# Creating domain-specific translation memories for machine translation fine-tuning: the TRENCARD bilingual cardiology corpus

Gokhan Dogru

revista.tradumàtica
tecnologies de la traducció

Gokhan Dogru
Universitat Autònoma de Barcelona;
gokhan.dogru@uab.cat;
ORCID: 0000-0001-7141-2350

## Abstract

This article investigates how translation memories (TMs) can be created by translators or other language professionals in order to compile domain-specific parallel corpora, which can then be used in different scenarios, such as machine translation training and fine-tuning, TM leveraging, and/or large language model fine-tuning. The article introduces a semi-automatic TM preparation methodology that primarily leverages translation tools used by translators, in the interests of data quality and control by translators themselves. This semi-automatic methodology is then used to build a cardiology-based Turkish → English corpus from bilingual abstracts of Turkish cardiology journals. The resulting corpus, called TRENCARD Corpus, has approximately 800,000 source words and 50,000 sentences. Using this methodology, translators can build custom TMs in a reasonable time and use them in tasks requiring bilingual data.

Keywords: bilingual corpus preparation, translation memory, machine translation, TRENCARD corpus.

## Resumen

Este artículo investiga cómo las memorias de traducción (MT) pueden ser creadas por traductores y otros expertos lingüísticos a fin de compilar corpus paralelos específicos de un dominio, que luego pueden ser utilizados en varios escenarios, como el entrenamiento de la traducción automática y el ajuste de parámetros, la optimización de las MT y/o el ajuste de parámetros de grandes modelos de lenguaje. El artículo presenta una metodología semiautomática para la preparación de MT, que aprovecha principalmente herramientas de traducción utilizadas por traductores, en beneficio de la calidad y el control de los datos por parte de los traductores. Esta metodología semiautomática se utiliza para construir un corpus turco → inglés en el ámbio de la cardiología a partir de resúmenes bilingües de revistas turcas de cardiología. El corpus resultante, llamado Corpus TRENCARD, tiene aproximadamente 800.000 palabras de origen y 50.000 frases. Con esta metodología, los traductores pueden construir sus propias MT en un tiempo razonable y usarlas en tareas que requieran datos bilingües.

..

*Rebuda: 10 de octubre de 2022 / Acceptació: 3 de abril de 2024 / Publicació avançada: 20 de desembre de 2024*

Palabras clave: Preparación de corpus bilingües, memoria de traducción, traducción automática, corpus TRENCARD..

Resum

Aquest article investiga com els traductors i altres experts lingüístics poden crear memòries de traducció (MT) per tal de compilar corpus paral·lels específics d'un domini, que després poden ser utilitzats en diversos escenaris, com ara l'entrenament de la traducció automàtica i l'ajustament de paràmetres, l'optimització de les MT i/o l'ajustament de paràmetres de grans models de llenguatge. L'article presenta una metodologia semiautomàtica per a la preparació de MT, que aprofita principalment eines de traducció utilitzades per traductors, en benefici de la qualitat i el control de les dades per part dels traductors. Aquesta metodologia semiautomàtica s'utilitza per construir un corpus turc → anglès en l'àmbit de la cardiologia a partir de resums bilingües de revistes turques de cardiologia. El corpus resultant, anomenat Corpus TRENCARD, té aproximadament 800.000 paraules d'origen i 50.000 frases. Amb aquesta metodologia, els traductors poden construir les seves pròpies MT en un temps raonable i utilitzar-les en tasques que requereixin dades bilingües..

Paraules clau: Preparació de corpus bilingües, memòria de traducció, traducció automàtica, corpus TRENCARD.

## 1. Introduction

The rapid advancements in translation and localisation technologies pose today's professional translators a dilemma in terms of technology use. On the one hand, there is the technological deflation paradigm: the most advanced proprietary translation platforms integrate vendor management, project management, computer-assisted translation (CAT) tools and machine translation (MT) features, among other things, and provide a simple user interface that translators log into so as to work on translation segments without the need to control or master any individual technology involved. With web-based CAT tools such as Smartcat,[1] Crowdin[2] and Lokalise,[3] some of the leading technology companies integrate all these features on platforms and manage the technological complexity on behalf of the translators at the backend. This consolidation of features is so widespread that, as the European Language Industry Survey (2023) argues, "with the integration of technologies into suites that combine most if not all required functionalities, it may soon be futile to look at individual tools" (ELIS 2023, p. 40). On the other hand, there is the technological inflation paradigm: the same wave of technological advancements and the rapid growth of the translation and localisation industry have paved the way for the development of many new translation technologies, including a new generation of CAT tools, terminology tools, MT systems, customisation platforms, etc., in both proprietary and free and open-source versions. The 2023 Nimdzi Technology Atlas (Nimdzi, 2023) report lists 920 language technology tools used in the

[1] Smartcat. https://www.smartcat.com/ (last access: 21.02.2024)
[2] Crowdin. https://crowdin.com/ (last access: 21.02.2024)
[3] Lokalise. (last access: 21.02.2024)

translation and localisation industry. In line with the boom in new technologies, Rothwell & Svoboda (2019) report a notable increase in time devoted to teaching technology in translator training in their survey. While the technological deflation scenario seems to be more convenient for professional translators, as it apparently minimises the need to learn new technologies, it undermines the agency of translators (Moorkens, 2017), their ownership over translation data (Moorkens & Lewis, 2019b; Moorkens, 2022) and their control over the workflow, which, as a side effect, creates precarious working conditions for translators who work via large digital platforms on (in most cases) small chunks of translation projects without the guarantee of a constant flow of jobs or income, as highlighted by the concept of the "Uberization of translation" (Fırat, 2021). Although there is no deterministic reason for the technological deflation paradigm to aggravate the working conditions of translators, in practice it favours the owners of the platforms and accumulates economic power on their side, making translators overdependent on a few platforms. Acknowledgment of the technological inflation paradigm, on the other hand, opens up a plethora of technological possibilities for translators. Under this paradigm, while translators need to constantly learn new technologies, they can maximise their agency (Moorkens, 2017), be empowered as experts (O'Brien, 2012) and potentially obtain a bigger share of the efficiency gains provided by these technologies. And having more control over the technologies and workflows may give them more bargaining power over rates, among other things.

Within the abovementioned framework, this study focuses on domain-specific translation memory (TM) creation for MT fine-tuning. The current tendency is for translators to work with CAT tools into which MT systems are integrated (Farrell, 2022). They tend to use proprietary generic MT systems, such as Google Translate and DeepL, or custom MT systems provided by their clients, over which they have very limited control, if any. Since these MT systems are usually provided on the client side, and translators are expected to be their consumers, there tends to be an accompanying expectation of price discount, to the disadvantage of the translator (for a discussion of the effect of MT on pricing and productivity, see do Carmo, 2020). As an alternative to this scenario, we argue that translators are able to avoid this route by using free and open-source tools and by getting involved in different stages of MT system creation. This includes not only preparing parallel corpora in the form of TMs, but also evaluating the quality of training data, fine-tuning pre-trained neural machine translation (NMT) models, evaluating the quality of translation output, and deploying their own MT systems in their CAT tools. For instance, free and open-source pre-trained NMT engines based on the Marian NMT[4] system, and offered by the OPUS-MT project (Tiedemann & Thottingal, 2020), can be used in a desktop environment through the free, open-source OPUS-CAT[5] (Nieminen, 2021) software program that runs on Windows, thus allowing translators to connect MT engines to their CAT tools. This software also offers scope for the local fine-tuning of these pre-trained NMT engines through the addition of custom translator

[4] Marian NMT. https://marian-nmt.github.io/ (last access: 23.02.2024)
[5] Opus CAT. https://helsinki-nlp.github.io/OPUS-CAT/ (last access: 23.02.2024)

data in the form of TMs. This makes the software even more useful, potentially, for translators as it allows them to customise MT engines with their own data. It can often be the case that TMs owned by translators are not big enough for fine-tuning the MT engines and, consequently, translators cannot benefit from the potential productivity gains that MT fine-tuning offers. Mikhailov (2022:224) observes that aligned corpora are still not sufficiently common, are concentrated around a few language pairs and are available mostly as general domain data rather than specific domain data. Hence, in the spirit of empowering translators in a technological inflation paradigm, we have developed a semi-automatic TM creation procedure using a selection of tools familiar to translators.

It is important to note that translators who typically work with CAT tools may not be familiar with programming languages (such as Python) or other command-line programs commonly used for web crawling and automatic bilingual corpus preparation. Hence, while the procedure for TM data compilation presented here is not as advanced as methodologies using fully automatic web crawlers, it nevertheless allows translators to create TMs with a small amount of high quality, domain-specific data to be used in an MT fine-tuning procedure. In order to create very large monolingual, bilingual or multilingual corpora from web content, translators should ultimately resort to fully automatic web crawlers, as detailed in section 3, where different available tools are described. In this respect, it is important to note that the creator of OPUS-CAT, Nieminen (2021:291), highlights that fine-tuning with even a relatively small corpus of 10,000 sentences may provide improvements in quality, as opposed to training an MT completely from scratch, which may require parallel corpora comprising in excess of millions of sentences (Pérez-Ortiz et al., 2022:148). In a study on the productivity gains of NMT fine-tuning in the finance domain, Läubli et al. (2019) found that translators worked 59.74% faster in the German-French language pair and 9.26% faster in the German-Italian pair. Additionally, Gilbert (2020) fine-tuned Google AutoML with 1,367 sentence pairs and observed that even with such a small amount of fine-tuning data, the system began to learn from the style of the translator. In a comparative quality evaluation study on MT fine-tuning in the English-Turkish, English-Spanish and English-Catalan pairs in the localisation domain, Dogru & Moorkens (2024) observed that human reviewers rated the fine-tuned engines more highly, across the three language pairs, in ranking, adequacy and fluency tasks. As Nieminen (2021:290) points out, domain fine-tuning strategies have been implemented at least since Koehn & Schroeder (2007), and numerous studies have shown different levels of quality gains.

## 2. Previous work

TMs have been used in the translation industry since the 1980s, and the emergence of corpus-based MT created “an unanticipated connection” between TMs and this MT approach (Melby & Wrigh, 2015:675). This connection attributed a new role to TMs as “a type of parallel corpora” (Zanettin, 2012:169) that can be used for MT training and fine-tuning. The possibility of creating TMs by aligning previously translated documents

on a sentence level to provide parallel corpora for MT training and fine-tuning opened up a new horizon for translators working in a CAT tool environment.

In the context of translation studies, the use of corpora in translation research and practice can be traced back to the 1990s and early 2000s (Baker, 1993; Aston, 1999; Bowker & Pearson, 2002). Those early works, as well as subsequent ones by, for instance, Sánchez-Gijón (2009) and Marco & von Lawick (2009), focused on using corpora to support the development of translation competence and for research. Do-it-yourself (DIY) corpus procedures (Sánchez-Gijón, 2009) were also suggested in the abovementioned works. Yet corpus preparation for MT training and/or fine-tuning has not been commonly addressed in translation studies and has mostly been performed by computer scientists.

One of the earliest large-scale projects to create parallel corpora for MT was the Europarl Project, based on the proceedings of the European Parliament from the official website in 11 languages (Koehn, 2005). Later, the Opus Corpus repository (Tiedeman & Nygaard, 2004) introduced and compiled more language pairs and toolkits that can be used for different purposes, including MT training and fine-tuning. Moreover, several EU-funded large-scale projects contributed to greater access to parallel corpora in different languages: Paracrawl (Esplà-Gomis et al., 2019) for all official EU languages; the EuroPat corpus (Heafield et al., 2022) for six official European languages (German, Spanish, French, Croatian, Norwegian and Polish) paired with English; MaCoCu (Bañón et al., 2022) for 11 low-resourced European languages; and Gourmet[6] for global under-resourced languages in the media domain. While these projects have helped solve the data quantity problem for many languages, there is still a need for high quality, domain-specific corpora in most languages, especially for the purpose of MT fine-tuning/domain adaption. For example, while it is possible to acquire the English-Spanish parallel corpus of the European Medical Agency (medical domain) or that of the United Nations (politics domain) through Opus Corpus, these corpora are not available for the English-Turkish language pair. Hence, there is still a need for corpus creation on different scales and in different domains.

One group of users who may be interested in TM preparation is the community of translators, who can choose to deploy and adapt their own MT systems, whether through OPUS-CAT in their local Windows environment or through commercial platforms such as ModernMT[7] and Google AutoML Translation.[8] Translators can use the base MT engine of the relevant provider directly or fine-tune the engine by uploading their own TMs. If TMs are the product of their own translations, the size may be too small to effectively improve MT output to the extent that it would result in a significant productivity increase after fine-tuning (experimentation may be needed to decide upon sufficient corpus size, depending on the domain and language pair). Hence, another approach open to translators is to create their own TMs out of similar documents and/or websites.

[6] Global Under-Resourced Media Translation. https://gourmet-project.eu/ (last access: 23.02.2024)
[7] ModernMT. https://www.modernmt.com/ (last access: 23.02.2024)
[8] Google AutoML Translation. https://cloud.google.com/translate/automl/docs (last access: 23.02.2024)

One important concern in the literature has been the ownership of translation data (Moorkens, 2017 and 2022; Moorkens & Lewis, 2019a and 2019b). Translators already create TMs as a byproduct of their translation work. However, as Moorkens & Lewis (2019b) observe, the ownership of this translation data is not clearly defined in many jurisdictions and, in practice, translators transfer their ownership to their clients at the outset of their collaborations to provide their services. This transfer usually means letting the client use such data not only for TM leveraging but also for all other uses, including MT training and fine-tuning. This unsustainable practice exemplifies the technological deflation paradigm we outlined above. Moorkens & Lewis, (2019b) also question the sustainability of this approach and suggest treating translation data as a shared knowledge resource. On the other hand, when it comes to acquiring digital content from the Web for MT fine-tuning or for other purposes, it should be taken into consideration that this act may constitute an infringement of copyrighted material. Websites usually include copyright warnings or licences detailing the permitted use cases of their content and they may have no-robot policies to stop crawling. An anonymisation toolkit, such as the one created by the MAPA Project,[9] may be needed to solve this problem. With the advent of data-driven artificial intelligence systems, such as ChatGPT, which depend on a huge amount of internet data, concerns about data use permissions continue to be raised and jurisdictions such as the EU are preparing laws to regulate these uses.

The following section outlines a semi-automatic procedure for compiling a domain-specific bilingual corpus from the Web which can later be used for fine-tuning an MT engine, among other use cases.

## 3. Methodology for TM creation

Translators work with online and offline CAT tools with varying degrees of complexity, and permissions to implement TMs, glossaries, and MT integrations. When they offer translation services coupled with MT (e.g. post-editing) or are allowed to use the MT engine of their choice, they may improve the quality of the MT results by fine-tuning the engine through their TM resources.

While the underlying toolkit for corpus creation from the Web has changed since the 2000s and new tools have been introduced, such as Bitextor and Bicleaner (Esplà-Gomis et al., 2019), the overall automatic procedure remains very similar today: resource selection, web crawling, document alignment, sentence alignment, and file-type conversions of the final corpus (see, for example, Koehn, 2005). Based on the abovementioned studies, we outline the end-to-end process of TM creation in Figure 1. This workflow can be completely manual, semi-automatic or fully automatic. In our suggested procedure, we have a semi-automatic, translator-friendly design. The semi-automatic approach allows for the revision of the automatic sentence alignments and ensures their overall quality, although the resulting corpus size tends to be smaller compared to fully automatic approaches.

[9] MAPA project. https://mapa-project.eu/ (last access: 12.12.2023)

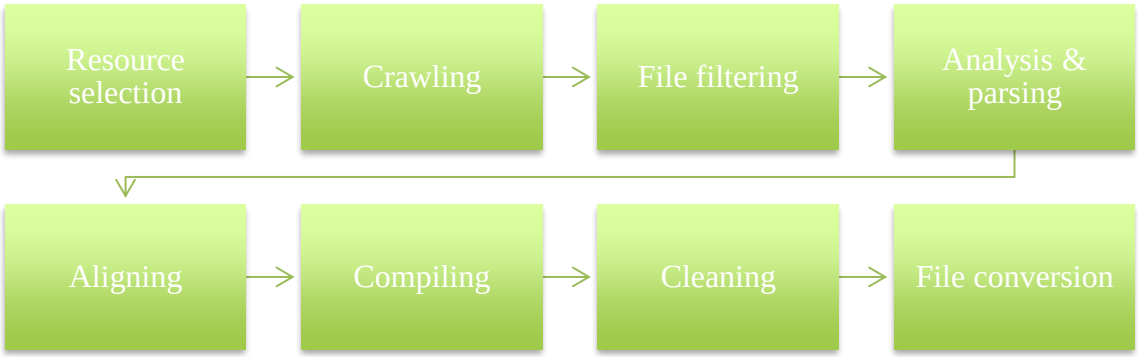


*Figure 1. Pipeline for TM creation.*

Below, we will explain each step in the pipeline, the tools used and the reasons for using each tool. In order to better illustrate the process, we have used this procedure to create a Turkish-English parallel corpus in the form of a TM for the cardiology domain from bilingual abstracts from four Turkish cardiology journals. The practical example is described in detail in section 4.

### *3.1. Semi-automatic corpus preparation pipeline*

The procedure set out below describes a process that begins with bilingual resource selection from the Web, downloading web pages in different file formats with code pieces. A translator may skip some of these steps if they already have enough parallel documents in editable formats, such as DOCX, XLXS, ODF, etc. Furthermore, some tools may allow for corpus cleaning before compiling or after file conversion; hence, the order of some of the steps may change slightly. In a nutshell, as mentioned above, some steps, such as resource selection, corpus cleaning, etc., can be ignored or swapped around, depending on the available resources and toolkits, as long as a final bilingual corpus is created.

#### *3.1.1. Resource selection*

Resources for TM creation may vary from client-provided source and target files in different formats to bilingual physical publications and Web content. Clients may not always provide sufficiently large bilingual files, and digitalising physical publications may be too time consuming. Although one should be careful about data ownership permissions and data reliability, the Web remains a popular alternative for data collection and is also the usual source for parallel corpus acquisition. In his book on translation-focused corpora, Zanettin (2012:56) highlights that: "The World Wide Web is the largest content distribution system and the most extensive and accessible repository of textual data." Using the Web, translators can search for bilingual websites with enough parallel content in their specific domain and opt to use it to create parallel corpora in the form of TMs. The quality and quantity of this content plays a crucial role. The quality aspect includes not only translation quality but also the alignability of the content. Regarding quantity,

in general, the more specific the domain is, the smaller the possibility of acquiring large quantities of domain-specific data will be. Hence, it can be ideal to have the largest possible domain-specific parallel corpora available.

One approach to resource selection may be to consult the official websites of international organisations and governmental bodies to check whether they have content in the specific language pair involved. For example, the United Nations' Turkey website includes the UN Sustainable Development Goals in English[10] and Turkish.[11] This website can provide some parallel corpora for fine-tuning an engine for the legal/political domain. Professional associations, such as those for engineering, architecture, medicine, etc., usually have terminologically dense multilingual content. Zanettin (2012) provides a systematic overview of sources for translation-driven corpora. Le Bruyn et al. (2022) approach the subject from a more technical perspective, focusing on the representativeness of the corpus while collecting sources. Finally, the selection phase may be facilitated by the client giving the translator permission to freely use the content of their website. Alternatively, the client may provide bilingual files directly. The expertise of the translator in the relevant domain will be of great benefit where the selection and curation of files are concerned.

Finally, when selecting resources from websites, data use permissions should be considered. Some websites have Creative Commons[12] symbols detailing the type of licence they provide and whether there are any limitations on the use of data.

### *3.1.2. Crawling*

Crawling is the process of downloading content (textual, audiovisual and other types of content) from websites using a particular code or program. Crawlers, or spiders, can download a whole top-level domain, a list of URLs, or content from a specific link. Users can set parameters to limit the time spent on the task and the amount of data to download and/or the type of files to be downloaded. HTTrack Website Copier[13] is a multiplatform, free software tool very commonly used for this process. It has a graphical user interface (GUI), and one can download a website by simply pasting its link inside a search field in the software. It is possible to limit file sizes and types. Unless there is a no-robot policy or limitation on the website, the download process will begin to download all the content of the website. Another common tool that is easy to use is WGET.[14] This is a command-line tool that runs on the Linux operating system (OS). Most translators work on Windows, but they can create a virtual machine, using Oracle VM VirtualBox, to run a free Linux OS[15] inside their Windows OS and, thus, run WGET. WGET can be configured to download parts of or whole websites,

[10] https://turkiye.un.org/en/sdgs/1 (last access: 12.12.2023)
[11] https://turkiye.un.org/tr/sdgs/1 (last access: 12.12.2023)
[12] https://creativecommons.org/share-your-work/cclicenses/ (last access: 12.12.2023)
[13] HTTrack Website Copier. https://www.httrack.com/ (last access: 23.02.2024)
[14] WGET. https://www.gnu.org/software/wget/ (last access: 23.02.2024)
[15] VirtualBox. (last access: 23.02.2024)

much like HTTrack. Usually, one line of code is sufficient to trigger the download of a website. In some cases, WGET may prove to be more useful than HTTrack Website Copier, depending on how the final downloaded files are displayed (see our experiment in section 4). Once the Linux Ubuntu Terminal is initiated, a code snippet such as the one in the figure below will be enough to start the download.

```
wget --mirror -p --convert-links --content-disposition --trust-server-names -P corpus
    http://khd.tkd.org.tr/
```

*Figure 2. The script for downloading content from a website.*

While HTTrack may change file names, depending on the server settings on the website, WGET tends to keep the file names as they are, which is important when aligning web content. Once the download process is completed, there will be lots of unnecessary non-textual or non-bilingual files downloaded. The translator needs to select the source and target documents to be aligned and remove all "noisy" files (if they have not done so already when configuring the crawling settings). At this point, the translator can enter their folder and manually remove files such as images, configurations files, or types of files with no content. If the operation of copying the files with textual content and pasting them into another folder requires less time than removing all non-textual files, the former may be a better option. For example, if there are 100 HTML files with textual content and 3,000 files with images, configurations files or other non-textual content, copying the 100 HTML files and pasting them into another folder would be more advisable. Depending on the expected size of the fine-tuning corpora and the file structure of the website, noise removal can be carried out quickly.

### *3.1.3. File filtering*

File filtering is a manual process of finding relevant documents that can be aligned. In this manual filtering process, the downloaded folder structure and file names play a crucial role, especially, when there are many files. Once there are an equal number of source and target documents with consistent names, the analysis and parsing processes can begin.

### *3.1.4. Analysis and parsing*

Parsing is the operation of separating textual strings from code in the context of translation. Documents downloaded from the Web come in their web format and not as plain texts. The usual file formats are HTML, XML, PHP, etc. Hence, depending on the downloaded file type and the portion of the file to be aligned, translators can either use default file filters in their CAT tools, such as OmegaT, memoQ, Trados Studio, etc., or create custom filters using the filtering function of their CAT tools. To decide what type of filter/parser will be needed, translators should open the files in a text editor, such as NotePad++, which is a free option (see, for example, Figure 4

in section 4.1.1). Once patterns are found in the analysis, a parsing strategy can be developed, which may include the use of some regular expressions (regex). The screenshot and regex text filter examples in section 4.1.1 show how these operations are conducted. There may also be ready-made external filters, such as the Okapi Filters Plugin for OmegaT. This plugin can extend the capabilities of OmegaT to filter out advanced file types, such as HTML and JSON. While large language service companies usually task localisation engineers with file processing, individual translators can also become familiar with filtering translatables from non-translatables, and CAT tools that provide a GUI for simple filter customisation, such as memoQ, let translators perform this operation directly within a CAT environment.

### *3.1.5. Aligning*

Alignment is a key operation for translators to leverage previous bilingual files that are not in a TM format. Kraif (2002) provides the following definition of alignment:

> Aligning consists in finding correspondences, in bilingual parallel corpora, between textual segments that are translation equivalents. (p. 275)

Source segments can be paired with target segments automatically. To pair source and target documents sentence by sentence, the alignment software uses certain source and target sentence parameters, such as paragraph order, punctuation marks, inline tags, formatting, reference bilingual terminology, and relative word counts. One common command-line tool used for this operation is HunAlign,[16] which automatically aligns sentences and provides a confidence score based on the parameters. Fully automatic corpus preparation pipelines use different versions of HunAlign to create aligned documents, since it is rather flexible. However, translators can use the alignment feature of their CAT tool. The free, open-source OmegaT has an alignment feature that can quickly align a pair of documents into a TMX[17] file, but it cannot align multiple files in a single step. MemoQ's LiveDocs feature allows the alignment of multiple files and also has an interface for editing misalignments, which is useful for improving alignment results. It is possible to adjust alignment parameters and establish a reference bilingual terminology to help the alignment process. LiveDocs also supports custom file filters before the start of alignment. Trados Studio, which is also a common CAT tool among translators, includes a similar feature with similar capabilities. In general, the crucial features for an alignment component in a CAT tool are configurable parameters, an easy-to-use alignment editor, and support for multilingual file uploads.

[16] HunAlign sentence aligner. https://github.com/danielvarga/hunalign (last access: 23.02.2024)
[17] Translation Memory eXchange, a standard format for TM data exchange.

### 3.1.6. Compiling

In the context of the methodology presented here, compiling means combining all individually aligned file pairs into a single file. Once the corpus is aligned, the next step is to perform this compiling operation. Doing so is optional, as sometimes it may be useful to keep the files separate to allow for a fine-grained analysis of results in each of them. If our choice is to compile all files into a single one, the use of the TMX format allows for interoperability between different CAT tools. Moreover, MT platforms such as ModernMT, KantanMT[18] and OPUS-CAT MT support TMX as an input file format for training and/or fine-tuning. Therefore, compiling aligned files into a single TMX file can be useful. MemoQ's LiveDocs feature allows for the importation of the content of all aligned files into a single TMX file in one operation. As an extra step, once a TMX is created, it is possible to easily obtain the source side and the target side as two separate plain text files through Okapi Rainbow.[19] Once a single file has been obtained, a cleaning process can be initiated in order to remove noise from the TMX file.

### 3.1.7. Cleaning

The quality of training data determines the resulting quality of the MT engine. For this reason, the training data (in the form of a TM) needs to be cleaned of any noisy data, including inline tags, non-textual characters, misaligned segments, duplicate sentences, incorrect encodings and very long sentences. Quality assurance features of CAT tools can help detect such problems, but there are also standalone tools, such as Heartsome TMX Editor and Goldpan TMX/TBX Editor, and Okapi Framework tools, such as CheckMate and Olifant, which can be used to detect and remove this noisy content. Depending on the quality of the raw TMX file, this manual step can take some time, but it can improve the quality of NMT engines since NMT is quite sensitive to the quality of training data.

### 3.1.8. File conversion

The necessity of this last step depends on the compiled file format. As mentioned above, the expected end product of TM creation for translators is a TMX file, but data used for training an MT system requires the use of a different format. Okapi Rainbow has a file conversion utility. A TMX file can be divided into two aligned plain text files that can later be uploaded for MT training or fine-tuning. Resulting plain text files can also be used for other operations, such as monolingual term extraction or other text mining operations.

[18] KantanMT. https://www.kantanai.io/ (last access: 23.02.2024)
[19] Okapi Rainbow. https://okapiframework.org/wiki/index.php?title=Rainbow (last access: 23.02.2024)

## 4. A practical example of TM preparation: the TRENCARD Corpus

Using the procedure detailed in section 3, we compiled the TRENCARD Corpus,[20] a Turkish-English TM for research and experimentation purposes. Below, we explain our procedure, step by step.

To begin with, we selected four cardiology journals that publish Turkish and English abstracts together with the scientific articles on their websites: *Archives of the Turkish Society of Cardiology,* the *Turkish Journal of Cardiovascular Nursing,* the *Turkiye Klinikleri Journal of Cardiology* and the *Turkish Journal of Thoracic and Cardiovascular Surgery.* Abstracts in these academic cardiology journals were chosen based on achievable possible TM size, reliability of translation, existence of structured data for crawling, and terminological density. An important consideration while compiling the TM from the Web was related to licences. The journal abstracts from *Archives of the Turkish Society of Cardiology,* the *Turkish Journal of Cardiovascular Nursing* and the *Turkiye Klinikleri Journal of Cardiology* are under a restrictive Attribution-NonCommercial 4.0 International (CC BY-NC 4.0) licence,[21] while the abstracts from the *Turkish Journal of Thoracic and Cardiovascular Surgery* have a more permissive Attribution-NonCommercial 4.0 International licence,[22] which grants permission to "redistribute, remix, transform, and build upon the material" without a commercial purpose. For this reason, although we explain our TM preparation methodology for the abstracts from the four journals, we only redistributed the TM produced from those of the fourth journal in GitHub. The TM in question has a size of more than 150,000 source words and approximately 14,000 sentences. For other details of this TM, refer to section 4.4. By replicating our methodology, as explained below, a comparable TM can be obtained from the abstracts of the remaining three journals subject to the licence limitations.

Abstracts in these Turkish medical journals are generally bilingual, in Turkish and English. The remaining parts of the journals are mostly in Turkish. For this reason, and due to licence limitations, we chose to build the TRENCARD Corpus from abstracts. However, abstracts typically contain 100-300 words and are short texts. Therefore, in order to build a TM of a relevant size, we needed to compile a large number of abstracts. An advantage of using abstracts is that their content is typically very structured, which can help in parsing and alignment. A cardiology abstract starts with a section called "background" or "objective", then includes "methods", "results", "conclusion" and "keywords". Since word count is limited in abstracts, more terminology is used to express the content in a restricted number of words, which leads to terminological density. Considering that these four scientific journals are authoritative in their fields, it is convenient that they are published openly and that translations of the abstracts are of high quality (since they undergo a peer review process) rather than the product of

[20] TRENCARD Corpus. https://github.com/gokhandogru/trencard

[21] Attribution-NonCommercial-NoDerivs 2.0 Generic. https://creativecommons.org/licenses/by-nc-nd/2.0/ (last access: 23.02.2024)

[22] Attribution-NonCommercial 4.0 International. https://creativecommons.org/licenses/by-nc/4.0/ (last access: 23.02.2024)

volunteer or crowd translations. In combination, these four cardiology journals reflect the last 30 years of cardiology study in Turkey, which makes the TM representative of the Turkish cardiology domain. The oldest online issues are from 1990. The content of the English and Turkish abstracts represents the common terminology used in this field.

### *4.1. TM preparation with abstracts from Archives of the Turkish Society of Cardiology*

*Archives of the Turkish Society of Cardiology* "is a peer-reviewed journal that covers all aspects of cardiovascular medicine", is published in Turkish and English, and "accepts papers on a wide range of topics, including coronary artery disease, valve diseases, arrhythmias, heart failure, hypertension, congenital heart diseases, cardiovascular surgery, basic science and imaging techniques" (*Archives of the Turkish Society of Cardiology,* n.d.). The journal is published by the Turkish Society of Cardiology, which "is the scientific, nonprofit, nongovernmental organization of Turkish cardiologists, established on May 21st, 1963. Its 2360 members cover almost all the academicians and practitioners of cardiology and the related specialists in Turkey."[23] The online archive of the journal covers the period 1990-2024. As of 2005, each volume includes eight issues and a varying number of supplements, which amount to more than 152 issues and 52 supplements up to the current issue, 52 (1), from 2024. Issues have an editorial section, original articles, case reports, case images and a section called Perspectives.

The website of the journal is in Turkish and English, and access to the content of the articles is open. Furthermore, the website has a well-structured HTML that allows for easy processing of its pages. As explained below, we firstly used HTTrack Website Copier to download the content of the website. However, the file names of the downloaded HTML pages were not displayed in an ordered way; hence, we chose to use WGET instead. Below, we explain our experience with these two website downloaders.

Using the main URL of the website (https://www.archivestsc.com/), the entire website was downloaded to our local computer by the default workflow of HTTrack Website Copier. All items inside the website were downloaded, including HTML files, style sheets, images and PDFs. Since folder structure was maintained, all English and Turkish abstracts were saved to separate folders. All unnecessary files were deleted, leaving only the HTML pages including each abstract (there was one abstract in each HTML page). This resulted in 390 pages for each language. However, we observed that file names for each page were changed in a way that made it hard, if not impossible, to align files. The files' names, such as "jvi0a30" and "jvi0aa5", did not appear to have a logical basis. Normally, on the website, each HTML page is hosted under a URL structure, such as "website name + /jvi.aspx?un=TKDA-72699". Here, the number after the "TKDA" corresponds to the unique identity of each page. As we were not able to align the files properly in this setting, we searched for another website crawler and used WGET. WGET works on the Ubuntu operating system. Since we were working in a Windows OS, we downloaded

[23] Turkish Society of Cardiology. https://tkd.org.tr/en/menu/10/history (last access: 23.02.2024)

Oracle VirtualBox to be able to work on Ubuntu. After opening the terminal in Ubuntu, we typed this code to download all the website again:

```
wget --mirror -p --convert-links --content-disposition --trust-server-names -P TurkishCard
    https://www.archivestsc.com/
```

*Figure 3. Script for downloading content from the website.*

The name "TurkishCard" was given to the folder that would include all the files and subfolders. Similarly to when we used HTTrack Website Copier, all content was downloaded, including JavaScript files, style sheets, images and other elements, and, as before, folder structure was maintained. Yet, unlike with the previous download, the file names were kept as they were on the website. After the completion of the download, the folder was moved back to Windows and all unnecessary files were again removed, leaving only the files with abstracts. Each page including a Turkish abstract had a name such as "jvi.aspx_pdir=tkd&plng=tur&un=TKDA-00090", in which the section "jvi.aspx_pdir=tkd&plng=tur&un=" was the same for all the Turkish abstracts. For the English abstracts, the structure of the names was "jvi.aspx_pdir=tkd&plng=eng&un=TKDA-24582", where "jvi.aspx_pdir=tkd&plng=eng&un=" was standard in all abstracts. This consistent pattern allowed us to use batch tasks. Windows 10 has an advanced command-line terminal called Power Shell[24] through which it is possible to batch rename files.[25] In order to simplify the handling of the files and match Turkish and English files, the Power Shell terminal for renaming was implemented by using the following command:

```
PS D:\Academia\2019 - 2020 Thesis Completion Phase\PhD\Chapter
    7_Methodology\Ready Corpora 2019\1. TKDA Journal\v2_TKDA
    Journal\www.archivestsc.com\uzun en> Dir | Rename-Item -NewName {$_.name -
    replace "jvi.aspx_pdir=tkd&plng=tur&un=","tr-"}
```

Through this command, we replaced "jvi.aspx_pdir=tkd&plng=tur&un=" with "tr-". Hence, the file names were abbreviated, giving short names such as "tr-TKDA-0900", which were easier to handle. The same process was repeated for the English abstracts. We observed that abstracts were repeated twice in both folders. Duplicates were removed. The total sum of abstracts was around 3,920 in each language. There was a difference of eight abstracts between the Turkish and English abstract folders. Since finding the non-matching files would have been a time-consuming task, we temporarily changed the names of the Turkish abstracts from "tr-" to "en-", so that they became equal to the English names, and software called AllDup 4.2 was used to identify and delete the non-matching files. After finding and deleting the non-matching files, we renamed the Turkish abstracts again.

[24] Windows Power Shell. https://docs.microsoft.com/en-us/powershell/scripting/getting-started/getting-started-with-windows-powershell?view=powershell-7 (last access: 23.02.2024)

[25] If the number of files is low and the translator wishes to avoid using the command line, the files can be matched or aligned manually. However, this process may take longer if there are many files to be aligned/matched.

Finally, we had 3,918 files for each language; in other words, 3,918 abstracts for Turkish and 3,918 for English.

After the file selection and alignment steps, sentence-level alignment was initiated. This process started with analysis and a decision on the most efficient strategy for parsing the files, since we only needed the abstracts on each page. Manually copying and pasting each abstract into a plain text file is time consuming, whereas a strategy for parsing each file and then aligning the files at sentence level provides the possibility of saving time. The free, open-source CAT tool OmegaT has an alignment feature, but it does not make it possible to batch process files and parse a certain part of a file. HunAlign, meanwhile, requires advanced technical skills for achieving our goals. For these reasons, we opted for memoQ's LiveDocs feature, in which it was possible both to customise the file parsing filters and include only a certain part of the file, and, thereafter, align the files directly. Once the automatic alignment was completed, an editor window was opened for editing the misaligned sentences in memoQ. Finally, when all sentence alignments were confirmed, all the sentences could be imported into a TM. In brief, our subsequent steps were as follows:

1. Open memoQ's LiveDocs feature and create a new corpus
2. Add alignment pairs to the new corpus (all files in the Turkish folder on one side and all files in the English folder on the other side)
3. Create a parsing filter for the Turkish side and one for the English side
4. Select the correct language encoding
5. Start the filtering and aligning process
6. Check the alignment editor for any mismatches
7. Import all sentences into a TM
8. Export this TM in TMX format

As mentioned in section 3.1, the order of the steps involved in TM preparation may change; since we used memoQ, the alignment and parsing steps were swapped. When an HTML file is imported into memoQ, the program uses its default HTML filter.[26]

### *4.1.1. Parsing the Turkish abstracts*

All text content (menu items, website-related general texts, etc.) is imported with the default HTML filter. However, this can be changed, and either other default filters can be applied or custom filters can be created, using a regex text filter to extract only a specific part of a file. Moreover, more than one filter can be used to filter content from the code selectively. These consecutive filters are called cascading filters. Our cascading filter had a regex text filter (to extract only the relevant content) and an HTML text filter (to correctly visualise inline HTML tags in the file). In order to configure the filter for extracting the English and Turkish abstracts, we analysed a few HTML files from our corpus. Note that for this filter to work on 3,918 files, they must all have the same structural pattern. Figure 4 shows the HTML structure of a file.

[26] We use the term "filter" with the same meaning as "parser".


*Figure 4. HTML structure of one of the files including a Turkish abstract, as displayed in Notepad++ Editor. The grey area is the area to be extracted and the rest is not imported.*

Our regex filter was able to filter the relevant text from the file. All abstracts had the same title structure between "<h2 class='journalArticleinTitletur'>" and "</h2>" tags. Secondly, content between "<br><p>" and "<hr noshade size=4 align=center color=#d3d3d3>" was the content to extracted. In other words, we wanted to filter the content of the abstract up to the end of the keywords. Consequently, we had two regex rules for our first filter, which were as follows:

**Import only the following content:**

1- <h2 class='journalArticleinTitletur'>.*</h2> (Import everything between these two tags)

2- <br><p>.*<hr noshade size=4 align=center color=#d3d3d3> (Import everything between these two tags)

*Figure 5. Filter configuration.*

The ".*" regex symbols mean "every character". As can be observed in Figure 2, there were some inline tags remaining among the extracted content perceived as plain text, such as "<br>METOD</br>". These characters affect sentence segmentation during alignment and may also affect MT quality when used in training and/or fine-tuning. Hence, they had to be removed in the subsequent steps. To facilitate this process, we applied a second filter (HTML filter) to the extracted content to recognise these characters as HTML tags. It was then possible to remove the HTML tags automatically using a tool with an automatic tag removal feature. The last important consideration in this step was the selection of the correct encoding for the language. Windows encoding (Windows-

1254) provided the correct character set in this scenario.[27] And after the application of the cascading filter (regex text filter + HTML filter), we ended up with a segmented, clean Turkish abstract. Figure 6 shows how the filtered Turkish abstract and English abstract look in the LiveDocs Alignment Window.

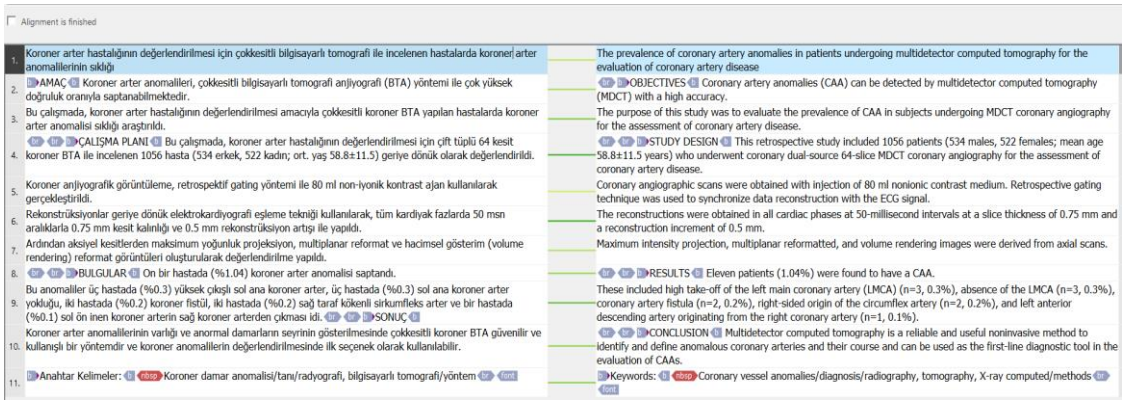


*Figure 6. Alignment of source and target files in memoQ. The items coloured purple and red are HTML tags.*

### *4.1.2. Parsing the English abstracts*

We followed a similar procedure for the English abstracts. Again, a cascading filter configuration was used for parsing. Only the first rule of the regex filter was slightly different:

**Import only the following content:**

1- <h2 class='journalArticleinTitleeng'>.*</h2> (Import everything between these two tags)

2- <br><p>.*<hr noshade size=4 align=center color=#d3d3d3> (Import everything between these two tags)

*Figure 7. Filter configuration in Memoq.*

Note that above, only the "<h2 class='journalArticleinTitleeng'>" tag is different in the filter. Another difference was the encoding. We selected Western European (Windows) encoding for the English files. Once the parsing filters were set for both languages, we initiated the alignment step. Figure 6 shows how two files are aligned in memoQ. In this case, there were no misalignments. However, there may be misalignments that need to be edited in other cases. We conducted some tests to make sure this parsing and alignment methodology can be used for all the files. The tests involved importing 10 file pairs into memoQ, applying the filters, carrying out alignment, and then checking whether there were misalignments. Once we had made sure that the majority of the segments were correctly aligned, batch processing for aligning all the 3,918 Turkish files with the 3,918 English files was conducted using the corresponding filters. The process was completed with 27,279 aligned sentences (segments).

[27] We discovered this after a few trials with different character sets, such UTF-8.

All those sentences were compiled in a TM and exported as a TMX file. In order to remove any tags, special characters, or any other kind of noise, we further processed the TM using cleaning tools. Once we had combined all the content of the 3,918 Turkish abstracts and the 3,918 English abstracts within a single TMX file, we could start the cleaning step. TM maintenance tools, such as Heartsome TMX Editor and Goldpan TMX/TBX Editor, and Okapi Framework tools, such as CheckMate and Rainbow, are useful for implementing batch cleaning operations like removing inline HTML tags, inconsistent numbers, and duplicate sentences. Essentially, this process is similar to translation quality assurance performed by professional translators. Figure 8 shows all the checks implemented by Goldpan TMX/TBX Editor.

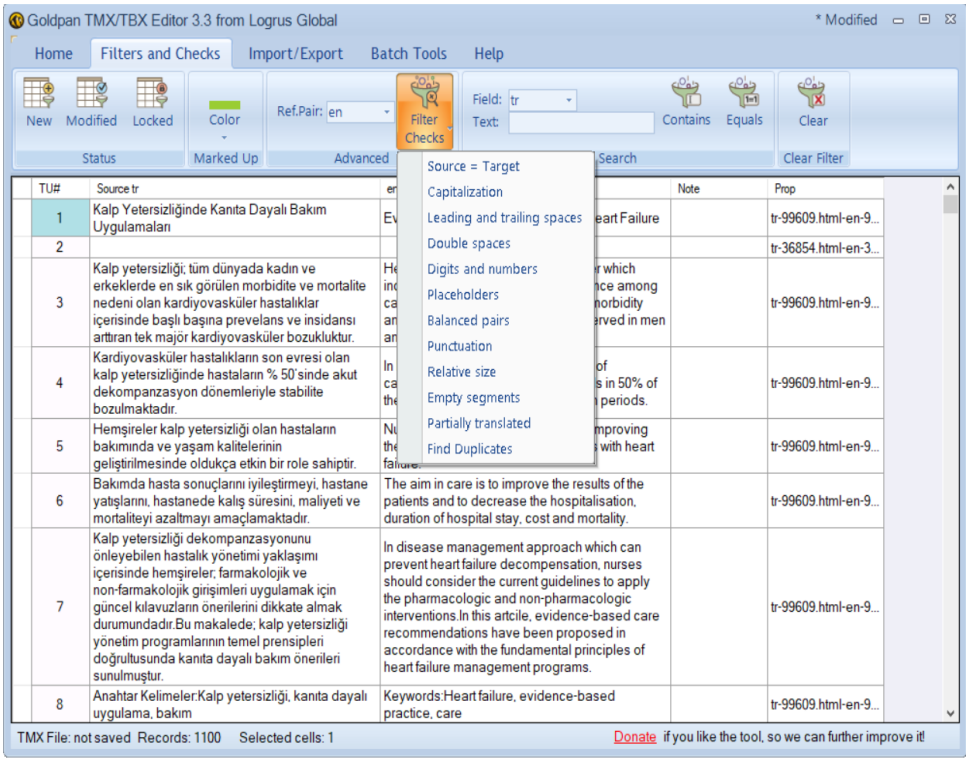


*Figure 8. Goldpan TMX/TBX Editor has filter checks that allow 12 different checks to remove inconsistencies. Moreover, the Home tab includes a function for cleaning the tags in different formats. These options were used to optimise our TM.*

We only used Goldpan to clean our TM. When we completed cleaning, we ended up with a corpus in the form of a TMX file. Since MT platforms such as KantanMT, OPUS-CAT and MutNMT[28] support the importation of TMX files, we did not need to further process our file. In some MT tools, the files would need to be prepared as two aligned plain text files. For example, older versions of MTradumàtica[29] require two distinct corpus files. The Rainbow tool can be used for this operation. It is worth noting that MT toolkits or systems such as KantanMT include internal operations for cleaning a TM automatically, in a similar way to the cleaning procedures mentioned above.

[28] MutNMT. https://ntradumatica.uab.cat/ (last access: 23.02.2024)
[29] MTradumàtica. https://mtradumatica.uab.cat/ (last access: 23.02.2024)

After importing the TMX file into Rainbow, going to Utilities > Conversion Utilities > File Format Conversion > Parallel Corpus Files and executing the command creates an aligned Turkish plain text file and an aligned English plain text file with the desired encoding. Although not necessary for training/fine-tuning purposes, we conducted this step to be able to analyse the files when required. In the table below, we summarise the steps taken and the tools used in preparing our first journal's TM.

| | Phase | Tools |
|---|---|---|
| 1. | Crawling & downloading | WGET |
| 2. | Analysis | NotePad++ |
| 3. | Parsing | MemoQ (Filters features) |
| 4. | Aligning | MemoQ (LiveDocs feature) |
| 5. | Compiling | MemoQ (Export to TMX feature) |
| 6. | Cleaning | Goldpan TMX/TBX Editor |
| 7. | File conversion | Okapi Rainbow |

*Table 1. The tools used and steps followed to build the TM.*

Following the steps shown, we created a TM containing 27,279 sentences and 496,327 source words,[30] with a word/sentence rate of 18.19.[31] Below, we display the profile of our TM together with some meta information about it.

| Journal name | *Archives of the Turkish Society of Cardiology* |
|---|---|
| Domain | Cardiology |
| UNESCO code | 3205.01 |
| Source word count | 496327 |
| Target word count | 570082 |
| Sentence count | 27279 |
| Source word / sentence rate | 18.19 |
| Target word / sentence rate | 20.89 |

*Table 2. TM profile for the first journal, as calculated by memoQ's Statistics feature.*

[30] Word count based on the Statistics feature of memoQ.
[31] Based on dividing the number of words by the number of sentences.

Lastly, the challenges encountered while preparing this TM should be mentioned. Our first trials of crawling with HTTrack resulted in files that were randomly named, or named in a way that was not conducive to grasping the naming pattern. In order to solve this problem, we transitioned to WGET in Ubuntu. Secondly, while the HTML pages were encoded in UTF-8, using this encoding yielded noisy characters. By trial and error, we found the correct encoding for the TM. Some translations into English had not been performed sentence by sentence, which resulted in misalignments. In order to minimise this, we used terms as anchors in memoQ to increase alignment reliability. We derived the terms in question from the "keywords" section of the abstracts. Our manual checking of the sentence-level alignments was minimal, since the overall alignment was considered to be adequate and possible misalignments were detected by Goldpan and then removed.

We used the procedure explained in Table 1 for all four journals, with some minor changes. Therefore, in the next sections, we will briefly explain the preparation procedure, repeating the same steps one by one.

### *4.2. TM preparation with abstracts from the Turkish Journal of Cardiovascular Nursing*

The *Turkish Journal of Cardiovascular Nursing* is another journal published by the Turkish Society of Cardiology, more specifically by the Cardiovascular Nursing Technicians Working Group. It is "an Open Access, peer-reviewed e-journal that considers scientific research, case reports, reviews, translations, letters to the editor, news and abstracts presented at the National Congress of Cardiology".[32] The topics covered include "the field of coronary artery disease, valvular heart disease, arrhythmias, heart efficacy, hypertension, congenital heart disease and all articles related to the coronary intensive care nursing".[33] The website of the journal has a similar design to that of the previous journal, as both are from the same society; hence, we used a similar procedure for this journal. We crawled the website with WGET, selected the relevant files, analysed them, determined the parsing filter to use, and implemented it. We used a cascading filter with a regex text filter and an HTML filter, just like we did with the first journal. The regex for Turkish abstracts included only one rule:

**Import only the following content:**

1. <font color=#515151><h2>.*<br></font><br><hr noshade size=4 align=center color=#d3d3d3><font color=#515151><h2>

*Figure 9. Filtre configuration.*

The reason why only one rule was used is that the structure of the HTML allowed for easier parsing. And the above tags were only included once in the file. If they had been included more than once, it would have been impossible to use this rule. The example of the fourth journal, presented in section 4.4, will better illustrate this. After

[32] http://khd.tkd.org.tr/EN/about (last access: 23.02.2024)
[33] http://khd.tkd.org.tr/EN/about (last access: 23.02.2024)

the regex text filter, an HTML filter was applied too. The cascading filter for the English abstracts included a regex filter and one HTML filter. The regex text included a single rule:

**Import only the following content:**

1. <font color=#515151><h2>.*<br></font><br><hr noshade size=4 align=center color=#d3d3d3><font color=#515151><h2>

*Figure 10. Filtre configuration.*

As can be observed, the same parsing filter rule was used for both cases. The only difference between the English and Turkish files was the order of the abstracts: in the Turkish files, the Turkish abstract came first, while in the English files, the English abstract came first. Hence, using the same filter configuration, we were able to filter and align the journal abstracts. After alignment, we imported the content into a TM and exported it as a TMX file. In total, we had 1,093 sentences, 17,471 source words and 21,019 target words.

| Journal name | *Turkish Journal of Cardiovascular Nursing* |
|---|---|
| Domain | Cardiology |
| UNESCO code | 3205.01 |
| Source word count | 17471 |
| Target word count | 21019 |
| Sentence count | 1093 |
| Source word / sentence rate | 15.98 |
| Target word / sentence rate | 19.23 |

*Table 3. TM profile for the second journal, as calculated by memoQ's Statistics feature.*

We also followed a cleaning procedure similar to the one used for the TM derived from the journal in the previous section, and removed the tags from the TMX files as well as unnecessary content through Goldpan. The preparation of this TM did not pose any challenges. The only problem is that it is relatively small. However, its content is useful for the TRENCARD Corpus.

### *4.3. TM preparation with abstracts from the Turkiye Klinikleri Journal of Cardiology*

The *Turkiye Klinikleri Journal of Cardiology* is another cardiology journal focusing on research publication in Turkey. It was published between 1988 and 2005. Its archive is distributed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0) licence. After crawling its website with WGET, we

developed a strategy to parse the files. We obtained 1,018 files (abstracts). In this case, the files were downloaded with their original article title names without number references. Hence, file alignment was not possible. However, each file included both the Turkish and English abstracts. Using only one file in the parsing step, we were able to divide the source content and the target content, and then align them sentence by sentence. In a similar setup, a cascading filter with a regex text filter and an HTML filter was used. The regex for Turkish abstracts included only one rule:

**Import only the following content:**

1. ÖZET<br>.*

*Figure 11. Filter configuration.*

And the regex for English abstracts also included only one rule:

**Import only the following content:**

1. ABSTRACT<br>.*

*Figure 12. Filter configuration.*

The abstract content was between tags that appeared only once in the file. Otherwise, this kind of extraction would not have been possible. The fourth journal will illustrate such a case.

| Journal name | *Turkiye Klinikleri Journal of Cardiology* |
|---|---|
| Domain | Cardiology |
| UNESCO code | 3205.01 |
| Source word count | 118314 |
| Target word count | 133997 |
| Sentence count | 7384 |
| Source word / sentence rate | 16.02 |
| Target word / sentence rate | 18.14 |

*Table 4. TM profile for the third journal, as calculated by memoQ's Statistics feature.*

After parsing and alignment, we obtained our TMX file and cleaned it of HTML tags. This yielded 7,384 sentences, 118,314 source words and 133,997 target words, as displayed in the table above.

### *4.4. TM preparation with abstracts from the Turkish Journal of Thoracic and Cardiovascular Surgery*

The *Turkish Journal of Thoracic and Cardiovascular Surgery* is the last journal that we crawled for TM preparation. It is another journal published in Turkish and English.

The *Turkish Journal of Thoracic and Cardiovascular Surgery* "is an international, open-access journal that welcomes articles on the subjects of cardiovascular surgery, cardiovascular anesthesia, cardiology, and thoracic surgery. The journal publishes all relevant clinical, surgical, and experimental studies, editorials, current and collective reviews, technical "How to Do It" articles, case reports, interesting images, video articles, reports of "New Ideas", correspondences, and commentaries."[34]

It is licensed under a Creative Commons Attribution-NonCommercial 4.0 International licence. The website of the journal has a similar structure to those of the first two journals. However, in this case, the content of the abstracts could not be extracted with the cascading filter (regex text filter and HTML filter) used before since it was not included between unique tags or texts. For example, the tag " " was repeated several times; hence, when we tried to parse the content between this tag and another tag, the tool could not decide where to start parsing. Different combinations of regex text filters were experimented with; however, it was not possible to extract content with this method.

Since a typical abstract file in this journal did not include too much noisy content, we decided to use a default HTML filter. The noisy content that had to be imported included menu items, the journal description in English, and some meta data about how many times an abstract had been viewed, authors, etc. Since the content in question would be in both the source segment and the target segment without change, it was possible to remove it by using the "Remove Duplicate" function in Goldpan TMX Editor. To sum up, we parsed the abstracts with the HTML filter, aligned them, and exported them into a TMX file. Then, in Goldpan, we removed the tags and the duplicated segments so that, in the end, only the sentences of the abstract remained. That meant spending more time on the cleaning phase. However, the result was the same. In the end, we had 13,937 sentences, 155,934 source words and 182,284 target words.

It can be observed that it is possible to achieve the same TM by concentrating on different steps of the TM preparation procedure, by first analysing the files in detail, knowing the specificities of each tool in the stack, and by implementing each step accordingly.

[34] http://tgkdc.dergisi.org/static.php?id=4

| Journal name | *Turkish Journal of Thoracic and Cardiovascular Surgery* |
|---|---|
| Domain | Cardiology |
| UNESCO code | 3205.01 |
| Source word count | 155934 |
| Target word count | 182284 |
| Sentence count | 13937 |
| Source word / sentence rate | 11.18 |
| Target word / sentence rate | 13.07 |

*Table 5. TM profile for the fourth journal, as calculated by memoQ's Statistics feature.*

## 5 TRENCARD Corpus compiled

Following the TM preparation procedure, we obtained four corpora on cardiology in TMX format. Despite the fact that multiple files can be combined into a single TMX file, keeping them separate allows for a more fine-grained analysis of the results for each TM. Besides, as mentioned in section 4, due to licence limitations, it was only possible to redistribute the TM produced from the abstracts of the *Turkish Journal of Thoracic and Cardiovascular Surgery,* which amounted to approximately 20% of the whole study corpus.

| Name | TRENCARD CORPUS |
|---|---|
| Domain | Cardiology |
| UNESCO code | 3205.01 |
| Source word count | 788046 |
| Target word count | 907382 |
| Sentence count | 49693 |
| Source word / sentence rate | 15.85 |
| Target word / sentence rate | 18.25 |

*Table 6. TRENCARD Corpus word, sentence, and word/sentence counts, as calculated by memoQ's Statistics feature.*

Table 6 shows the compiled profile of the TRENCARD Corpus. We created a TM comprising 49,693 sentences, 788,046 source words and 907,382 target words. The source word/sentence count gives an idea of the average length of sentence in the

TM. The TRENCARD Corpus with content from all four journals is available on Google Drive and access for research purposes is subject to permission from the author.[35]

## 5. Limitations and discussion

The semi-automatic procedure explained above requires the use of more than one stand-alone tool. While translation tools are usually easy to use and are part of translator training (Rothwell & Svoboda, 2019), switching between different tools can still be somewhat time-consuming. In addition, automatic alignment may not yield correct alignment pairs and posterior manual alignment may require some time. Since CAT tools are not specifically made for batch alignment, they may lack some flexibilities that tools like HunAlign may have. Furthermore, if there are many files to be aligned, the alignment process may take several days.

In the case of a very narrow domain, even after performing all the steps in the methodology, the resulting domain-specific TM may still not be large enough to influence the output quality of a fine-tuned NMT engine and further corpus compilation may be needed. Ramírez-Sánchez (2022:174) describes all the technical steps of preparing domain-specific data for custom MT and highlights the necessity of having a "generous" amount of domain-specific data, but admits that it is hard to predict an exact amount. Dogru & Moorkens (2024) found that fine-tuning a pre-trained NMT engine with custom TMs in the localisation domain with 500,000 source words (69,500 sentences) provided significant quality improvements (compared to the baseline engine) across three language pairs. Fine-tuning studies with smaller amounts of domain-specific data can be performed to evaluate the effectiveness of such amounts of data. Nieminen (2021) and Balashov (2021) hint that as few as 10,000 sentences can be enough to get significant results. Ramírez-Sánchez (2022) also notes that even tiny amounts of data are now being considered in adaptive or incremental MT scenarios, indicating a shift towards recognising the value of any amount of data for learning and customisation of MT systems. Besides, preliminary studies involving fine-tuning with large language models (LLMs) are paving the way for better MT quality to be attained with a smaller amount of custom TM content. In this respect, see, for instance, Moslem *et al.* (2023), where 20,000 sentences improved MT quality in the medical domain, or the introduction of new technologies like GPTs.[36]

Data permissions and ethical data use constitute another limitation of our study. TM creation from the Web must observe the copyrights and use permissions of online resources, although this limits the available domain-specific data even more. Meticulous

[35] TRENCARD Corpus with all four corpora. https://drive.google.com/drive/folders/1E5UasfHEO9Qn668zgu4n8-UTe5cl3WCH?usp=sharing.
As mentioned in section 4, the TM with the content from the *Turkish Journal of Thoracic and Cardiovascular Surgery* is on GitHub and can be re-used for non-commercial purposes, including research.

[36] Introducing GPTs. https://openai.com/blog/introducing-gpts (last access: 23.02.2024)

observance of data permissions, ownership statuses, and ethical guidelines in the assembly of parallel corpora or TMs from Web sources is, in any case, imperative.

## 6. Conclusion

Translators can create a sizable parallel corpus in the form of a TM using a semi-automatic procedure with tools that are already available. Using the procedure outlined in this paper, we created a TM of 788,046 source words in a very specific domain in only four days. Based on a standard daily translation output of 2,500 words and 20 working days in a month, we can estimate that a translator can translate roughly 600,000 words a year. This makes the TRENCARD Corpus larger than the annual output of a translator. The use of this methodology can help translators compile sufficiently large TMs in their specialisation field and lets them fine-tune their locally installed NMT engines in user-friendly OPUS-CAT software, empowering them in their professional work with the use of state-of-the-art technology without the need to depend on generic proprietary MT systems. And since they will not need to train an MT engine from scratch, they will not need advanced technical skills or intensive computational power. This procedure can also be helpful to create parallel corpora in the form of TMs for low-resource languages.

Whether in TM form or in other parallel corpus forms, domain-specific data will continue to be important for translators, especially with data-intensive technologies such as LLMs. For this reason, it would be interesting in the future to introduce a new tool, following the inflation paradigm, combining all the corpus preparation steps described in our study within a single GUI, thus accelerating this preparation phase.

## Acknowledgements

This work was partly supported by the DESPITE-MT project “Description of Posteditese in Machine Translation” (grant number PID2019-108650RB-I00 [MICINN]) and by European Union-NextGenerationEU funding through a Margarita Salas postdoctoral grant. Moreover, this work is derived from my PhD research; I would like to thank Adria Martin Mor and Anna Aguilar Amat for their contribution to this research. I am also grateful to Dr. Joss Moorkens for his comments and feedback on this article.

## Bibliography

*Archives of the Turkish Society of Cardiology*. ISSN 1016-5169 | E-ISSN 1308-4488.URL: https://archivestsc.com/ [Accessed: 20241201].

Aston, Guy. (1999). Corpus Use and Learning to Translate. *Textus*, XII(2), 289–314.

Baker, Mona. (1993). Corpus Linguistics and Translation Studies – Implications and Applications. In Baker, Mona, Francis, Gill & Tognini-Bonelli, Elena. *Text and*

*Technology*. (pp. 233–252). Amsterdam/Philadelphia: John Benjamins. https://doi.org/10.1075/z.64.15bak . [Accessed: 20241201].

Balashov, Yuri (2021). OPUS-CAT: A State-of-the-Art Neural Machine Translation Engine on Your Local Computer. *The ATA Chronicle*. URL: https://www.atanet.org/tools-and-technology/opus-cat-a-state-of-the-art-neural-machine-translation-engine-on-your-local-computer [Accessed: 20241201].

Bañón, Marta; Esplà-Gomis, Miquel; Forcada, Mikel L.; García-Romero, Cristian; Kuzman, Taja; Ljubešić, Nikola; van Noord, Rik; Sempere, Leopoldo Pla; Ramírez-Sánchez, G

ema; Rupnik, Peter; Suchomel, Vít; Toral, Antonio; van der Werff, Tobias; Zaragoza, Jaume. (2022). MaCoCu: Massive collection and curation of monolingual and bilingual data: focus on under-resourced languages. *Proceedings of the 23rd Annual Conference of the European Association for Machine Translation*, 303–304. https://aclanthology.org/2022.eamt-1.41 [Accessed: 20241201].

Bowker, Lynne, & Pearson, Jennifer. (2002). *Working with Specialized Language: A Practical Guide to Using Corpora*. London & New York: Routledge. https://doi.org/10.4324/9780203469255 [Accessed: 20241201].

do Carmo, Felix. (2020). 'Time is money' and the value of translation. *Translation Spaces*, 9(1), 35–57. https://doi.org/10.1075/ts.00020.car [Accessed: 20241201].

Dogru, Gokhan, & Moorkens, Joss. (2024). Data Augmentation with Translation Memories for Desktop Machine Translation Fine-tuning in 3 Language Pairs. *The Journal of Specialised Translation*, (41), 149–178. https://doi.org/10.26034/cm.jostrans.2024.4716 [Accessed: 20241201].

Esplà-Gomis, Miquel; Forcada, Mikel; Ramírez-Sánchez, Gemma; & Hoang, Hieu. (2019). ParaCrawl: Web-scale parallel corpora for the languages of the EU. *Proceedings of MT Summit XVII*, volume 2, (pp. 118 - 119).

ELIS (2023). *European Language Industry Survey 2023. Trends, expectations and concerns of the European language industry*. https://elis-survey.org/wp-content/uploads/2023/03/ELIS-2023-report.pdf [Accessed: 20241201].

Farrell, Michael. (2022). Do translators use machine translation and if so, how? Results of a survey held among professional translators. *Proceedings of 44th Conference Translating and the Computer*. https://doi.org/10.13140/RG.2.2.33996.69768 [Accessed: 20241201].

Fırat, Gokhan. (2021). Uberization of translation: Impacts on working conditions. *The Journal of Internationalization and Localization*, 8(1), 48–75. https://doi.org/10.1075/jial.20006.fir [Accessed: 20241201].

Gilbert, Devin. (2020). Using Commercially Available Customizable NMT to Study Translator Style. *TT5 Translation in Transition: Human and Machine Intelligence*.

Heafield, Kennet; Farrow, Elaine; van der Linde, Jelmer; Ramírez-Sánchez, Gema; Wiggins, Dion. (2022). The EuroPat Corpus: A Parallel Corpus of European Patent

Data. *Proceedings of the Thirteenth Language Resources and Evaluation Conference* (pp. 732–740). European Language Resources Association. https://aclanthology.org/2022.lrec-1.78 [Accessed: 20241201].

Koehn, Phillipp. (2005). Europarl: A Parallel Corpus for Statistical Machine Translation. *Conference Proceedings: the Tenth Machine Translation Summit*, (pp. 79-86). Phuket. Retrieved from http://homepages.inf.ed.ac.uk/pkoehn/publications/europarl-mtsummit05.pdf [Accessed: 20241201].

Koehn, Phillipp & Schroeder, J. (2007). Experiments in Domain Adaptation for Statistical Machine Translation. In C. Callison-Burch, P. Koehn, C. S. Fordyce, & C. Monz (Eds.), *Proceedings of the Second Workshop on Statistical Machine Translation* (pp. 224–227). Association for Computational Linguistics. https://doi.org/10.3115/1626355.1626388 [Accessed: 20241201].

Kraif, Olivier. (2002). Translation Alignment and Lexical Correspondence. Altenberg, Bengt and Granger, Sylviane (Eds). *Lexis in Contrast. Corpus-based approach* (pp. 271-290). Amsterdam & Philadelphia: John Benjamins. https://doi.org/10.1075/scl.7.19kra [Accessed: 20241201].

Läubli, Samuel; Amrhein, Chantal; Düggelin, Patrick; Gonzalez, Beatriz; Zwahlen, Alena; Volk, Martin (2019). Post-editing Productivity with Neural Machine Translation: An Empirical Assessment of Speed and Quality in the Banking and Finance Domain. *Proceedings of Machine Translation Summit XVII: Research Track* (pp. 267–272). European Association for Machine Translation. https://aclanthology.org/W19-6626 [Accessed: 20241201].

Le Bruyn, Bert; Fuchs, Martin; van der Klis, Martijn; Liu, Jianan; Mo, Chou; Tellings, Jos; de Swart, Henriette (2022). Parallel Corpus Research and Target Language Representativeness: The Contrastive, Typological, and Translation Mining Traditions. *Languages*, 7(3), Article 3. https://doi.org/10.3390/languages7030176 [Accessed: 20241201].

Marco, J., & von Lawick, H. (2009). Using corpora and retrieval software as a source of materials for the translation classroom. In Beeby, Allison; Rodríguez Inés, Patricia & Sánchez-Gijón, Pilar (Eds). *Corpus Use and Translating*: (pp. 9-28). Amsterdam & Philadelphia: John Benjamins.

Melby, Alan. K., & Wrigh, Sue, Ellen (2015). Translation Memory. In S.-W. Chan, *Routledge Encyclopedia of Translation Technology* (pp. 662-667). Routledge.

Mikhailov, Mikhail. (2022). Text corpora, professional translators and translator training. *The Interpreter and Translator Trainer*, 224-246. https://doi.org/10.1080/1750399X.2021.2001955 [Accessed: 20241201].

Moorkens, Joss. (2017). Under pressure: Translation in times of austerity. *Perspectives*, 25, 464–477. https://doi.org/10.1080/0907676X.2017.1285331 [Accessed: 20241201].

Moorkens, Joss. (2022). Ethics and machine translation. In Dorothy Kenny(ed.), *Machine translation for everyone: Empowering users in the age of artificial intelligence* (pp. 121–140). Language Science Press. https://doi.org/10.5281/zenodo.6759984 [Accessed: 20241201].

Moorkens, Joss., & Lewis, Dave. (2019a). Copyright and the reuse of translation as data. In M. O'Hagan (Ed.), In: O'Hagan, Minako, (ed.) *The Routledge Handbook of Translation and Technology*. Routledge Translation Handbooks (pp. 469–481). Routledge. http://dx.doi.org/10.4324/9781315311258-28 [Accessed: 20241201].

Moorkens, Joss., & Lewis, Dave. (2019b). Research Questions and a Proposal for the Future Governance of Translation Data. *The Journal of Specialised Translation*, 2–25. https://doi.org/10.4324/9781315311258-28 [Accessed: 20241201].

Moslem, Yasmen., Haque, Rajwanul., Kelleher, John. D., and Way, Andy. (2023). Adaptive Machine Translation with Large Language Models. In *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*, pages 227–237, Tampere, Finland. European Association for Machine Translation.

Nieminen, Tommi. (2021). OPUS-CAT: Desktop NMT with CAT Integration and Local Fine-tuning. *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, (pp. 288–294). https://doi.org/10.18653/v1/2021.eacl-demos.34 [Accessed: 20241201].

Nimdzi. (2023). Nimdzi Language Technology Atlas: The Definitive Guide to the Language Technology Landscape. URL: https://www.nimdzi.com/language-technology-atlas/ [Accessed: 20241201].

O'Brien, Sharon. (2012). Translation as human–computer interaction. *Translation Spaces*, 1(1), 101–122. https://doi.org/10.1075/ts.1.05obr [Accessed: 20241201].

Ramírez-Sánchez, Gemma. (2022). Custom machine translation. In Kenny, Dorothy (Ed.), *Machine translation for everyone: Empowering users in the age of artificial intelligence* (pp. 165-186). Berlin: Language Science Press.. https://doi.org/10 . 5281/zenodo.6760022 [Accessed: 20241201].

Rothwell, Andrew, & Svoboda, Tomas. (2019). Tracking translator training in tools and technologies: Findings of the EMT survey 2017. *The Journal of Specialised Translation*, 2019.

Pérez-Ortiz, Juan Antonio; Forcada, Mikel.; Sánchez-Martínez, Felipe (2022). How neural machine translation works. In Kenny Dorothy, *Machine translation for everyone: Empowering users in the age of artificial intelligence* (pp. 141-164). Dublin: Language Science Press.

Sánchez-Gijón, Pilar. (2009). Developing Documentation Skills to Build Do-It-Yourself Corpora in the Specialized Translation Course. In Beeby, Allison; Rodríguez Inés, Patricia & Sánchez-Gijón, Pilar (Eds). *Corpus Use and Translating* (pp. 109-127).

Amsterdam & Philadelphia: John Benjamins. https://doi.org/10.1075/btl.82.08san [Accessed: 20241201].

Tiedemann, Jörg & Nygaard, Lars. (2004). The OPUS Corpus - Parallel and Free: Http://logos.uio.no/opus. In M. T. Lino, M. F. Xavier, F. Ferreira, R. Costa, & R. Silva (Eds.), *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*. European Language Resources Association (ELRA). http://www.lrec-conf.org/proceedings/lrec2004/pdf/320.pdf [Accessed: 20241201].

Tiedemann, Jörg, & Thottingal, Santhosh. (2020). OPUS-MT – Building open translation services for the World. *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation* (pp. 479-480). Lisboa: European Association for Machine Translation.

*Turkish Journal of Cardiovascular Nursing*. e-ISSN 2149-4975. https://khd.tkd.org.tr/ [Accessed: 20241201].

*Turkiye Klinikleri Journal of Cardiology Journal Identity*. 1988 – 2005. https://www.turkiyeklinikleri.com/journal/journal-of-cardiology/42/identity/en-index.html [Accessed: 20241201].

*Turkish Journal of Thoracic and Cardiovascular Surgery*. e-ISSN: 2149-8156. ISSN: 1301-5680. https://tgkdc.dergisi.org/index.php [Accessed: 20241201].

Zanettin, Federico. (2012). *Translation-Driven Corpora*. New York: Routledge.