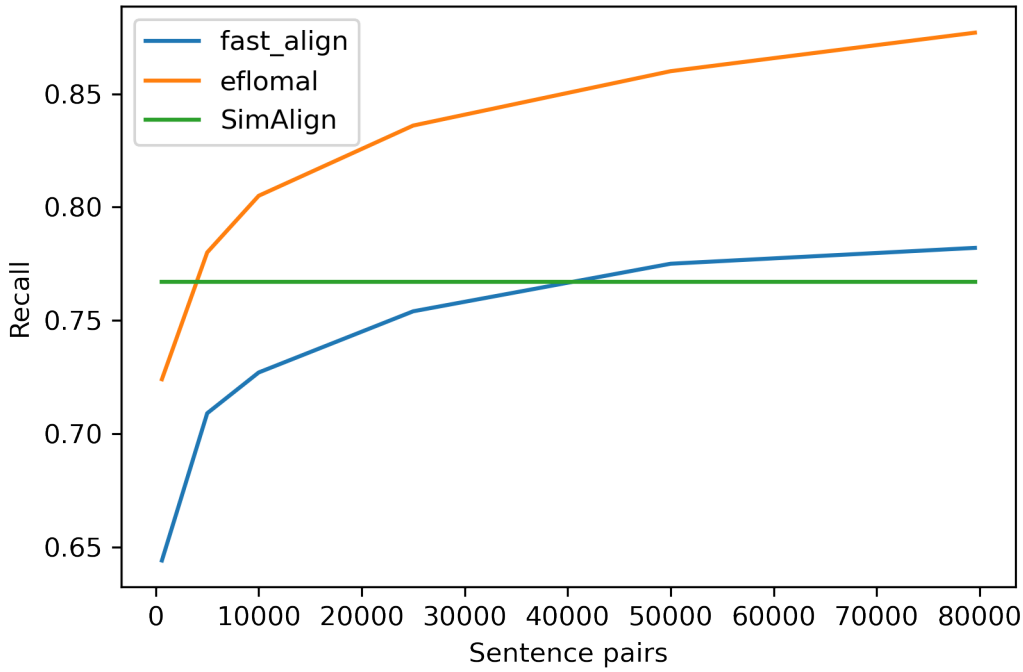


Figure 7.2: Comparing recall between the systems for different dataset sizes.

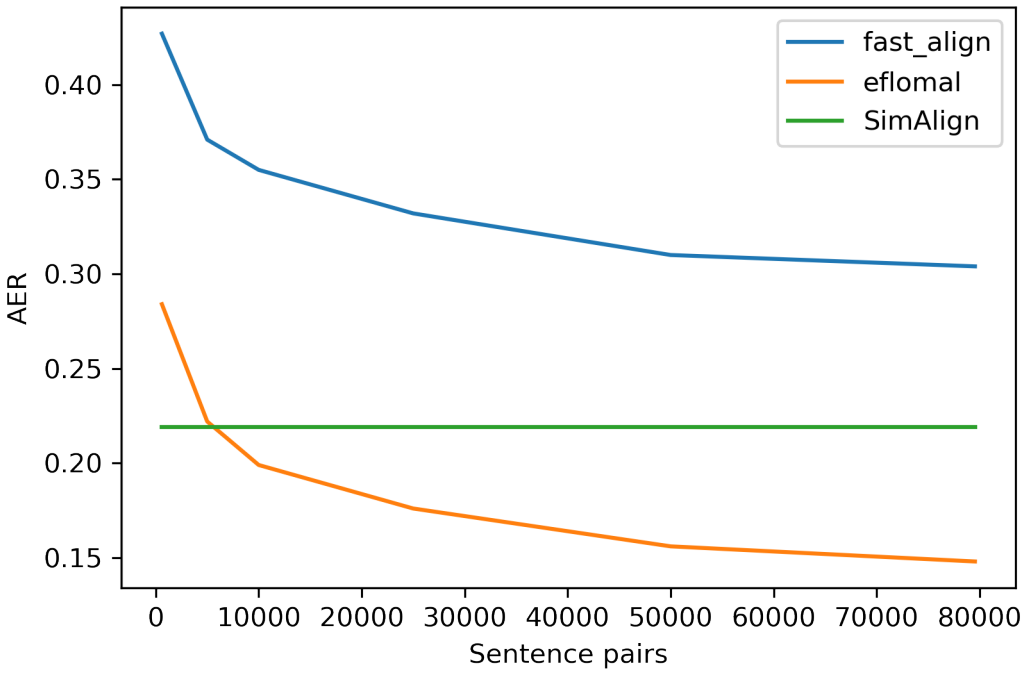


Figure 7.3: Comparing AER between the systems for different dataset sizes.

sion performance of a model that generates 1-to-many alignments, although they 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 recall.

Problems with the Gold Standard for German-Romansh

As already explained in Section 6.5, 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 6.4.3) 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 0.015 points, a difference of 2%.

All that being said, I believe the excellent performance of eflomal proves that the gold standard is of good quality and is sensible for measuring the performance of word alignment models on German-Romansh.

7.5 Summary

I evaluated the performance of the two statistical baseline models (fast_align and eflomal) against the performance of SimAlign, a similarity based word alignment model, using a gold standard of 600 annotated sentence pairs in German-Romansh, which I had created myself. I compared the performance of two baseline statistical models with the performance of SimAlign using multilingual embeddings in a zero-shot setting. 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 etc.

See Appendix A for some alignment examples.

Chapter 8

Concluding Words

8.1 Goals

The goals of this work was twofold:

- Enlarge the amount of resources that is available containing 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 since 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, which allow for fast and simple queries.

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

I am glad to make the corpus the I collected, as well as the aligned sentence pairs and triplets 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 of parts thereof.

8.3 Gold Standard

In order to evaluate word alignment systems, a gold standard is needed. In the context of word alignment, a gold standard is a collection of sentence pairs annotated for word correspondences. Since there is no gold standard for German-Romansh, I annotated 600 sentences, which I also gladly make available for future stundents.

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 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 2016). 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 an NER task 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 6.5).

Glossary

Graubünden The Canton of Grisons. 1, 5

Acronyms

AER Average Error Rate. 39, 48, 51, 52, 53, 61

NER Named Entity Recognition. 2

POS Part of Speech. 2

List of Tables

List of Figures

Bibliography

Ahrenberg, Lars (2007). *LinES 1.0 Annotation: Format, Contents and Guidelines*. Tech. rep.

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.

Bossong, Georg (1998). *Die Romanischen Sprachen: Eine vergleichende Einführung*. Hamburg: Helmut Buske Verlag.

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).

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.

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.

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.

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

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*. 1965th ed. 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/.

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 Manul 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.

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/.

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.

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.

Schmid, Heinrich (1982). *Richtlinien für die Gestaltung einer gesamtbündnerromanischen Schriftsprache RUMANTSCH GRISCHUN*. Lia Rumantscha. Chur.

Sennrich, Rico, Barry Haddow, and Alexandra Birch (Aug. 2016). "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.

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.

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.

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.

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

Appendix A

Algnment Examples

Below are some examples for word alignment of the different systems. Green squares are the gold standard, circels are SimAlign and boxes are eflomal.

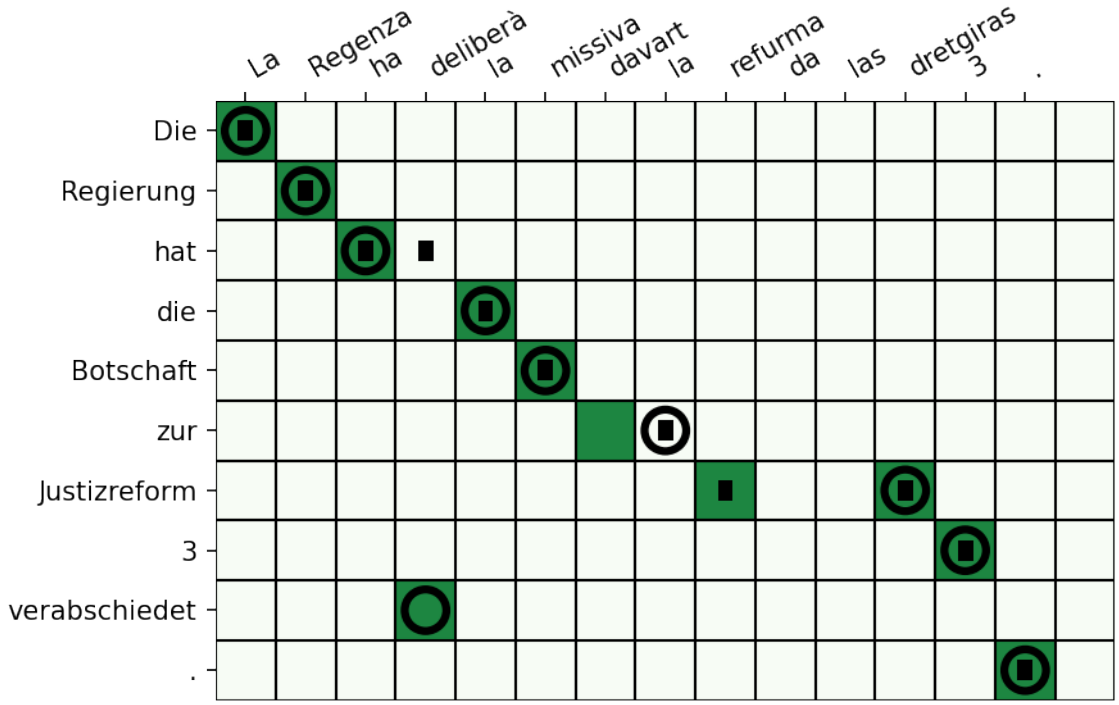


Figure A.1: Word alignment example 1

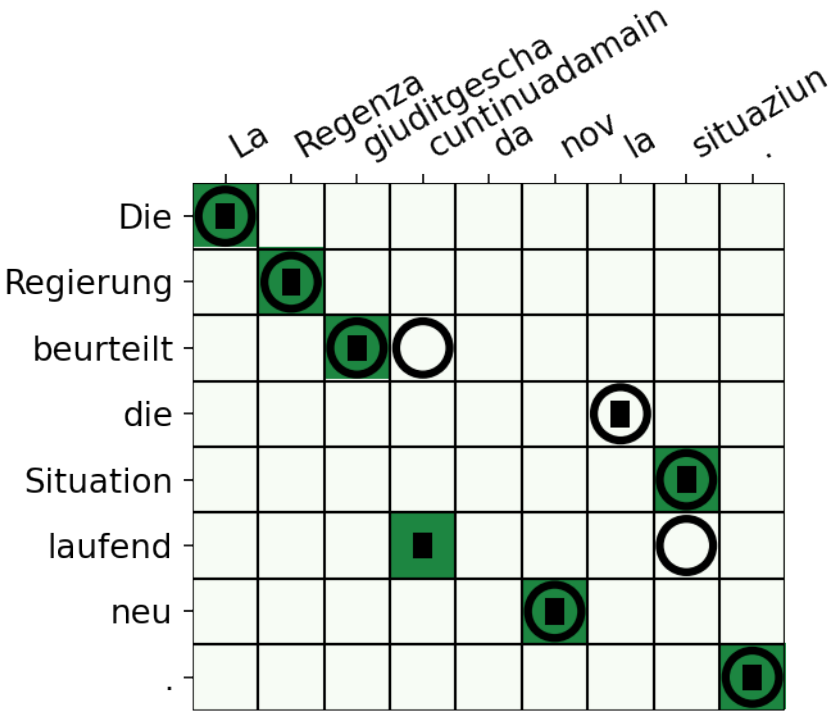


Figure A.2: Word alignment example 2

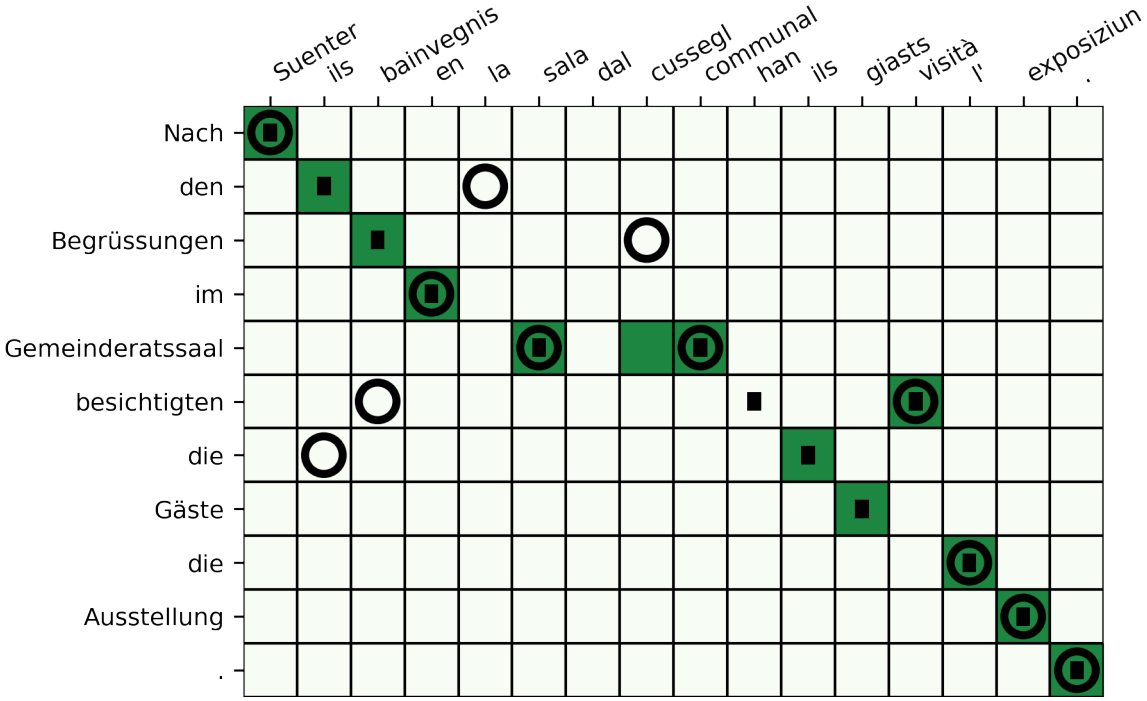


Figure A.3: Word alignment example 3

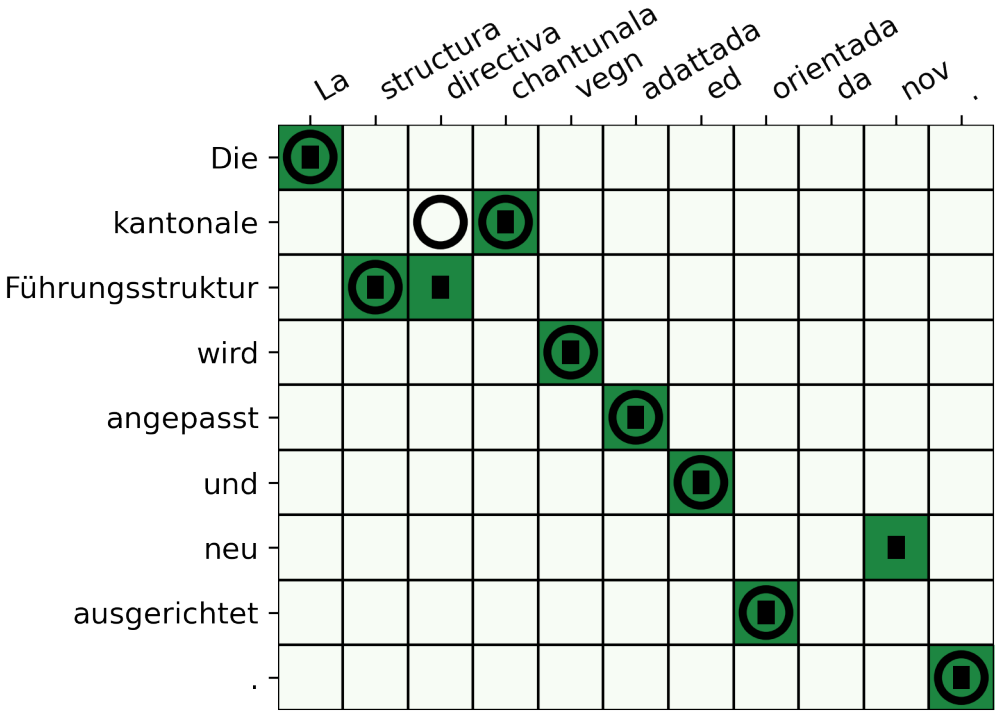


Figure A.4: Word alignment example 4

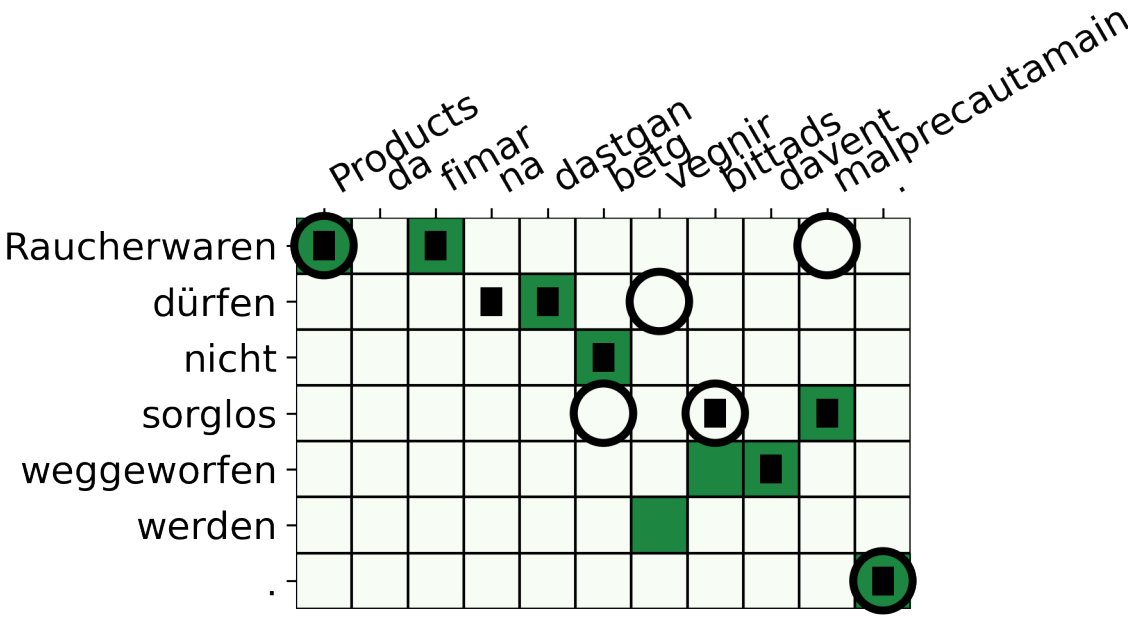


Figure A.5: Word alignment example 5