Contents

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

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

(Koehn 2009, p. 56)

This means that each set in this pair of sets can consist of one or more sentences. The number of sentences in each set is referred to as *alignment type*. A 1-to-1 alignment is an alignment where exactly one sentence of language e is aligned to exactly one sentence of language f. In a 1-to-2 alignment, one sentence in language e is a aligned to two sentences in language f. There are also 0-to-1 alignments, in which a sentence of language f is not aligned to anything of language e. Sentences may not be left out and each sentence may only occur in one sentence pair (Koehn 2009, p. 57).

4.2 Method Overview

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

4.2.1 Length Based

One early method for sentence alignment is 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 is based on the assumption that longer sentences in language e are usually translated into longer sentences in language f and vice-versa—shorter sentences in one language correspond to shorter sentences in the other language.

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

A program based on this method was tested against a human-made alignment on two pairs of languages: English-German and English-French. The program made a total of 55 errors out of a total of 1,316 alignments (4.2%). By taking the best scoring 80% of the alignments, the error rate 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, which is around 3.5 times faster than previous algorithms (Gale and Church 1991).

4.2.2 Partial Similarity Based

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

[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.

are cognates 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, in order to find further isolated cognates within these boundaries, until no more anchor points can be found.

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

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

4.2.3 Translation based

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

4.2.4 Hybrid models

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

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

Another hybrid method was presented by Varga et al. 2005. It combines a dictionary- and a length-based method. Here, a sort of a dummy translation of the source text is produced using a translation dictionary supplied to the program[2]. 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

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

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

4.2.5 Summary

All the methods presented here perform very well on clean, well-structured data in similar languages. Already the Gale & Church algorithm from 1993 achieved a precision of 98% on the Canadian Hansards[3], which Gale and Church acknowledge are easy to align. What seems to have led researchers to develop better sentence alignment algorithms are speed (Chen 1993; Varga et al. 2005) and better performance on noisy data (such as 1-to-many alignments and misrecognized paragraph boundaries (Sennrich and Volk 2010)).

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

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

4.3 More Recent methods

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

4.3.1 Bleualign

One of these methods was presented in Sennrich and Volk 2010 and has been dubbed since as Bleualign. It arose as a method addressing 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 or

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

the Europarl[4] which are considered easy to align because they are well-structured—they provide markup information to identify speakers which is useful for creating anchor points and the subsequent alignment (Simard and Plamondon 1996; Sennrich and Volk 2011). However, when aligning pairs of languages which are fundamentally different and/or of less structured texts, the alignment task becomes more difficult (Sennrich and Volk 2010).

Bleualign uses BLEU as a similarity score to find sentence alignments. BLEU, which stands for Bilingual Evaluation Understudy, is a popular automatic metric for evaluating machine translation models. It measures the similarity between two sentences by considering matches of several n-grams[5] [6]. The higher the BLEU score, the higher the similarity between two sentences (Koehn 2009, p. 226).

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

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

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

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

[4] Parliamentary proceedings of the EU Parliament

[5] Sequences of tokens of length n

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

neural machine translation (NMT) systems. This is especially true in scenarios with less data for low-resource NMT (Thompson and Koehn 2019).

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

4.4 Sentence alignment pipeline

I shall now describe the 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 around 100,000 sentences takes about three minutes).

I tried, just for the sake of interest, to use Vecalign on a small portion of my corpus. 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 (Thompson and Koehn 2019).

The process of obtaining the sentence alignment took quite some time—around 10 minutes for 300 sentences—and by quick inspection with the bare eye, the result wasn't better than the one achieved with hunalign, but rather worse. Obviously, this may be due to the fact that Romansh is not one of the languages LASER was trained on. That being said, LASER *has* been said to generalize to unseen languages that are similar to the ones the model was trained on, e.g., Swiss German or West Frisian, which are similar to German

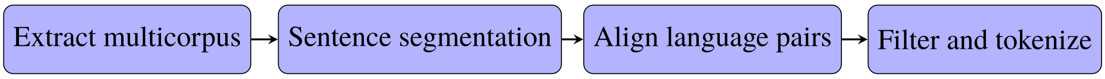

Figure 4.1: Sentence alignment pipeline

and Dutch, respectively (Artetxe and Schwenk 2019) [7].

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 only 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 sentence per line, and one file per year. This is done by querying the SQLite database for all the aligned documents for each year separately.

4.4.3 Sentence segmentation

Sentence segmentation (also called sentence tokenization) was done using NLTK's Punkt tokenizers (Bird, Loper, and Klein 2009). Since I wasn't able to integrate a sentence tokenizer for Romansh into the pipeline, I used the an NLTK Punkt tokenizer model which was trained on Italian. After instantiating both the German and the Italian models, I extended the list of abbreviations[8]to enhance the performance of the tokenizer and avoid wrong segmentation.

In the course of sentence segmentation, paragraphs are retained by converting line breaks 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 for an example.

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

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

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

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

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[9].

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*[10] to support hunalign even further.

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

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

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.

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 an NMT model trained on this corpus (Thompson and Koehn 2019). 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 sentences, one sentence per line and word-tokenized. The script removes sentences containing E-Mails, URLs or phone numbers, as well as sentences where source and target languages are identical or where the difference in sentence length between source and target is too large (more than three times), 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 (Bird, Loper, and Klein 2009) word tokeniziation functions, while applying the German model for German text and the Italian model for Romansh and Italian text. The justification for the latter is that Romansh, in a manner very similar to Italian, uses apostrophes to attach enclitics (articles and pronouns) to neighboring words, which should be separated for word tokenization. An inspection with the bare eye looked precise enough. In the course of annotating the word alignment, I had to correct the tokenization less than 10 times for 600 sentences.

4.5 Results

The resulting final parallel corpus consists of three files containing around 80,000 unique sentence pairs for each of the three language combinations: German–Romansh, German–Italian and Romansh–Italian. Each line in the file contains a sentence pair, separated by a tab character (see Listing 4.2). Table 4.1 elaborates on the number of sentences, tokens and type for each combination.

Combination	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. “Sentences” are sentence pairs. “Source” refers to the language on the left and “Target” to the language on the right, not necessarily to the actual source language of the translation.

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

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

Glossary

Graubünden The Canton of Grisons. 1, 5, 9

Standeskanzlei State Chancellery of Grisons. 9

Acronyms

AER average error rate. 39, 48, 51, 52, 53, 61

NER named entity recognition. 2

NMT neural machine translation. 25, 28

POS part of speech. 2

SMT statistical machine translation. 24

List of Tables

List of Figures

List of Listings

Bibliography

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.

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

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.

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.

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

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/.

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.

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.

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