# Towards a Clean Text Corpus for Ottoman Turkish

**Fatih Burak Karagöz** and **Berat Doğan** and **Şaziye Betül Özateş**
Boğaziçi University, Turkey

{fatih.karagoz,berat.dogan}@std.bogazici.edu.tr, saziye.ozates@bogazici.edu.tr

**Abstract**

Ottoman Turkish, as a historical variant of modern Turkish, suffers from a scarcity of available corpora and NLP models. This paper outlines our pioneering endeavors to address this gap by constructing a clean text corpus of Ottoman Turkish materials. We detail the challenges encountered in this process and offer potential solutions. Additionally, we present a case study wherein the created corpus is employed in continual pre-training of BERTurk, followed by evaluation of the model's performance on the named entity recognition task for Ottoman Turkish. Preliminary experimental results suggest the effectiveness of our corpus in adapting existing models developed for modern Turkish to historical Turkish.

## 1 Introduction

Natural Language Processing (NLP) has been extensively facilitated through widely spoken modern languages. These models cover various fields, from sentiment analysis to medical assessments and question-answering. Such applications require extensive data sources, which are relatively more straightforward to collect and optimize for modern languages due to the abundance of digitized documents available on the Internet. However, integrating such applications into historical and less commonly spoken languages presents significant challenges due to the need for more available resources in these languages. Collecting and optimizing documents in these languages is more complex, necessitating more efficient data extraction methods to achieve comparable performance levels.

The development of software tools and the application of automation for historical languages are crucial for scholarly research, offering invaluable insights into political, sociological, and historical contexts. Among these languages, Ottoman Turkish has a significant legacy in history, literature, culture, and science, influencing three continents over 600 years. This extensive historical impact highlights the importance of applications and studies involving Ottoman Turkish, as they can generate profound value across various fields.

One of the pathways to such value creation is through the use of pre-trained language models (PLMs), which have revolutionized natural language processing by achieving state-of-the-art performance across many tasks. However, the availability of clean text corpora is essential for the automation of any language, as it is necessary for training algorithms to perform tasks such as text analysis (Agarwal et al., 2007), language modeling (Snæbjarnarson et al., 2022), and information extraction (Hamdi et al., 2020; Li et al., 2020). This is particularly important for pre-training language models, which require a comprehensive understanding of a language's statistical properties and intricate patterns. Therefore, our study aims to create a clean data corpus for the natural language processing of Ottoman Turkish. Despite its importance, integrating Ottoman Turkish into modern Natural Language Processing (NLP) frameworks presents significant challenges. Ottoman texts' unique linguistic and structural characteristics require specialized approaches for effective digitization, standardization, and analysis. Leveraging advanced NLP techniques, such as pre-training BERT models, has the potential to scale these studies and meet the growing demand for research in this area.

While striving to create a clean corpus for Ottoman Turkish texts, we faced several challenges that impeded effective data collection. These challenges were addressed in four phases: (i) Converting PDF documents into clean text files, (ii) normalizing unique characters, (iii) handling intertwined bidirectional text in Arabic and Latinized Turkish, and (iv) minimizing the impact of decorative textures. The solutions needed to be simple and cost-efficient in terms of computational power allocation, adhering to the philosophical principle of

Occam's Razor (Bowen and Breuer, 1992), which advocates simplicity. We chose Regular Expression (Regex) methods over competing hypotheses and sophisticated machine learning techniques because Regex requires minimal additional memory, in which optimizing computational overhead, and allows for immediate application through modes of Non-deterministic Finite Automata (NFA).

This paper summarizes our initial efforts to create a clean text corpus of Ottoman Turkish texts that can be used for various purposes including automatic processing of Ottoman Turkish. We mention some related work on data cleaning and extraction in Section 2. We explain the methodology followed to create the intended corpus of Ottoman Turkish texts and state the main challenges faced and possible solutions to them in Section 3. Then we provide a case study in Section 4 where we further pre-train the BERTurk (Schweter, 2020) model using our corpus to adapt it to Ottoman Turkish texts and fine-tune the model on a named entity recognition (NER) dataset for NER tagging of Ottoman Turkish. The preliminary experiment results suggest that further pre-training the BERTurk model, initially designed for modern Turkish, with Ottoman Turkish data is effective for Ottoman NER tagging. We conclude the paper and state future directions of this study in Section 5. To the best of our knowledge, this study represents the first attempt to provide language resources and models for state-of-the-art natural language processing of Ottoman Turkish.

## 2 Related Work

Modern languages provide relatively clean corpora when obtained from web text sources (Sharoff, 2006). In contrast, historical languages require a different approach due to the variety of digitization methods used by various institutions (Piotrowski, 2012). Unique challenges, such as non-standard orthography, mixed scripts, and the scarcity of digitized texts, necessitate specialized approaches. This section reviews several studies that have addressed these challenges and contributed to developing effective methods for language processing, which can be utilized in historical document automation.

The Impresso project (Ehrmann et al., 2020) focuses on the semantic indexing of a multilingual corpus of digitized historical newspapers. This interdisciplinary research involves computational linguists, designers, and historians collaborating to transform noisy and unstructured textual content into semantically indexed, structured, and linked data. The authors highlight the challenges posed by large-scale collections of digitized newspapers, including incomplete collections, extensive and messy data, noisy historical text, and the need for robust system architecture. The project emphasizes the importance of transparency and critical assessment of inherent biases in exploratory tools, digitized sources, and annotations.

Piotrowski (2012) highlights the importance of text normalization in processing historical languages. The study presents various techniques for handling non-standard orthography, including using historical dictionaries and context-based normalization algorithms. These methods help standardize the text, making it more suitable for NLP tasks. The challenges of interpreting private use area (PUA)[1] characters and integrating Arabic sentences within Ottoman Turkish texts are addressed through mapping systems and regular expressions.

Jain et al. (2021) propose an extensible parsing pipeline to process unstructured data, particularly within network monitoring and diagnostics. Their methodology employs rule-based extraction techniques to transform unstructured data into structured formats, utilizing pattern mining strategies and heuristics-based analysis. This approach demonstrates resilience to changes in data structure and effectively filters out extraneous information. Notably, the use of regular expressions for pattern recognition and data extraction mirrors the techniques employed in our study for processing Ottoman Turkish texts. The pipeline's capability to handle diverse data structures without necessitating labeled training data or manual intervention underscores its robustness and efficiency in data extraction tasks.

Nundloll et al. (2021) discuss the automation of information extraction from historical texts using the Journal of Botany as a case study. They document the use of OCR-based software for digitization and the subsequent application of NLP frameworks to customize entity recognition models. Tools like Prodigy and Spacy were employed to identify specific entities such as plant names, observers, locations, and attributes. The authors em-

[1]The code points in these regions are not standardized characters within Unicode. They remain intentionally undefined, enabling third parties to create their own characters without clashing with the assignments made by the Unicode Consortium.

phasize the importance of creating training and test datasets to evaluate the accuracy of the entity recognition models, highlighting the iterative process of model training, error-checking, and annotation.

## 3 Methods, Challenges, and Solutions

This study employs a systematic methodology to extract, standardize, and analyze Ottoman Turkish texts from historical documents. The process involves several key steps and leverages various tools and libraries to ensure efficient and accurate data handling. In the following subsections, we first explain the source of data and then give the two main steps of our method[2] for creating a clean corpus of Ottoman Turkish texts. For each step, we state the challenges faced and explain our solutions to them.

### 3.1 Data

There are two primary Ottoman periodicals used as data source: Sebilürreşad and Sırat-ı Müstakim magazines.

Sırat-ı Müstakim was a prominent Ottoman Turkish magazine that was first published on 1908. During the period of the Second Constitutional Monarchy, Mehmed Akif, a famous Turkish writer at the time, took the position of editorial writer for this magazine. Published weekly, the magazine included written texts about various topics, including religious, national, literary, and political issues (Gündoğdu, 2008). The periodical was published under the same name until 1912, and from issue 183 it continued to be published under the name Sebilürreşad until 1925. Sebilürreşad had to suspend its publication starting from its 641st issue on 1925. Later, it resumed publication in May 1948, until March 1965, during which it released 359 issues (Ceyhan, 1991). These periodicals, first published as Sırat-ı Müstakim and later as Sebilürreşad, contributed to Turkish culture, art, literature, and intellectual life for a total of thirty-four years (Çakır, 2014).

In our study, the issues of both periodicals published between 1908-1925 have been taken into account. These issues have been collected as twenty five volumes (seven of them under the name Sırat-ı Müstakim and remaining eighteen as Sebilürreşad) and made publicly available as images by National Library of Turkey[3] and as PDF documents in Latin script by Bağcılar Municipality.[4] Figure 1 shows example segments from each periodical.

### 3.2 Step 1: Data Extraction

The two periodicals used as our data source were in PDF format. These PDF documents were created using OCR systems and have a complex structure due to the usage of both Latinized Turkish and Arabic letters, recursive polluted texts such as dates, page numbers, prices of the magazine, and illustrations used in Ottoman textures. Also, the source documents followed inconsistent margin formats and illustrations. These inconsistencies varied from document to document and page to page within the same file. Variation in calligraphy, irrelevant notes, and irregular design further exacerbating these challenges. These elements often led to misconducting of text and incorrect character recognition, leading to a poor corpus cleaning. Such variations posed significant challenges for accurate text extraction and processing.

Given these complexities, using multiple post-OCR approaches could have been a more optimal strategy for document assembly modeling (Lopresti and Jiangying, 1997; D'hondt et al., 2017; Schulz and Kuhn, 2017). However, OCR performance heavily depends on image quality, and implementing document-specific OCR solutions would be computationally intensive and inefficient. Also, most of these approaches require labeled training data for post-OCR correction. Hence, we opted not to use these supervised approaches in the data extraction phase.

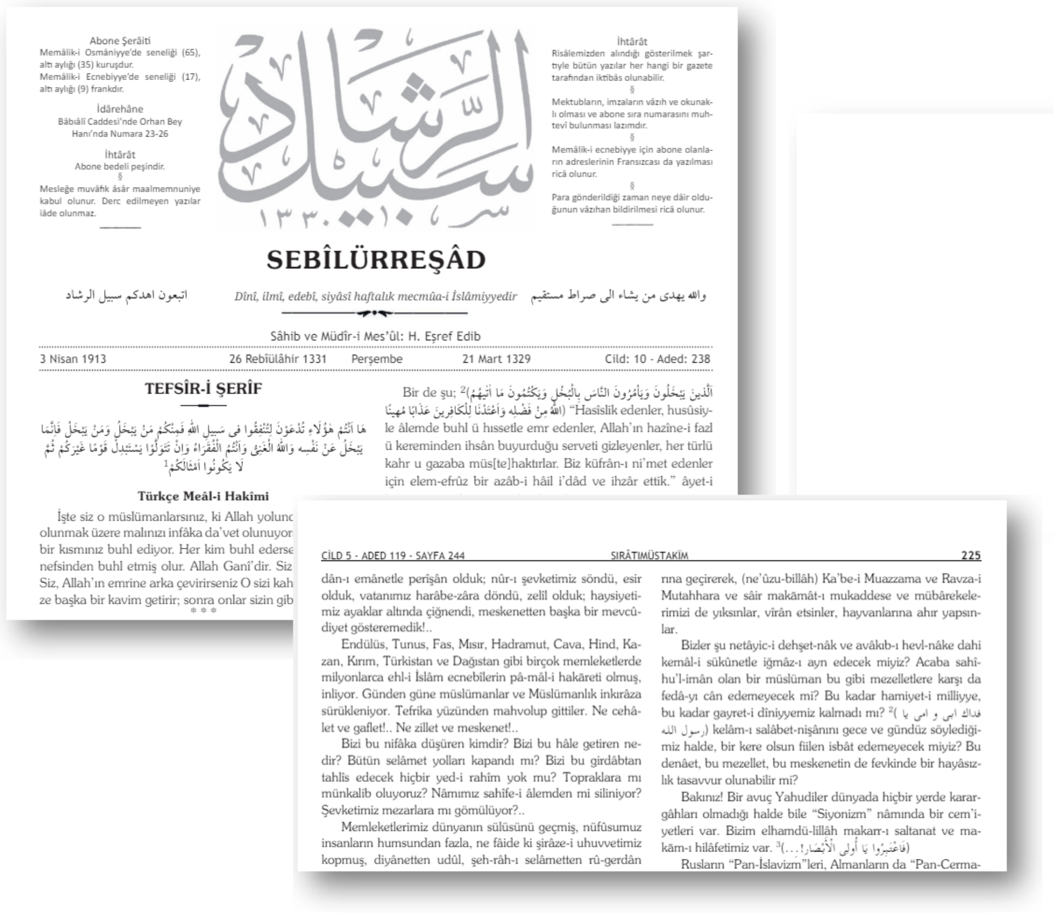
In order to represent the text data in a simpler and cleaner format, we convert PDFs into TXT format while maximizing text extraction accuracy. The goal is to ensure that the resulting text files represent the original content of the historical documents. For this purpose, we utilize a range of Python tools and libraries.

Initially, we utilized Pdfplumber, a Python library known for its effective text extraction capabilities. Our first attempt was defining rigid constraints on page margins to capture central text bodies, excluding footnotes, repetitive dates, and titles outside of these margins. However, these constraints were less effective when the text alignment varied. Some documents started with wider margins in a single column and changed arbitrarily. Consequently, margin framing failed to extract

[2]The code is available under `https://github.com/Ottoman-NLP/Ottoman_LLM_Repos`.

[3]Milli Kütüphane. https://www.millikutuphane.gov.tr/

[4]Bağcılar Belediyesi. https://www.bagcilar.bel.tr/



Abone Şerâiti
Memâlik-i Osmâniyye'de seneliği (65), altı aylığı (35) kuruşdur.
Memâlik-i Ecnebiyye'de seneliği (17), altı aylığı (9) frankdır.

İdârehâne
Bâbıâlî Caddesi'nde Orhan Bey Hanı'nda Numara 23-26

İhtârât
Abone bedeli peşindir.
§
Mesleğe muvâfık âsâr maalmemnuniye kabul olunur. Derc edilmeyen yazılar iâde olunmaz.

İhtârât
Risâlemizden alındığı gösterilmek şartıyle bütün yazılar her hangi bir gazete tarafından iktibâs olunabilir.
§
Mektubların, imzaların vâzıh ve okunaklı olması ve abone sıra numarasını muhtevî bulunması lazımdır.
§
Memâlik-i ecnebiyye için abone olanların adreslerinin Fransızcası da yazılması ricâ olunur.
§
Para gönderildiği zaman neye dâir olduğunun vâzıhan bildirilmesi ricâ olunur.

SEBÎLÜRREŞÂD

اتبعون اهدكم سبيل الرشاد *Dînî, ilmî, edebî, siyâsî haftalık mecmûa-i İslâmiyyedir* والله يهدى من يشاء الى صراط مستقيم

Sâhib ve Müdîr-i Mes'ûl: H. Eşref Edib

3 Nisan 1913 26 Rebîülâhir 1331 Perşembe 21 Mart 1329 Cild: 10 - Aded: 238

TEFSÎR-İ ŞERÎF

هَا اَنْتُمْ هٰؤُلَاءِ تُدْعَوْنَ لِتُنْفِقُوا فى سَبِيلِ اللهِ فَمِنْكُمْ مَنْ يَبْخَلُ وَمَنْ يَبْخَلْ فَاِنَّمَا يَبْخَلُ عَنْ نَفْسِه وَاللهُ الْغَنِىُّ وَاَنْتُمُ الْفُقَرَاءُ وَاِنْ تَتَوَلَّوْا يَسْتَبْدِلْ قَوْمًا غَيْرَكُمْ ثُمَّ لَا يَكُونُوا اَمْثَالَكُمْ[1]

Türkçe Meâl-i Hakîmi

İşte siz o müslümanlarsınız, ki Allah yolunc
olunmak üzere malınızı infâka da'vet olunuyor
bir kısmınız buhl ediyor. Her kim buhl ederse
nefsinden buhl etmiş olur. Allah Ganî'dir. Siz
Siz, Allah'ın emrine arka çevirirseniz O sizi kah
ze başka bir kavim getirir; sonra onlar sizin gib

* * *

Bir de şu; [2](اَلَّذِينَ يَبْخَلُونَ وَيَأْمُرُونَ النَّاسَ بِالْبُخْلِ وَيَكْتُمُونَ مَا اٰتٰيهُمُ اللهُ مِنْ فَضْلِه وَاَعْتَدْنَا لِلْكَافِرِينَ عَذَابًا مُهِينًا) "Hasîslik edenler, husûsiyle âlemde buhl ü hissetle emr edenler, Allah'ın hazîne-i fazl ü kereminden ihsân buyurduğu serveti gizleyenler, her türlü kahr u gazaba müs[te]haktırlar. Biz küfrân-ı ni'met edenler için elem-efrûz bir azâb-i hâil i'dâd ve ihzâr ettik." âyet-i

CİLD 5 - ADED 119 - SAYFA 244 SIRÂTIMÜSTAKÎM 225

dân-ı emânetle perîşân olduk; nûr-ı şevketimiz söndü, esir olduk, vatanımız harâbe-zâra döndü, zelîl olduk; haysiyetimiz ayaklar altında çiğnendi, meskenetten başka bir mevcûdiyet gösteremedik!..

Endülüs, Tunus, Fas, Mısır, Hadramut, Cava, Hind, Kazan, Kırım, Türkistan ve Dağıstan gibi birçok memleketlerde milyonlarca ehl-i İslâm ecnebîlerin pâ-mâl-i hakāreti olmuş, inliyor. Günden güne müslümanlar ve Müslümanlık inkırâza sürükleniyor. Tefrika yüzünden mahvolup gittiler. Ne cehâlet ve gaflet!.. Ne zillet ve meskenet!..

Bizi bu nifâka düşüren kimdir? Bizi bu hâle getiren nedir? Bütün selâmet yolları kapandı mı? Bizi bu girdâbtan tahlîs edecek hiçbir yed-i rahîm yok mu? Topraklara mı münkalib oluyoruz? Nâmımız sahîfe-i âlemden mi siliniyor? Şevketimiz mezarlara mı gömülüyor?..

Memleketlerimiz dünyanın sülüsünü geçmiş, nüfûsumuz insanların humsundan fazla, ne fâide ki şirâze-i uhuvvetimiz kopmuş, diyânetten udûl, şeh-râh-ı selâmetten rû-gerdân rına geçirerek, (ne'ûzu-billâh) Ka'be-i Muazzama ve Ravza-i Mutahhara ve sâir makāmât-ı mukaddese ve mübârekelerimizi de yıksınlar, vîrân etsinler, hayvanlarına ahır yapsınlar.

Bizler şu netâyic-i dehşet-nâk ve avâkıb-ı hevl-nâke dahi kemâl-i sükûnetle iğmâz-ı ayn edecek miyiz? Acaba sahîhu'l-imân olan bir müslüman bu gibi mezelletlere karşı da fedâ-yı cân edemeyecek mi? Bu kadar hamiyet-i milliyye, bu kadar gayret-i dîniyyemiz kalmadı mı? [2](فداك ابى و امى يا رسول الله) kelâm-ı salâbet-nişânını gece ve gündüz söylediğimiz halde, bir kere olsun fiilen isbât edemeyecek miyiz? Bu denâet, bu mezellet, bu meskenetin de fevkinde bir hayâsızlık tasavvur olunabilir mi?

Bakınız! Bir avuç Yahudiler dünyada hiçbir yerde karargâhları olmadığı halde bile "Siyonizm" nâmında bir cem'iyetleri var. Bizim elhamdü-lillâh makarr-ı saltanat ve makām-ı hilâfetimiz var. [3](فَاعْتَبِرُوا يَا أُولِى الْأَبْصَارِ!...)

Rusların "Pan-İslavizm"leri, Almanların da "Pan-Cerma-

Figure 1: Segments of two randomly selected pages from the magazines Sebilürreşad (back) and Sırat-ı Müstakim (front).

text from documents that did not adhere to defined rules.

After the unsuccessful trial in setting up one script to extract all varied documents, we consider another scheme where each document should have a unique margin frame sets handling the extraction processes. However, as discussed earlier, any given document might not consistently follow the same format, as the format occasionally changes from page to page as well. Furthermore, creating scripts for each document is an energy deterrent process and disfavors automation extraction for later models.

For these reasons, the library used for PDF conversion was later changed to PyMuPDF, a high performance Python library for data extraction and analysis for PDF documents, to ensure the expected extraction of documents. In this alternative script, the PDFs are converted without applying any margin rules into plain texts, and documents are subsequently reformatted using regular expression rules during the text manipulation. By removing margin constraints and employing direct text extraction, the processing time for each document is significantly reduced. This improvement in throughput enables the handling of larger datasets within shorter time-frames. We then utilized regular expressions to define a regular pattern recognition module. The illustrations are ignored and remaining Arabic sentences did not require any intelligent character recognition. This proved to be a computationally efficient and time-effective method, allowing us to have a more robust and efficient text processing pipeline by addressing the specific challenges posed by the unique characteristics of Ottoman Turkish documents without using post-OCR techniques.

## 3.3 Step 2: Text Standardization

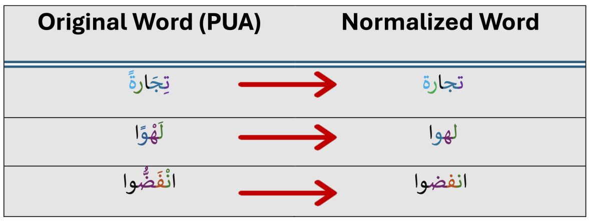
Standardizing the extracted text is crucial for proponent analysis. Regular Expressions (Regex) are employed to normalize the text, addressing inconsistencies such as varying orthographic representations and diacritics. This step ensures a uniform text format, facilitating more reliable processing and analysis.

We utilize various techniques for noise filtering, which involves removing non-essential components, segmenting relevant categories, and cleansing the data. In this step, pattern recognition is integral in identifying and extracting specific patterns within the text, such as dates, names, or other structured information. In the following subsections we discuss the challenges we faced in the text standardization step and explain how we overcome these challenges.

### 3.3.1 Normalization and Mapping

Some Arabic characters in the documents are occasionally interpreted as Private Use Area (PUA) characters (Unicode Consortium, 2021), as well as some Latinized Turkish words loaned from Arabic. This misinterpretation leads to poor identification of Arabic text using standard Unicode ranges. Initially, a function was used to eliminate characters with ordinals above 128; however, this approach inadvertently removed both standard Arabic and PUA characters, resulting in incomplete and inaccurate text standardization.

In historical documents, it is plausible for PUA characters to appear due to various factors such as font issues, scanning techniques, OCR software limitations or non-standard encoding. This is where character normalization technique plays a pivotal role as it standardizes various representations of characters to ensure document uniformity. This hurdle is overcome through normalization process where different forms of the same letter, such as accented characters, are converted to a common form. We developed a mapping system to identify and replace PUA characters with their equivalent standard Arabic variations. If a character had no equivalent, it was removed. As character normalization applied the text, UNICODE range matching for character level precision became operable. Figure 2 depicts this process on example words.



| Original Word (PUA) | | Normalized Word |
|---|---|---|
| تِجَارَةً | → | تجارة |
| لَهْوًا | → | لهوا |
| انْفَضُّوا | → | انفضوا |

Figure 2: Normalization process of PUA characters by mapping.

In addition to the normalization of Arabic characters, we also mapped accented Latin letters that are no longer present in the current Turkish alphabet to their equivalents. This is for reducing the number of unknown words due to the accented letters when adapting a model developed for modern Turkish to Ottoman Turkish. The outcome of this operation is visible in Figure 3 which shows a segment from the Sebilürreşad magazine and its processed version.

### 3.3.2 Right-to-Left Scripts (RTL)

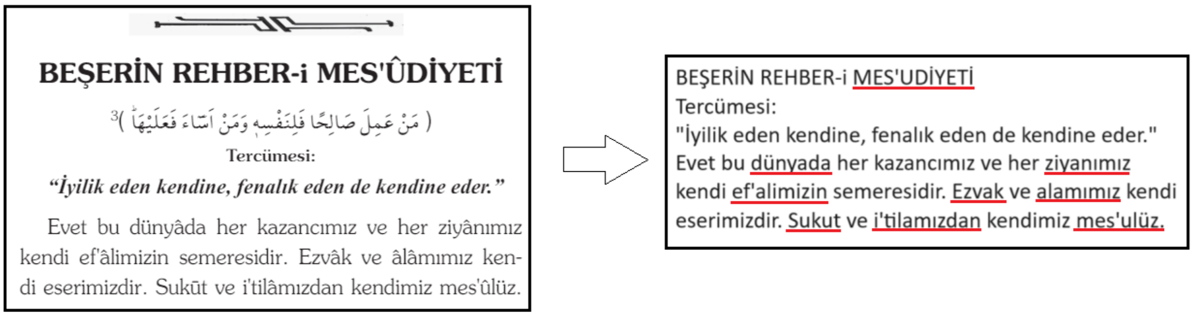
Arabic sentences posed another significant challenge for regex patterns, as well. Since Arabic is written from right to left, sentences containing both Latinized Turkish and Arabic text intertwined causing data-pollution as can be seen in the example in Figure 4a. We debated whether to filter out Arabic sentences altogether. However, Arabic texts often provide relevant references, adding potential contextual value to the documents. Therefore, it is concluded that more viable options should be employed by preserving Arabic sentences to increase document enhancement.

Therefore, we implemented a method to move Arabic sentences to a new line and separate them from Turkish sentences. The effect of this method is visible in Figure 4b. This approach ensures that each language maintains its correct orientation and readability in different lines. Additionally, this separation makes it easier to define pattern rules for different orthographies between Turkish and Arabic. By segmenting Arabic characters, we effectively isolated the two languages, preserving regex pattern functionality.

### 3.3.3 Optional Translation Feature

After this segmentation process, Arabic sentences were intended to be translated through an API to enhance the contextual information of the main text. However, directly integrating the translated Arabic sentences within the main document proved to be resource-intensive and time-consuming. Prior to API translation, the script needed to encapsulate Arabic sentences individually to flag their positions for replacing the original text with the translations. While processing the document in chunks helped manage memory and reduce complexity, the NFA-like backtracking and segmentation functionality increased inefficiency, led to incorrect translations, and caused frequent hits to the API rate limit.

Consequently, we proposed a new method: segmenting Arabic sentences into a separate text doc-



BEŞERİN REHBER-i MES'ÛDİYETİ

³( مَنْ عَمِلَ صَالِحًا فَلِنَفْسِهٖ وَمَنْ اَسَاءَ فَعَلَيْهَا )

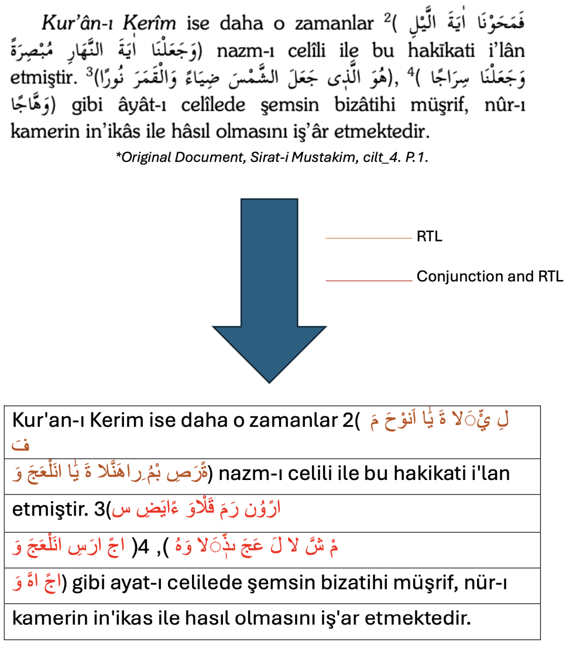
Tercümesi:

*"İyilik eden kendine, fenalık eden de kendine eder."*

Evet bu dünyâda her kazancımız ve her ziyânımız kendi ef'âlimizin semeresidir. Ezvâk ve âlâmımız kendi eserimizdir. Sukūt ve i'tilâmızdan kendimiz mes'ûlüz.

BEŞERİN REHBER-i MES'UDİYETİ
Tercümesi:
"İyilik eden kendine, fenalık eden de kendine eder."
Evet bu dünyada her kazancımız ve her ziyanımız kendi ef'alimizin semeresidir. Ezvak ve alamımız kendi eserimizdir. Sukut ve i'tilamızdan kendimiz mes'ulüz.

Figure 3: A segment from the PDF document of the 628th issue, 25th volume of Sebilürreşad magazine (on the left) and its processed version (on the right). Note the omission of the Arabic sentence and normalization of accented letters (in underlined words) in the processed version.



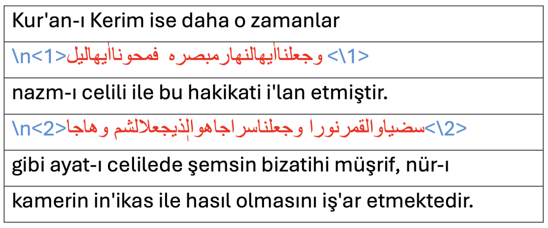
*Kur'ân-ı Kerîm* ise daha o zamanlar ²( فَمَحَوْنَا اٰيَةَ الَّيْلِ وَجَعَلْنَا اٰيَةَ النَّهَارِ مُبْصِرَةً) nazm-ı celîli ile bu hakīkati i'lân etmiştir. ³(هُوَ الَّذٖى جَعَلَ الشَّمْسَ ضِيَاءً وَالْقَمَرَ نُورًا), ⁴( وَجَعَلْنَا سِرَاجًا وَهَّاجًا) gibi âyât-ı celîlede şemsin bizâtihi müşrif, nûr-ı kamerin in'ikâs ile hâsıl olmasını iş'âr etmektedir.

**Original Document, Sirat-i Mustakim, cilt_4. P.1.*

RTL
Conjunction and RTL

| |
|---|
| Kur'an-ı Kerim ise daha o zamanlar 2( مَ حَوْنَا يٰة لا َّيْ لِ فَ |
| وَ جَعَلْنَا يٰة َالنَّهَا رِ مُبْ صِرَةً) nazm-ı celili ile bu hakikati i'lan |
| etmiştir. 3(س ضِيَاءً وَالْقَ مَرَ نُوْرًا |
| وَ جَعَلْنَا سِرَا جًا )4 ,( هُوَ الَّذِى جَعَ لَ ال شَّ مْ |
| وَ هَّا جًا) gibi ayat-ı celilede şemsin bizatihi müşrif, nür-ı |
| kamerin in'ikas ile hasıl olmasını iş'ar etmektedir. |

(a) PDF extraction by non-RTL formatting causes divergence.



| |
|---|
| Kur'an-ı Kerim ise daha o zamanlar |
| \n<1>فمحوناآيهالليل وجعلناآيهالنهارمبصره <\1> |
| nazm-ı celili ile bu hakikati i'lan etmiştir. |
| \n<2>هوالذيجعلالشمسضياوالقمرنورا وجعلناسراجاوهاجا<\2> |
| gibi ayat-ı celilede şemsin bizatihi müşrif, nür-ı |
| kamerin in'ikas ile hasıl olmasını iş'ar etmektedir. |

(b) Expected format for the text in Figure 4a.

Figure 4: The effect of RTL formatting.

ument and removing them from the primary text documents. These sentences' line order and text positions were mapped back to the main text separately. This segmentation allowed Arabic sentences to be translated independently, ensuring that contextual information relevant to the corresponding Turkish sentences was preserved without requiring complex filtering over the primary text documents. This approach significantly enhanced performance and simplified the rule-setting process for text manipulation. In the current version of the corpus however, we do not use this feature. Hence, the corpus does not include the translations of the Arabic sentences in it. At present, we exclude all Arabic texts from the corpus and only include Latinized Ottoman Turkish texts in it for the sake of simplification.

## 4 Evaluation on the Named Entity Recognition Task

As the result of the data extraction and cleaning steps explained in Section 3, we created a 17M-token corpus of Ottoman Turkish texts. In its current version, this corpus is too small to pre-train a transformer-based language model from scratch. Hence, in order to see its effect on a downstream NLP task, we further pre-train a PLM which was already pre-trained on various modern Turkish corpora. Further or continual pre-training is a common approach to train language models for low-resource languages (Liu et al., 2021; Micallef et al., 2022). By this way, we hope to benefit from the PLM's prior knowledge on Turkish words and grammatical structures that are common in modern Turkish and Ottoman Turkish.

In our preliminary experiments, we observed that among different architectures and models, BERTurk (Schweter, 2020), a Turkish language model utilizing the BERT architecture and pre-trained on modern Turkish text, reached the highest F1 scores on several NLP tasks. Hence, we chose to utilize BERTurk for our experiments. As in the architecture of the original BERT base model, BERTurk has 12 transformer layers. Each transformer layer consists of 12 attention heads and the number of hidden units is 768. The model includes a total of 110 million parameters that are fine-tuned during the pre-training phase on a large corpus of Turkish text data.

### 4.1 Continual Pre-training

We further pre-trained the BERTurk model with sentences from our corpus (885K sentences in total) to adapt it to the Ottoman Turkish context using Masked Language Modelling with 15% masking probability. We used Adam optimizer with the learning rate of 5e-5 and the batch size is 32. The training was performed on NVIDIA L4 accelerator with 22.5 GB of GPU RAM and a system RAM of 62.8 GB.

### 4.2 Fine-tuning on the Named Entity Recognition Task

As an extrinsic evaluation of our further pre-trained BERTurk model, we utilize the model for the task of named entity recognition (NER) on an Ottoman Turkish NER dataset. The main reason behind this choice is Ottoman Turkish being extremely low-resource in terms of labeled datasets and we have a newly annotated NER dataset for Ottoman Turkish, albeit small. This dataset contains 462 training, 200 validation, and 200 test sentences sourced from Servet-i Fünun journal. Due to the very rare occurrence of other entity types, annotation has been performed only for `PERSON` and `LOCATION` entities. The total number of `PERSON` entities in the dataset is 386 while the number of `LOCATION` entities is 794. The inter annotator agreement (IAA) between the two annotators of the dataset is measured as 0.82.

To observe the performance of our pre-trained model on the NER task, we fine-tuned the model on the mentioned Ottoman Turkish NER dataset for 10 epochs using Adam optimizer with 5e-5 learning rate. We chose the batch size to be 32. Table 1 shows the entity-level precision, recall and F1 scores of BERTurk with and without the further pre-training step on the test set of our NER dataset. We see that there is only a slight improvement in the performance when the model is further pretrained on Ottoman Turkish texts.

### 4.3 Ablation Study

In order to analyze this outcome more deeply, we performed an ablation study. Here, we further pre-trained the model by incrementally increasing the pre-training data, and at each stage, we fine-tuned it on the NER data to test its performance on this task. Figure 5 depicts the results of this study. We observe that the model reached its peak performance after training with around 50% of the pre-training corpus and started to decrease afterwards. Only at the end of the pre-training we see a significant increase in the performance. Table 2 shows the exact scores in this case.

| Models | Precision | Recall | F1 |
|---|---|---|---|
| **BERTurk** + PT | 0.820 | **0.9** | **0.858** |
| BERTurk | **0.829** | 0.872 | 0.850 |

Table 1: Performance of the models on the test split of the NER dataset.

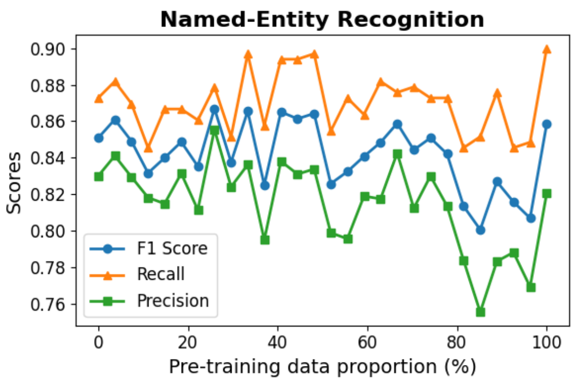


Figure 5: NER task performance as the pre-training data size grows

One possible reason for this mixed performance might be our current approach to the inline Arabic sentences or sentence parts in the corpus texts. At present, we exclude all Arabic texts from the main text as they are not relevant in a Turkish corpus. Yet, deleting them might lead to gaps in the meaning of text. We detect some cases in the corpus where omitting Arabic phrases embedded in Turkish sentences resulted in meaningless sentence parts in the text. We believe dealing with the Arabic parts of the corpus in a way that will not distort the context will result in a cleaner corpus and better pre-training performance.

One way of dealing with the Arabic parts in the sentences could be facilitating machine translation in reconstructing fragmented sentences and filling in missing alphanumeric characters as proposed in Section 3.3.3. However, the application of such models is deterred by the intensive computational resources required. Our preliminary analysis indicated that approximately 8% of our data contains Arabic sentences or sentence parts, with less than .9% of it is being completely unusable. Thus, although employing generative models to predict and reconstruct the semantics and pragmatics of corrupted Arabic sentences is highly beneficial, the costs associated with this approach are outweighed by the potential benefits, especially considering the significant yet smaller proportion of Arabic sentences compared to Turkish.

| Models | Precision | Recall | F1 |
|---|---|---|---|
| **BERTurk** **+ 50% PT** | **0.833** | 0.896 | **0.864** |
| BERTurk + 100% PT | 0.820 | **0.9** | 0.858 |
| BERTurk | 0.829 | 0.872 | 0.850 |

Table 2: Performance of BERTurk when further pretrained on the half of the corpus and on the whole corpus.

## 5 Conclusion

In this study, we presented the first foundations of a clean Ottoman Turkish text corpus. We discussed the challenges faced in extracting clean text data from documents with complex structures and explained our approaches in handling these challenges. By domain adapting a PLM initially created for modern Turkish to Ottoman Turkish using our corpus, we demonstrated the potential to bridge the gap between historical languages and modern NLP frameworks. The preliminary experimental findings highlight the effectiveness of our corpus in enhancing NER tagging for Ottoman Turkish, showcasing its utility for various NLP applications.

Our study takes one of the first steps towards providing comprehensive resources for the state-of-the-art natural language processing of Ottoman Turkish. Future research directions may involve expanding the corpus, refining preprocessing techniques, and exploring additional NLP tasks to further enrich the resources available in this historical language.

## Limitations

There are certain limitations of our study. Firstly, the reliance on periodicals from a specific time frame may introduce biases in the diversity of language usage and topics covered. Additionally, while efforts were made to ensure accuracy and completeness, there may exist inherent errors or omissions in the preprocessing step of the documents. Moreover, the current size of the corpus might be too small to properly pre-train a BERT model, so careful consideration should be given to the scalability and generalizability of the model.

## Ethics Statement

The source of text data used to create the corpus are periodicals published between 1908 and 1925. As these periodicals are publicly available and there are no copyright restrictions associated with them, we adhered to ethical guidelines in utilizing the data for research purposes. In addition, Ottoman Turkish is an extremely understudied language in natural language processing. So, there is no risk of exposure for our case.

## References

Sumeet Agarwal, Shantanu Godbole, Diwakar Punjani, and Shourya Roy. 2007. How much noise is too much: A study in automatic text classification. In *Seventh IEEE International Conference on Data Mining (ICDM 2007)*, pages 3–12. IEEE.

Jonathan P Bowen and Peter T Breuer. 1992. Occam's razor: The cutting edge of parser technology. *Proc. TOULOUSE*, 92.

Ömer Çakır. 2014. II. Meşrutiyet Dönemi'nde Sırat-ı Müstakîm ve Sebilürreşad dergilerine Türk dünyasından gönderilen bazı mektuplar. *Çankırı Karatekin Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, 5(1):237–246.

Abdullah Ceyhan. 1991. *Sırat-ı müstakim ve Sebilürreşad mecmuaları fihristi*, volume 55. Diyanet İşleri Başkanlığı Yayınları.

Eva D'hondt, Cyril Grouin, and Brigitte Grau. 2017. Generating a training corpus for ocr post-correction using encoder-decoder model. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1006–1014.

Maud Ehrmann, Matteo Romanello, Simon Clematide, Phillip Benjamin Ströbel, and Raphaël Barman. 2020. Language resources for historical newspapers: the impresso collection. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 958–968, Marseille, France. European Language Resources Association.

Abdullah Gündoğdu. 2008. Sırat-ı Müstakim (later, Sebilürreşad) and the origin of the Japanese image in Turkish intellectuals. *Annals of Japan Association for Middle East Studies*, 23(2):245–259.

Ahmed Hamdi, Axel Jean-Caurant, Nicolas Sidère, Mickaël Coustaty, and Antoine Doucet. 2020. Assessing and minimizing the impact of OCR quality on named entity recognition. In *Digital Libraries for Open Knowledge: 24th International Conference on Theory and Practice of Digital Libraries, TPDL 2020, Lyon, France, August 25–27, 2020, Proceedings 24*, pages 87–101. Springer.

Rakesh Jain et al. 2021. Rule-based and relationship-based extraction in network monitoring. *International Journal of Data Processing*, 18(3):121–139.

Lin Li, Tiong-Thye Goh, and Dawei Jin. 2020. How textual quality of online reviews affect classification performance: A case of deep learning sentiment analysis. *Neural Computing and Applications*, 32:4387–4415.

Zihan Liu, Genta Indra Winata, and Pascale Fung. 2021. Continual mixed-language pre-training for extremely low-resource neural machine translation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2706–2718, Online. Association for Computational Linguistics.

Daniel Lopresti and Zhou Jiangying. 1997. Using consensus sequence voting to correct ocr errors. *Computer Vision and Image Understanding*, 67(1):39–47. Cited on p. 34.

Kurt Micallef, Albert Gatt, Marc Tanti, Lonneke van der Plas, and Claudia Borg. 2022. Pre-training data quality and quantity for a low-resource language: New corpus and BERT models for Maltese. In *Proceedings of the Third Workshop on Deep Learning for Low-Resource Natural Language Processing*, pages 90–101, Hybrid. Association for Computational Linguistics.

Vinay Nundloll, Koichi Watanabe, and Alan Cohen. 2021. Automating information extraction: Case study of the Journal of Botany. *Journal of Heliyon*, pages 4–6.

Michael Piotrowski. 2012. *Natural Language Processing for Historical Texts*, volume 17 of *Synthesis Lectures on Human Language Technologies*. Morgan & Claypool.

Sarah Schulz and Jonas Kuhn. 2017. Multi-modular domain-tailored ocr post-correction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2716–2726.

Stefan Schweter. 2020. BERTurk - BERT models for Turkish.

Serge Sharoff. 2006. Creating general-purpose corpora using automated search engine queries. In Marco Baroni and Silvia Bernardini, editors, *Wacky! Working Papers on the Web as Corpus*. GEDIT, Bologna. Cited on p. 25.

Vésteinn Snæbjarnarson, Haukur Barri Símonarson, Pétur Orri Ragnarsson, Svanhvít Lilja Ingólfsdóttir, Haukur Jónsson, Vilhjalmur Thorsteinsson, and Hafsteinn Einarsson. 2022. A warm start and a clean crawled corpus - a recipe for good language models. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4356–4366, Marseille, France. European Language Resources Association.

Unicode Consortium. 2021. *The Unicode® Standard: Version 14.0 – Core Specification*, version 14.0 edition. Unicode, Inc., Mountain View, CA.