Exploiting Parallel Aligned Treebanks

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

Abstract

In this paper we discuss parallel aligned treebanks, mostly within the context of the way it is implemented in TEITOK: a framework that supports word-level aligned, as well as multi-level text alignment (text, paragraph, sentence). It also provides various ways to visualize aligned data, including a newly introduced parallel visualization for dependency trees, with mouse-over aligned highlighting. There is also a parallel search function under development, that allows queries and statistics that are both tree-capable and multi-level alignment capable, including word-level alignment.

Keywords: parallel corpora, corpus querying, treebanks

1. Introduction

Universal Dependencies (UD) is the most successful framework for cross-linguistic grammatical corpus annotation. It is so popular in large part because it provides a homogeneous way to annotate any language, making it hence possible to compare languages. By default, UD only provides comparable data, making it possible to run searches in two different languages, to get statistical information about the differences between those languages.

But by default, UD treebanks do not provide a direct comparison between languages where we can see how a specific phenomenon is expressed in one language and another. This issue was already addressed in 2017 by the introduction of parallel aligned treebanks. Parallel aligned treebanks make it possible to compare the same sentence in different languages and check for differences in the construction used in those languages.

There are various tools that provide different types of access to parallel treebanks to allow a direct comparison. But few that provide full access to aligned data, especially not with token-level alignments. In this paper, we present the implementation of parallel corpus access in TEITOK that is intended to provide full access to treebanks that have been aligned at various level, from text to token. Currently this approach is fully involved in the TEITOK source code where parallel visualization of aligned (treebank) data is concerned. Much of this visualization has a long history, but only recently made available, and the parallel visualization of token-level aligned dependency trees is completely new.

The final component, presented here but not yet released, is a search function that allows searching through, and doing frequency calculations on, parallel aligned dependency treebanks. What is outside of the scope of this paper is the alignment process itself - this paper only discusses how to utilize the alignment once it is encoded in the data. There are many publications on (semi) automatic alignment - for a discussion in the context of TEITOK, see (Janssen et al., 2025).

2. State of the Art

2.1. Parallel UD Treebanks

The first attempt at creating parallel aligned treebanks was made at the TLT (Treebanks and Linguistic Theories) conference in Cairo, where, during a workshop, a set of 20 sentences was translated into various languages, discussed, and annotated with UD. This formed the basis for the Parallel Universal Dependencies project (PUD) (Zeman et al., 2017), released in UD2.1 with a set of 1000 sentences in 18 languages (currently, 22 languages have PUD alignments). Various other projects follow the same idea, such as TueCL (Akhundjanova et al., 2025) for Turkic languages.

In recent versions, parallel sentences in UD treebanks are explicitly marked with a tag *parallel_id* marking various sentences in treebanks as parallel versions of the same sentence. Table 1 reports on the parallel sentences in UD2.17, indicating the project the parallel sentences belong to, the number of languages in which (at least some of) those parallel sentences occur, and the number of parallel sentences involved. Notice that JaGSD is a special case of a parallel corpus between different variants of the same language.

Project	Languages	Sentences
Cairo	22	20
PUD	22	1000
Bible	4	593
Atis	2	5432
Set	2	1489
Hk	2	1004
JaGSD	1	8100

Table 1: Parallel Sentences in UD2.17 Treebanks

Proceedings of the Ninth Workshop on Universal Dependencies (UDW 2026), pages 252–257
May 16, 2026. ©ELRA Language Resources Association (ELRA), 2026

Having the same sentences in various languages, all analyzed within the same (UD) paradigm, allows for a more direct comparison between the constructions used in the languages involved by comparing the actual dependency trees involved. And the various sizes and number of languages involved listed in table 1 provide useful data for a number of different scenarios.

There are, however, two drawbacks in the use of parallel aligned UD treebanks for comparing languages. The first is their size: as gold standard data, the treebanks are by nature modest in size, which reduces their usefulness for language comparison. Especially the 20 sentences in Cairo are more intended to help set up a treebank for a new language by having similar data to compare to than for serious analysis.

And the second drawback is the fact that there is no interface to actually find and compare the trees. With the currently available tools, you have to find a sentence in some source language, either directly in the conllu files or using one of the existing tools (PML-TQ, TEITOK, Grew-Match, or INESS). Once you have obtained the sentence, obtain the *parallel_id* for it, and look for the same ID in the target language (for which there are currently no tools). And then draw a tree for both sentences to compare them. The only exception is Grew-Match, that for SUD (surface UD) treebanks in the PUD collection provides the option to view the aligned sentences in any of the other SUD/PUD corpora.

The first of these two drawbacks is overcome by using (automatically annotated) parallel UD corpora instead of gold standard treebanks, as discussed in the next section. And as will be demonstrated there, the second point is also addressed in the GUI interface for some of the existing parallel UD corpora.

2.2. Parallel UD Corpora

When it comes to parallel corpora, two projects stand out: OPUS (Tiedemann, 2009) and InterCorp (Rosen et al., 2023).

OPUS is a resource infrastructure enabling access to parallel corpora. It is the largest available source of parallel data, and has an impressive 100B sentences in over 1000 different languages. All data are first sentence-aligned, and then word-level aligned using eflomal (Östling and Tiedemann, 2016). Where possible, the data are tagged and parsed as well, meaning that in the corpus query system that OPUS provides, we can search for data using the Corpus WorkBench (Evert and Hardie, 2011) to search for example sentences by POS tag or dependency relation. However, at present OPUS is not annotated in UD throughout, meaning that in order to get parallel UD dependency trees, it is necessary to first run the source and target sentence through a UD parser before being able to compare the trees.

InterCorp is a multilingual parallel corpus. Version 16UD contains 4800M words spread over 62 languages, of which 47 languages are annotated with UD, using UDPIPE[1].

InterCorp is searchable in Kontext[2] (Machálek, 2020), which makes it possible to search through the aligned texts in InterCorp. When executing a query in one of the languages, you can select an aligned language, and all results will be displayed alongside their translation in that language. In addition, it is possible to add search restrictions on target sentences.

For the source language, the full dependency tree can be visualized, and when clicking on a word in the search result, the corresponding word in the translation will be highlighted. This is not done using pre-recorded word-alignment, since Kontext only allows alignment on a single level, but rather is calculated on-the-fly. It does this using an alignment service developed by LINDAT as part of their translation service (Poláková et al., 2025).

So the Kontext interface for InterCorp makes using parallel UD corpora a lot more user friendly: we can use CQL to find examples, we can automatically get the aligned sentences, and we can put CQL restrictions on those aligned sentences. Furthermore, we can directly see the tree for the source or target sentence and can see which words in the source and target language correspond.

But there are still various things that are not that easy: since Kontext uses CQL, which is not a tree query language, we cannot search for dependencies. The word-level alignment is calculated, but cannot be used in the display of the trees, and cannot be used in searches in any way.

In conclusion, there are many tools available that can partially exploit parallel UD treebanks. However, there is currently no method for word-level alignment in treebanks, there is also no visualization for parallel treebanks. And search options are currently either not supporting parallel corpora (like PMT-TQ or Grew-Match) or do not support tree queries (like Kontext).

3. TEITOK

TEITOK (Janssen, 2016) is an online corpus platform. It provides increasing amounts of support for UD (Janssen, 2018). It also provides support for minority languages and for parallel aligned corpora.

The design for working with word-level parallel aligned corpora in TEITOK relies on three

[1] https://lindat.mff.cuni.cz/services/udpipe/

[2] https://www.korpus.cz/kontext/

things: the representation and creation of the parallel aligned corpora themselves, the visualization of the parallel sentences and dependency trees, and search options to search through the parallel aligned dependency trees. The first two parts are fully functional, with some existing parts and some new additions. The search functionality is still under construction as part of a larger overhaul of the search options in TEITOK.

3.1. Parallel Corpora

TEITOK is not inherently a UD tool, but rather a general-purpose corpus tools that provides an increasing amount of support for UD. It does not work natively with the CoNLL-U format, but rather a corpus consists of a collection of annotated TEI/XML documents. For the representation of the core UD token attributes, this is merely a difference in representation: where CoNLL-U stores token attributes as columns on a line representing the token, in TEITOK they are stored as attributes on an XML node representing the token. There are various scripts to seamlessly convert between the two functionally identical formats.

The use of TEI/XML makes for a more standardized and flexible representation of various other attributes. For sentence, paragraph, and text level attributes, there are well established standards, while for the comment lines performing this job in CoNLL-U there is currently no official standard. And for the representation of the information stored in the MISC column in CoNLL-U, there is in various cases a better established alternative in TEI/XML, for instance named entities are stored as nodes around tokens, not as information spread across various tokens in one of the formats UD supports for named entities, for instance the IOB format.

For the representation of alignment between nodes at any level, TEITOK uses basically the same strategy as applied in UD: There is an attribute that specifies an ID (the *parallel_id* in CoNLL-U, and `@tuid` in TEITOK, for *translation unit ID*), and when two nodes share the same ID, they are aligned. So, converting parallel treebanks of UD to the TEITOK format is trivial, by converting the *parallel_id* to a `@tuid` attribute on the relevant sentence node.

There are, however, two differences. Firstly, TEITOK can represent alignment, by a shared `@tuid` at various different levels at the same time - including larger units like paragraphs, or smaller units like tokens, with the same interpretation as those at sentence level: two tokens that share a `@tuid` are translations of each other. UD allows for the use of *Ref* in the MISC column, but *Ref* is just intended to handle partial correspondences between sentences, not alignment at token level. That is, however, merely a matter of practice and not a limitation of the CoNLL-U format: when exporting to CoNLL-U, TEITOK exports token-level alignments as parallelId in the MISC column with the same syntax as that used in the parallel_id at the sentence level.

So token-level alignment is handled in exactly the same way as any other level of alignment, and a document can have alignment at multiple levels, in fact it is uncommon to have word-level alignment without having sentence-level alignment as well, and typically sentence-level alignment in turn relies on paragraph-level alignment.

The second difference is that, for partial correspondences, `@tuid` attributes are not single IDs, but sets of IDs. The simplest case is when a single Spanish sentence is translated as two (or more) sentences in English (which is quite common due to different habits), the various English sentences each have a unique `@tuid`, while the Spanish sentence has `@tuid` , which is a list of all those IDs. Although this can become very complicated in theory, we have found that in practice the system is quite workable. A discussion of a parallel corpus made with this representation at the sentence level can be found in (Janssen et al., 2025), where we show that even in cases where we align various versions of an ancient Greek text in copies or translations produced several centuries later, assigning `@tuid` values is meticulous work, but not due to restrictions of the representation method, but purely due to the analysis of the texts themselves. That paper also describes a set of scripts that produce an alignment in TEITOK using a variety of different existing automatic alignment techniques.

The TEITOK tools repository[3] also provides a number of scripts that can convert from a number of existing aligned formats, most notably the TMX (Translation Memory eXchange) to a TEITOK corpus. This makes it possible either to take already aligned data from for instance OPUS or InterCorp, or to use the tools developed by those projects to align new data and then convert them to the TEITOK format.

3.2. Visualization

TEITOK provides various ways to visualize parallel texts, depending on the properties of the documents to be visualized. The most basic visualization is a table view in which each row shows an aligned pair, in much the same way as translation memory visualization works, or the search results for parallel corpora in InterCorp; if there are more aligned texts, multiple versions can be aligned in columns.

For straightforward cases of alignment, like the Cairo sentences in the UD treebanks, this visualization works very well. However, the more flexible set-

[3] https://github.com/ufal/teitok-tools

up in TEITOK creates a number of different problems. Given that a TEITOK document can have alignments on multiple levels, we have to indicate the level at which we want to see the alignment, typically choosing between sentences or paragraphs. This will create a table with all elements of that level in the left-most column (the source document). Since alignment does not have to be perfect, this has a number of consequences. Firstly, a single row in the source document can map to more elements or less elements in the other columns. So, if we align Spanish-English with the English text having shorter sentences, a single row will contain one Spanish sentence, but multiple English sentences. In the opposite direction, multiple English rows will show a single Spanish sentence along with it. Secondly, the table will only display the plain text of each sentence, leaving out the typesetting the TEITOK document might contain, since the typesetting might overlap with the table structure. Third, it will show the aligned elements in the order in which they appear. Especially in historical documents, the order in the copy or translation can vary, so the remaining columns can show the text shifted from its original order. And lastly, it will only show the selected elements in the source document and their translations. This not only means that if the target documents contain text that is not in the source document, they will not be shown. But even text outside the selected level will not be shown - if we select paragraphs, then any text that is not in a paragraph will not be displayed.

For documents in which the restrictions listed above make the visualization as a table not work, there is also the option to display entire documents next to each other, including any XML annotation they contain. This will not directly display any type of alignment, but moving the mouse over the text will highlight the first parent of the node the mouse hovers over that has a `@tuid`, and highlight that same element in all the other columns. In case there is no matching element for the first parent, but there are ancestor nodes that do have an alignment, that aligned element will be highlighted in a different colour.

Neither of these visualizations display or rely on dependency trees. Therefore, we added an additional visualization that displays parallel dependency trees. This visualization can be seen in figure 1, which shows a sentence from the NTREX corpus (Federmann et al., 2022), an English sentence on top, and the Czech translation on the bottom. Moving the mouse over a word in the tree will not only highlight that word in the full sentence, with all UD attributes of that node shown below it, as the normal dependency tree view does in TEITOK, but will also highlight the corresponding word in both the translated sentence and the dependency tree for the translation. In this case, it shows the word *Trump* in English, with all token-level attributes, and shows the position of the word in the tree. And at the same time, it highlights the word *Trumpovi* (Czech uses inflection on proper names) with its position both in the sentence and the dependency tree.

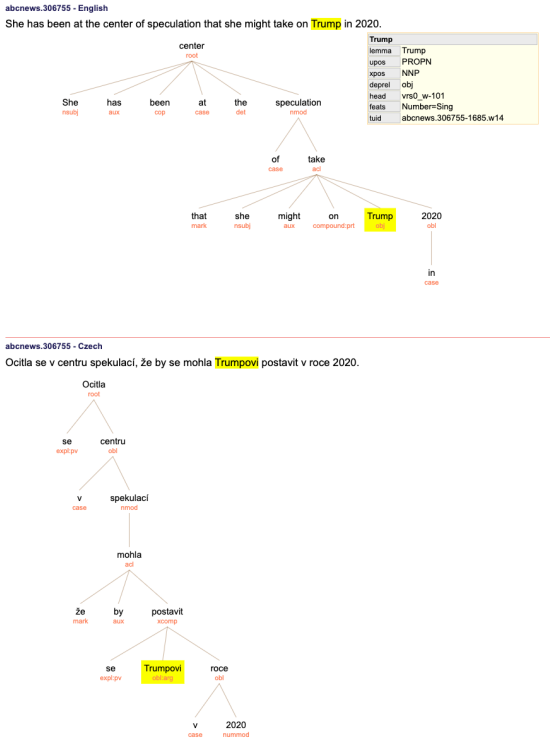

Figure 1: The TEITOK alignment tree view

This aligned dependency tree visualization makes it much easier to compare constructions between languages in actual use cases, even for a language that is not completely familiar, since the UD annotation will display all attributes and relations of the sentence in question.

3.3. Searching

Of course, in order to really exploit word-aligned parallel UD corpora, one would also want to be able to search through them and perform statistics on them. Currently, there is no platform that allows that. The main reason is that a quick survey suggests that there are no tree search algorithms at the moment that are fast enough to work with larger corpora. This is why the only platform listed on the UD website for searching that allows searching through all languages in UD2.17 at once, and does so because it only allows CQL searches. In order to allow searching through larger aligned corpora that allow at the same time tree-sensitive searches and alignment-sensitive searches, at the very least, we need a fast enough tree-sensitive search algorithm.

We are currently finalizing a new corpus search implementation based on translating a language-driven search query that is an extension to existing corpus query languages. And this query system adds the option to perform queries on aligned corpora. Not only in the way that, for instance, Kontext provides, where we can formulate a query, and then filter the results on sentences where the aligned sentence matches a different query, but also by a more expressive syntax that makes full use of the TEITOK-style alignment using @tuid attributes, by restricting searches to cases where the source and the target appear in a sentence or a paragraph with the same (or a shared) tuid value.

It can also make use of word-level alignment. This can be a simple use, by checking what is the most common translation of a given word. Or it can exploit dependency relations: we can, for instance, search for a noun used as *nmod* in Spanish, and then for all words in the English text that are translations of those words, in order to see what is their most frequent dependency relation, or what is the most frequent lemma of the parent node in question.

4. Conclusion

In this paper, we have shown a new approach in TEITOK to building and exploiting word-level parallel aligned UD corpora. This involves creating the alignments, which in TEITOK is done either by alignment scripts or by importing already aligned data. It also involves storing the alignment in the corpus, which in TEITOK is done by storing alignment IDs on any level, be it a text, a paragraph, a sentence, or a word.

It also involves visualizing the alignment, which in TEITOK can be done in a number of different ways: as a table of aligned elements; as full documents displayed alongside each other, where the highlighting is done on mouse-over. And a new visualization in which multiple dependency trees can be displayed next to each other, where on mouse-over the alignments are shown directly in the aligned trees.

And finally, to really exploit parallel UD data, a search system with statistical data would be needed. This search system is still under construction and is intended to allow users to search through aligned treebanks by making use of alignments either at the level of sentences or at the level of words.

4.1. Limitations

Even though much of the system for exploiting parallel UD corpora in TEITOK is in place, there are various limitations. The most obvious limitation is that the search functionality that allows searching through parallel dependency trees is not yet released. For many cases, the new search algorithm is considerably faster than alternatives we have tested, it will have to be shown in practice how scalable the approach really is, whether there is a recommendable maximum size above which search queries become too slow.

Even though the encoding of word-level alignment has been implemented in various corpora by now, whether adaptations will be needed for more complex cases will have to be seen in practice when the system is used in a wider range of corpora. And what the quality of the automatic alignment scripts is in practical cases is something that will have to be explored in the future.

Arofat Akhundjanova, Furkan Akkurt, Bermet Chontaeva, Soudabeh Eslami, and Cagri Coltekin. 2025. Parallel Universal Dependencies treebanks for Turkic languages. In *Proceedings of the Eighth Workshop on Universal Dependencies (UDW, SyntaxFest 2025)*, pages 129–136, Ljubljana, Slovenia. Association for Computational Linguistics.

Stefan Evert and Andrew Hardie. 2011. Twenty-first century corpus workbench: Updating a query architecture for the new millennium. In *Corpus Linguistics 2011*.

Christian Federmann, Tom Kocmi, and Ying Xin. 2022. NTREX-128 – news test references for MT evaluation of 128 languages. In *Proceedings of the First Workshop on Scaling Up Multilingual Evaluation*, pages 21–24, Online. Association for Computational Linguistics.

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

Maarten Janssen. 2018. TEITOK as a tool for dependency grammar. *Procesamiento del Lenguaje Natural*, 61:185–188.

Maarten Janssen, Piroska Lendvai, and Anna Jouravel. 2025. Alignment of historical manuscript transcriptions and translations. In *Proceedings of the 15th International Conference on Recent Advances in Natural Language Processing - Natural Language Processing in the Generative AI Era*, pages 462–470, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Tomáš Machálek. 2020. KonText: Advanced and flexible corpus query interface. In *Proceedings*

of the 12th Language Resources and Evaluation Conference, pages 7003–7008, Marseille, France. European Language Resources Association.

Robert Östling and Jörg Tiedemann. 2016. Efficient word alignment with Markov Chain Monte Carlo. *Prague Bulletin of Mathematical Linguistics*, 106:125–146.

Lucie Poláková, Martin Popel, Věra Kloudová, Michal Novák, Mariia Anisimova, and Jiří Balhar. 2025. Mitigating language barriers in education: Developing multilingual digital learning materials with machine translation. In *EDULEARN25 Proceedings*, pages 8754–8760, Valencia, Spain. IATED.

Alexandr Rosen, Michal Křen, Pavel Šmerk, and Michaela Svobodová. 2023. Intercorp: Multilingual parallel corpus with a uniform annotation. *Jazykovedný časopis*, 74(2):365–387. In English with Slovak abstract.

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.

Daniel Zeman, Martin Popel, Milan Straka, Jan Hajič, Joakim Nivre, Filip Ginter, Juhani Luotolahti, Sampo Pyysalo, Slav Petrov, Martin Potthast, Francis Tyers, Elena Badmaeva, Memduh Gokirmak, Anna Nedoluzhko, Silvie Cinková, Jan Hajič jr., Jaroslava Hlaváčová, Václava Kettnerová, Zdeňka Urešová, Jenna Kanerva, Stina Ojala, Anna Missilä, Christopher D. Manning, Sebastian Schuster, Siva Reddy, Dima Taji, Nizar Habash, Herman Leung, Marie-Catherine de Marneffe, Manuela Sanguinetti, Maria Simi, Hiroshi Kanayama, Valeria de Paiva, Kira Droganova, Héctor Martínez Alonso, Çağrı Çöltekin, Umut Sulubacak, Hans Uszkoreit, Vivien Macketanz, Aljoscha Burchardt, Kim Harris, Katrin Marheinecke, Georg Rehm, Tolga Kayadelen, Mohammed Attia, Ali Elkahky, Zhuoran Yu, Emily Pitler, Saran Lertpradit, Michael Mandl, Jesse Kirchner, Hector Fernandez Alcalde, Jana Strnadová, Esha Banerjee, Ruli Manurung, Antonio Stella, Atsuko Shimada, Sookyoung Kwak, Gustavo Mendonça, Tatiana Lando, Rattima Nitisaroj, and Josie Li. 2017. CoNLL 2017 shared task: Multilingual parsing from raw text to Universal Dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–19, Vancouver, Canada. Association for Computational Linguistics.