

Contents

1	Introduction	3
1.1	Motivation	3
1.2	Research Question and Goals	3
2	Romansh	5
2.1	Rhaeto-Romance	5
2.2	Romansh	6
2.3	Rumantsch Grischun	7
2.3.1	Lia Rumantscha	7
2.3.2	Rumantsch Grischun	7
2.3.3	Properties	7
2.3.4	Today	8
3	Compiling the Corpus	9
3.1	Introduction	9
3.2	Collecting the Data	9
3.3	Web Scraping	9
3.4	Building the Corpus	11
3.4.1	HTML Parsing	11
3.4.2	Document Alignment	12
3.5	Manual alignment of unlinked documents	15
3.6	SQLite database	15
3.7	Summary	15
4	Sentence Alignment	17
4.1	Introduction	17
4.1.1	Formal definition	17
4.2	Method Overview	18
4.2.1	Length Based	18
4.2.2	Partial Similarity Based	18
4.2.3	Translation based	19
4.2.4	Hybrid models	19
4.2.5	Summary	19
4.3	More Recent methods	20

4.3.1	Bleualign	20
4.3.2	Vecalign	21
4.4	Sentence alignment pipeline	21
4.4.1	Tool of choice	21
4.4.2	Pipeline	22
4.4.3	Sentence segmentation	22
4.4.4	Aligning language pairs	23
4.4.5	Filtering and tokenizing	24
4.5	Results	24
5	Word Alignment	26
5.1	Introduction	26
5.2	Overview of Methods	27
5.2.1	IBM Model 1	27
5.3	SimAlign	28
5.3.1	Word Embeddings	28
5.3.2	SimAlign	29
5.4	Computing the Word Alignments	29
6	Gold standard	28
6.1	Introduction	28
6.2	Sure and Possible Alignments	28
6.3	Evaluation Metrics	29
6.4	Gold standard for German-Romansh	29
6.4.1	Annotation tool	29
6.4.2	Guidelines	29
6.4.3	General principles	30
6.4.4	Examples	31
6.5	Flaws	33
7	Evaluation	35
8	Contributions	36
9	Summary	37
	Bibliography	37

Chapter 5

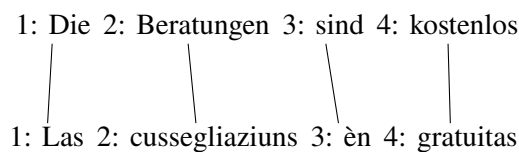
Word Alignment

We now reach the core of thesis, computing word alignments using the novel method SimAlign suggested by Jalili Sabet et al. 2020 and evaluating it against a baseline method.

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*.



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 exactly one word in the target sentence 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. A word in the target language may be aligned to several words in the source language (1-to-many), or several words in the target language may be aligned to one word in the source language (many-to-1). Sometimes words in the target have no relation to the source (for instance in case of untranslatable

words, but words might also generally be 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 review of word alignment methods, that is of the IBM Models, of `fastalign`, which is an improved version of the IBM Model 2 and serves as a baseline model in my experiment, and of `SimAlign`, the model based on the novel method of 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 principle of operation in a more intuitive way 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 language 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 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 w_e in language e was translated as a word w_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 given a sentence in language e .

Unfortunately, while sentence alignment is a relatively easy task (at least for well-structured texts), and while sentence aligned parallel corpora are not hard to compile or come by, we do not know which words correspond to which words in the sentence pairs. 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 unknown 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.

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. Probable links between words are strengthened while less likely weak links are weakened, until the model converges and the most likely word alignments have been learned by the model (Koehn 2009, pp. 88–92; Brown et al. 1993).

5.3 SimAlign

5.3.1 Word Embeddings

Word embeddings are simply put vector representations of words. Embeddings are learned as a sort of byproduct of a neural language model. A language model is a model that assigns probabilities to sentences of a given language—it can say how probable a given sentence is in a language. A language model can also be used to generate sentences by concatenating words based on their probability distributions.

Without going too much into details, when a simple neural language model learns the probability distributions for words given all the previous words, the weights in the hidden layer are adapted. We can then use the language model not for generating language, but extract the weights for a specific word from the inner layer, which are an n -dimensional vector. We call this vector word embedding.

It has been shown that words that occur in similar contexts have similar word embeddings. This vector similarity can be measured with the cosine similarity.

5.3.2 SimAlign

5.4 Computing the Word Alignments

Chapter 7

Evaluation

Chapter 8

Contributions

Chapter 9

Summary

Bibliography

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