Alignment of Historical Manuscript Transcriptions and Translations

Maarten Janssen
UFAL, Faculty of Mathematics and Physics
Charles University, Czechia
janssen@ufal.mff.cuni.cz

Piroska Lendvai
Dept. of Digital Humanities
Bavarian Academy of Sciences, Germany
piroska.lendvai@badw.de

Anna Jouravel
Dept. of Slavic Languages and Literatures
University of Freiburg, Germany
anna.jouravel@slavistik.uni-freiburg.de

Abstract

Using an XML-based framework, we compiled a gold standard for alignments in five primary as well as derived texts, related to *De Lepra ad Sistelium* by Methodius Olympius. These comprise diplomatic transcripts, editions, and translations of this work, involving both historical and modern languages. Using the TEITOK corpus platform, we created sentence-level gold standard alignments for our parallel resp. comparable texts, and applied both neural and classical alignment methods (SentenceBERT, Hunalign, AwesomeAlign). We evaluated the methods in terms of Alignment Error Rate. We show that for alignment of our historical texts, Hunalign performs better than deep learning based methods.

1 Introduction

Historical texts, especially prominent works, often exist in several surviving manuscripts and in their corresponding digital editions. Various types of editions – e.g. scholarly, critical, semi-paleographic – have their own structuring particularities, where variations of a text segment can be placed in an accompanying *apparatus*. Besides the primary text, it might have translations that may exist in one or more languages. All the above variants may be referred to across a text's different editions, either explicitly or implicitly; however, they are rarely (fully) aligned by means of explicit indications of specific text portions corresponding to one another. Manual alignment is a labor-intensive process in many respects: structural differences between editions and language variants need to be resolved, historical languages need to be interpreted, in which human resources are often a bottleneck.

For modern texts, large corpora of aligned data exist, like OPUS (Tiedemann, 2009) and InterCorp (Vavřín and Rosen, 2008), with their infrastructure and handling of automatic alignment. Recent tools can produce alignments of good quality on pairs of texts, cf. e.g. Forgac et al. (2023). These can be directly utilized or used as a basis for manual corrections to obtain full alignment. Comparisons of automatic alignment methods that focus on literal or close translations between well-supported languages are made e.g. by Och and Ney (2003).

However, for automatic alignment of historical material, there are unresolved challenges. (1) These texts are typically more difficult to process using NLP approaches, e.g. due to orthographic variability and under-resourcedness in terms of NLP tools and training data. (2) There may be larger differences between the texts to be aligned, since the source texts, their copies, and their respective translations may differ greatly, and larger spans of texts may be missing from all of these due to material

Proceedings of Recent Advances in Natural Language Processing,pages 462–470
Varna, Sep 8–10, 2025
https://doi.org/10.26615/978-954-452-098-4-055

loss. This requires alignment to function on so-called *comparable* texts as well (e.g. loose translations), besides so-called *parallel* texts (e.g. proper translations). (3) The digital modeling of a fully-fledged alignment structure is not trivial.

In this paper, we apply and evaluate specific automatic alignment methods on the sentence level, across five different variants – including modern translations – of a single, historical work that was fragmentarily preserved in medieval Greek but has an almost complete Slavic translation (see Section 2). The hitherto published editions of this text exhibit different levels of alignment. The earliest printed edition contains no alignment at all; the more recent printed edition provides section-level pairings between medieval text and modern German translation, and otherwise includes references to the corresponding passages in the earlier edition. The modern English translation (accessible online) provides – where the original text is preserved in both languages – the translations of both versions side by side in continuous text.

We describe our workflow that enables the alignment of all our source materials. We focus on sentence-level alignment and not word-level alignment. To obtain a gold standard for our automatic alignment setups, we initially manually and subsequently automatically pre-aligned all these texts. The corpus platform used in our experiments is TEITOK[1] (Janssen, 2016), a system for editing, publishing, and querying annotated corpora, in which each corpus document is stored as a tokenized and annotated TEI/XML document[2]. It enables maintaining available metadata, for example related to formatting, in the source material, including typesetting, footnotes, references, as well as alignments with facsimile images or multi-media sources, so that the documents can be displayed as digital editions.

We show how text versions are converted to a TEITOK/XML document, and how alignment is handled in TEITOK. The alignment methods we investigate comprise classical and neural approaches, but exclude existing tools such as BleuAlign[3], based on automatic translation, or eflomal[4], based on Bayesian models. For the applied use case at hand, we analyze the performance of each alignment method. This allows us to provide further insight about the interplay of alignment methods and text properties. All alignment scripts are available via a Github repository[5].

2 Source Materials

The historical work used in our experiments is *De Lepra ad Sistelium* by Methodius Olympius, an early Christian bishop and theologian active in Lycia (Asia Minor). The original text was written in Old Greek at the turn of the 4th century CE. The text allegorically interprets Old Testament regulations on leprosy (Leviticus 13), using the disease as a metaphor for spiritual and moral afflictions of the soul and of church community. For an overview of the history of its surviving Greek and Church Slavic versions, detailed textual traditions, and a critical edition including German translation cf. Jouravel and Sieber (2024), as well as Maksimczuk (2024).

Original Greek fragments of *De lepra* have survived within the Byzantine *Florilegium Coislinianum*, a compilation from the 9th to the 10th century that preserves quotations from church fathers. Several later copies of the excerpts of the text, in 10th century manuscripts such as *Parisinus graecus 924* and *Atheniensis, Bibliothecae Nationalis 464*. However, the old-

[1] www.teitok.org
[2] tei-c.org
[3] github.com/rsennrich/Bleualign
[4] github.com/robertostling/eflomal
[5] github.com/ufal/histalign

Identifier	Language	Description	Sentences	Tokens
A	grc	Edition by Sieber	52	1762
B	chu	Edition by Jouravel	267	5068
C	deu	Translation by Bonwetsch	241	6584
D	deu	Translation of B by Jouravel	270	6552
E	chu	Transcription of manuscript used for B	702	4561

Table 1: Source Material Overview

est full-surviving manuscripts are preserved in an Old Church Slavic translation dating from the 10th century, which, despite being more complete than the Greek fragments, is heavily abbreviated and paraphrased. It survived in numerous manuscripts from the 16th to the 19th centuries, predominantly preserved in Russian collections.

From the recent German edition of Jouravel et al. (2024), we took the following four (derived) texts of *De lepra*: *(A)* critical edition of the Greek fragments, *(B)* reconstruction of the Slavic translation, *(D)* German translation of the Slavic text, and *(E)* diplomatic edition of the Slavic text. In addition, we included *(C)* the older German edition (Bonwetsch, 1917), which is, however, non-trivial to align, since it does not provide the original texts with their corresponding translations, but rather resembles a patchwork in which the missing Greek passages have been filled in with the German translation of the corresponding sections preserved in the Church Slavic version.

An overview of the five different texts and their identifier letter is given in Table 1, along with an indication of the extent of each text in terms of sentences and tokens.

3 Alignment in the TEITOK Tool

All documents were converted to TEI/XML. For *A-E*, we used a digital version of the source material of (Jouravel and Sieber, 2024), kindly provided to us by the authors, and natively converted to TEI by the Classical Text Editor[6] in which it was created. This TEI was subsequently manually adapted for use in the TEITOK environment.

For the current task, we provide alignments at the level of chapters, paragraphs, and sentences, while word-level alignment will be enabled in future work. The alignment in TEITOK is set up as correspondences between two or more full-fledged TEI/XML documents. These correspondences are defined by an attribute `@tuid` (translation unit ID) - two nodes in two documents are aligned if they share a `@tuid`, which can be placed on any type of node. Translation units (TUs) do not have to be translations; they can also be copies or versions in the same language. A `@tuid` can be a list of values to link an element in a document to multiple elements in another.

In translation, it is common to group translation units by segments. In segment grouping formats such as Translation Memory eXchange[7] (TMX), there is a single document containing all translations, where a text is divided into translation units, and each unit lists the version of that segment in each language. The TEITOK strategy has the advantage that it can combine various levels of alignment at once, thus it can better deal with cases where the order of the elements is not the same in all versions.

In turn, automatic alignment typically proceeds in a pairwise fashion, i.e. aligning only two texts - a TEI compliant format for which

[6] cte.oeaw.ac.at/

[7] localizely.com/tmx-file/

is XCES (Ide et al., 2000). Pairwise alignment has the advantage that it can cater better to the particularities of each pair, often introducing pair-specific segmentation, and no concessions have to be made when a third language is added. However, pairwise alignment is more difficult to maintain when manual corrections are introduced since each pair has to be maintained independently. It makes it necessary to either have alignments between each pair of languages or to use a single language as a pivot. Pairwise alignment is also slightly more complicated to exploit in display and search.

For the current study, we created an interface to handle pairwise alignment in TEITOK. The pairwise alignment is stored in JSON, as an array of translation elements, where each element specifies the ID of the source and target, as well as the text of the source and target. This format is easily convertible to and from both TMX and XCES where needed, as can be generated from the output of most automatic alignment tools. The interface can visualize a pairwise alignment and edit it, and can compare the alignment to a gold standard to highlight the errors (see figure 1). We also added scripts to load a pairwise alignment into the `@tuid` representation, and create a pairwise alignment from two TEITOK documents adorned with `@tuids`.

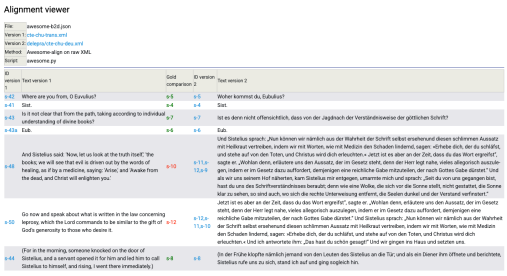

Figure 1: The automatic alignment view

There are various ways to utilize the alignments in TEITOK. There are two main visualization methods. The first visualization displays two or more versions of a text next to each other, and moving the mouse over a segment in one text will highlight the same segment in other versions. This is similar to the visualization by, for instance, the Versioning Machine (VM) (Schreibman et al., 2003)[8]. However, VM only supports a single level of alignment, whereas TEITOK can have alignments on several levels in the same document. The interface always shows the first ancestor of the node you are hovering over that has a `@tuid`. This visualization is illustrated in figure 2.

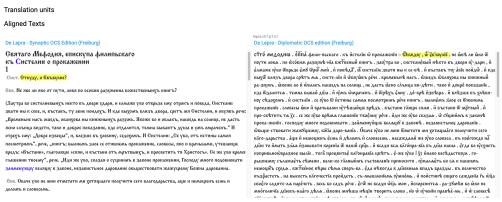

Figure 2: The TEITOK alignment view

Another visualization starts from a single alignment level in one document and displays a table with that alignment with all corresponding elements in one or more other documents. This has a number of consequences. Firstly, the order of the elements is shown as it appears in the source, meaning that the order in the target documents can get shuffled. Secondly, only the corresponding elements are shown, so if the target documents contain text that lacks alignment at that level, it will not be displayed. Also text in the source document can be left out: if an alignment is shown at the level of paragraphs and the source contains text that is not in a paragraph, it will not be shown.

[8] v-machine.org/samples/prophecy_of_merlin.html

3.1 Gold Standard Alignment

To create a gold standard alignment for the source material described in Section 2, we started from the (B) version, and then gradually aligned each of the other texts to the most similar text that had already been manually aligned. This was done by running an automatic alignment tool (see Section 4) to obtain a JSON pairwise alignment representation, after which we incorporate that JSON alignment into the XML alignment. The XML alignment in the newly incorporated version was then manually corrected.

For manual correction, we adopted the aligned visualization in TEITOK so that, for corpus administrators, it can also be used to edit the alignment, as shown in figure 3. In the image, the alignable element `lepra:ch1:p2:s1` (chapter 1, paragraph 2, sentence 1) relates sentence s-43 in version B to the sentences `s-3b` and `s-4` in version E. To change this alignment by, for instance, realigning `s-3b` to a different `@tuid`, it is sufficient to click on the `s-3b` to select it, and then select on the `@tuid` to which it should be attached. This will modify the underlying XML, and reload to the corrected alignment.

This editor, after some interface optimization, proved to be efficient for correcting the alignment, and with the relatively low amount of elements to needed to be manually realigned (which can be calculated from the AER of each pair in the automatic alignment and the sentence count of the documents involved), the manual alignment correction took an average of 1,5 hours per alignment pair. The manually aligned pairs were B-E, B-D, B-A and finally A-C.

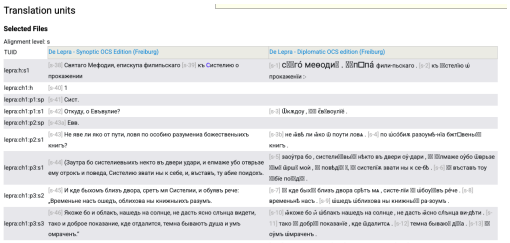

Figure 3: The TEITOK alignment editor

4 Automatic Alignment Methods

In this section, we will evaluate the accuracy of various automatic alignment methods by comparing the output of the automatic script to our gold standard aligned corpus for each pair of documents. This is done by calculating the Alignment Error Rate (AER - (Och and Ney, 2003)), which combines the recall and precision of the automatically generated alignment with the gold standard alignment incorporated and manually corrected our sources.

In order to create a workflow for automatic alignment that includes testing different alignment methods, we implemented several Python scripts that take two TEITOK/XML documents as input and create a JSON file as output, with the design described in Section 3. In the evaluation, we generated a JSON representation of the gold standard alignment in the TEITOK/XML files and compared it with the output of the various alignment scripts.

Automatic alignment is tested at the sentence level, thus throughout this study, the alignment present above the level of sentences (paragraph and chapter alignment) is ignored, whereas alignment below the level of sentences (word alignment) is typically not yet provided.

Note that the issue of sentence boundary ambiguity in Church Slavic, cf. Jouravel et al. (2024), is not problematic for us, since our input is pre-segmented.

4.1 Hunalign

The first alignment method that we implemented uses Hunalign (Halácsy et al., 2007), run as a command-line tool. Its output is converted into a pairwise JSON representation.

Hunalign is based on Gale-Church sentence-length and is widely used, for instance, in OPUS and InterCorp. Given its popularity, Hunalign serves as a baseline for alignment. It is language agnostic and as such not hampered by less resourced languages, although it can be made language aware by providing a translation dictionary for a specific language pair. We could not utilize this feature, given that no machine-readable translation dictionary is available for Old Church Slavic. We are aware that using a translation dictionary would improve its performance.

Table 2: AER for Hunalign

	A	B	C	D	E
A		1.000	0.711	1.000	1.000
B	1.000		0.439	0.100	0.126
C	0.711	0.439		0.139	0.940
D	1.000	0.100	0.138		0.949
E	1.000	0.126	0.940	0.944	

The accuracy of all 16 document alignments using Hunalign is given in Table 2. The scores show that the performance of Hunalign for manuscript alignment is inconsistent. It performs well in alignment of some pairs (BD, BE, and CD), performs marginally well in CD and AC, and does not handle the rest well. The reason behind the latter is that Hunalign assumes all sentences need to be matched. Since text A is much shorter than the rest, most sentences should remain unaligned. The reason why it still aligns marginally with C is that C contains some of the sentences of the Greek text verbatim, meaning that those sentences tend to get correctly aligned.

Also, the sentences in E are much shorter than in the other texts, which means that there is a need for several *1:n* relations. For text B this works well, since texts B and E are word for word identical except for titles and transcription, meaning that Hunalign manages to merge sentences.

4.2 Sentence Transformer

Many recent alignment methods are based on large language models, often involving SentenceTransformer or sBERT[9]. Our implementation uses the PolyFuzz package[10] for the alignment based on the LLM model. The model we used is the language-agnostic BERT Sentence Embedding (LaBSE) model (Feng et al., 2022). LaBSE typically scores well on cross-linguistic sentence alignment since it was trained on more than 100 (modern) languages.

Table 3: AER for LaBSE

	A	B	C	D	E
A		0.829	0.732	0.825	0.987
B	0.940		0.405	0.142	0.835
C	0.888	0.379		0.321	0.949
D	0.934	0.191	0.374		0.938
E	0.995	0.840	0.941	0.907	

The accuracy of all 16 document alignments using LaBSE is given in Table 3. Despite the typical advantages of LLM, the LaBSE model is not particularly good at aligning sentences for historical documents for the languages at hand. It does not outperform Hunalign, except for the cases where Hunalign does not provide any correct alignment.

A surprising observation is that LaBSE is less affected by not having been trained on Old Church Slavic (OCS), since it performs well for the pair B-D. Nonetheless, it is impacted by mismatches in sentence segmentation: despite the fact that the pair C-D involves two languages on which this model was trained, its performance is worse than on the BD pair where sentences are more similarly segmented.

[9] https://sbert.net/

[10] maartengr.github.io/PolyFuzz/

4.3 Word-level Alignment

One of the challenges that both Hunalign and LaBSE face is that not all sentences should be aligned. Therefore, instead of aligning sentences, an option could be to start by aligning words. To test this, we added an implementation of *Awesome-align* (Dou and Neubig, 2021), which is a tool for word-level alignment using multilingual BERT. Awesome-align is designed to take aligned sentences as input and produce word-level alignment as output. However, we can provide Awesome-align with longer segments, such as a paragraph, to get a word-level alignment (of course not as good as when used on sentences), and then use that word-level alignment to calculate an alignment between sentences.

Awesome-align is far from the only word-level alignment tool; a notable alternative is Ugarit (Yousef et al., 2022), which has been trained and tested on historical languages, including ancient Greek (though not OCS). Other popular tools are Giza++[11] and simalign[12]. However, testing the differences between a multitude of different word-level alignment tools is beyond the scope of this study.

It is common in alignment pipelines to progressively align texts by first aligning paragraphs before sentences and words. This is, for instance, what InterCorp does, where the paragraph alignment is partially manual. However, historical manuscripts often do not have paragraphs: There are no paragraphs in E, and the paragraphs that are present in B and C are interpretations of the text. Also, the chapter and paragraph breakdown in the two versions is quite different. Therefore, when paragraph alignment fails, the only resort is to use the entire text as input.

The accuracy of all 16 document alignments using Awesome-align is given in Table 4. In this, only the pair B-D uses paragraph level, while all other alignments are done based on the entire text. Awesome-align performs badly when it is used on an entire text, and the only pair that performs well is the pair B-D that was aligned starting from paragraph level. Nevertheless, it does not outperform either Hunalign or LaBSE even when starting from the paragraph level. This is not entirely surprising given that the tool was designed to refine sentence level alignment to word level alignment, and not for the task it was used for here.

Table 4: AER for Awesome-Align

	A	B	C	D	E
A		1.000	1.000	1.000	1.000
B	1.000		0.931	0.344	0.943
C	1.000	0.931		0.898	0.956
D	1.000	0.344	0.898		0.953
E	1.000	0.943	0.956	0.953	

We made the mapping from words to sentences in an unrestricted way: whenever any word in sentence X is aligned with a word in sentence Y, those two sentences are assumed to be aligned. For future studies, this could be further refined by adding a threshold that requires a minimum percentage of the words in both sentences to be aligned. However, given the low accuracy of the method, it is unlikely that it will radically improve the results.

4.4 Alignment using LLM-based translation

Various successful alignment tools, such as Bleualign, use a language-aware type of alignment by combining alignment with translation. Bleualign takes three input texts: the source, the target, and an automatic translation of the source into the target language. In order to emulate this, we generated a sentence-by-sentence

[11] https://www2.statmt.org/moses/giza/GIZA++.html

[12] github.com/cisnlp/simalign

automatic translation of the B text into English, keeping the same ID and `tuid` for each sentence as in the original. In this way, by aligning the English text, the OCS text implicitly gets aligned as well.

Since there are no dedicated automatic translation tools for OCS, we resorted to general AI tools for generating a translation. In particular, we used Llama3.3[13], and asked for a sentence-level translation that kept the sentence IDs. Since Llama does not handle large input requests well, the sentences needed to be treated in batches.

Table 5: Translation-based AER

	Hunalign	LaBSE	Awesome-Align
A	1.000	0.934	1.000
B	0.019	0.074	0.373
C	0.400	0.358	0.893
D	0.095	0.142	0.301
E	0.807	0.899	0.954

The AER results involving the automatic translation are given in Table 5. Translation before alignment has a positive effect, mostly when using awesome-align, though not massive. The exception is *B-E*, which is expected since the strong similarity disappears in the translation. The alignment of the translation of *B* with its original is good in Hunalign and LaBSE, but surprisingly bad in awesome-align.

5 Summary and Conclusion

In this paper, we looked at the use of automatic alignment tools on a collection of historical documents. We found that the performance of the approaches tested leaves room for improvement. Historical alignment faces major challenges, such as substantial differences between versions of the same text and data sparsity. Despite LLM models being able to increase accuracy for many NLP tasks, including alignment of modern, well-resourced languages, for the alignment of historical transcriptions and translations, none of the LLM based approaches outperformed Hunalign.

Even though our tested approaches are suboptimal for automatically aligning most pairs of our historical documents, we found cases where our method was sufficiently accurate to speed up the creation of their fully aligned set; e.g. by automatically aligning well-performing pairs and then manually correcting them. Employing a setup we showed using the TEITOK platform, where linking is done by shared attributes (`@tuid`), translation alignment will grow by transitivity with each new document aligned to one of the already aligned documents, eventually leading to a fully aligned set of documents. This work well in the TEITOK interface, since it facilitates the manual correction of automatic alignment.

One of the major reasons why LLM models do not perform well on the alignment of our historical data is that the relation between these documents is quite different from that in modern translations, and none of the tested LLM models was trained on this kind of data. Creating more gold standard data in line with those described in this study will enable investigating whether LLM models can be specifically adapted to this kind of data on the downstream task of automatic text parallelization.

The comparison in this study is limited in two respects. Firstly, it only tests a single text. This gives an indication of the extent of automatic alignment performance in a specific set-up on a small collection of historical works. Secondly, a limited set of automatic alignment methods is explored. But more methods and data can be easily added; our future work targets a more extensive study on a larger corpus.

[13] www.llama.com/docs/model-cards-and-prompt-formats/llama3_3/

References

D. G. Nathanael Bonwetsch, editor. 1917. *Methodius*. J.C. Hinrich, Leipzig.

Zi-Yi Dou and Graham Neubig. 2021. Word alignment by fine-tuning embeddings on parallel corpora. In *Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic bert sentence embedding.

Fero Forgac, Dasa Munkova, Michal Munk, and Lívia Kelebercová. 2023. Evaluating automatic sentence alignment approaches on english-slovak sentences. *Scientific Reports*, 13.

Péter Halácsy, Andras Kornai, Viktor Nagy, László Németh, and Viktor Trón. 2007. *Parallel corpora for medium density languages*, pages 247–258.

Nancy Ide, Patrice Bonhomme, and Laurent Romary. 2000. XCES: An XML-based encoding standard for linguistic corpora. In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC'00)*, Athens, Greece. European Language Resources Association (ELRA).

Maarten Janssen. 2016. TEITOK: Text-faithful annotated corpora. *Proceedings of the 10th International Conference on Language Resources and Evaluation, LREC 2016*, pages 4037–4043.

A. Jouravel, E. Renje, P. Lendvai, and A. Rabus. 2024. Assessing automatic sentence segmentation in medieval slavic texts. In *Proceedings of Digital Humanities Conference (DH-2024)*.

Anna Jouravel and Janina Sieber. 2024. *The Greek and Slavonic Transmission of Methodius' De lepra*, pages 11–30. De Gruyter, Berlin, Boston.

José Maksimczuk. 2024. *The Florilegium Coislinianum and the Greek Text of Methodius' De lepra*, pages 31–54. De Gruyter, Berlin, Boston.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Susan Schreibman, Amit Kumar, and Jarom McDonald. 2003. The versioning machine. *Literary and Linguistic Computing*, 18(1):101–107.

Jörg Tiedemann. 2009. News from OPUS - a collection of multilingual parallel corpora with tools and interfaces. In *Recent Advances in Natural Language Processing*, volume V, pages 237–248.

Martin Vavřín and Alexandr Rosen. 2008. Intercorp: A multilingual parallel corpus project. pages 97–104.

Tariq Yousef, Chiara Palladino, David J. Wright, and Monica Berti. 2022. Automatic translation alignment for Ancient Greek and Latin. In *Proceedings of the Second Workshop on Language Technologies for Historical and Ancient Languages*, pages 101–107, Marseille, France. European Language Resources Association.