Tafsir Extractor: Text Preprocessing Pipeline preparing Classical Arabic Literature for Machine Learning Applications

Carl Kruse[1], **Sajawel Ahmed**[1,2]
[1]Goethe University Frankfurt, Germany
[2]University of California, Davis, United States
{carl.kruse, sahmed}@em.uni-frankfurt.de {sajahmed}@ucdavis.edu

Abstract

In this paper, we present a comprehensive tool of preprocessing Classical Arabic (CA) literature in the field of historical exegetical studies for machine learning (ML) applications. Most recent ML models require the training data to be in a specific format (e.g. *XML*, *TEI*, *CoNLL*) to use it afterwards for Natural Language Processing (NLP) tasks such as Named Entity Recognition (NER) or Topic Modeling (TM). We report on how our method works and can be applied by other researchers with similar endeavors. Thereby, the importance of this comprehensive tool of preprocessing is demonstrated, as this novel approach has no predecessors for CA yet. We achieve results that enable the training of current ML models leading to state-of-the-art performance for NER and TM on CA literature. We make our tool along its source code and data freely available for the NLP research community.

Keywords: preprocessing, named entity recognition, topic modeling, machine learning, historical NLP, classical Arabic, low-resource languages, theological studies

1. Introduction

While working within the field of Classical Arabic (CA) literature and its genre of exegetical studies (*tafsir*), it becomes clear that it is a very broad field in which different textual components can be examined along their topics, e.g. the textual component of oral traditions (*hadith*) along the topics of juridical rulings (*fiqh*), linguistics (*lugha*), and judeo-christian sources (*israiliyat*). One exegetical work that is particularly emphasized to research such topics is the monumental book of the theological scholar Al-Tabari (d. 923). His work *Tafsir Al-Tabari* is regarded among the most important exegeses in the Islamic theology and contains a large part of all relevant oral traditions that were in circulation at the beginning of the 10th century (Ahmed et al., 2022a). Through his work it is possible to gain insights into the mentioned topics (e.g. juridical rulings) to understand a given verse and its circumstances for scholars of historical literature.

Given the substantial volume of Al-Tabari's work, extracting and compiling oral traditions on specific topics from classical works to enhance Quranic explanations is a complex task. This complexity is exacerbated by the vast array of topics available for analysis in exegetical works, totaling 15 according to the classical categorization by Al-Suyuti (1505). Therefore, it becomes imperative to undertake digital preparations for such texts. This digital transformation allows efficient access to a wealth of information within the realm of exegesis, facilitating a more effective exploration of diverse topics.

To this end, we develop in this work the tool *Tafsir Extractor*, a comprehensive text preprocessing pipeline for preparing gold data that can be used to train machine learning (ML) models. Several steps are necessary to digitally process the Al-Tabari corpus. First, the corpus is extracted from the resource platform *Gawami' al-Kalim*[1] (GK), which is prepared via an optical character recognition (OCR) process from the original manuscripts. Second, it is annotated manually in the XML/TEI formats according to the annotation guidelines (Ahmed et al., 2022b). Third, the data is converted using our TEI2CoNLL module, a method that has been developed with various different options to convert the data automatically into the CoNLL format (Tjong Kim Sang and De Meulder, 2003) which is necessary for current ML models (Lample et al., 2016; Devlin et al., 2019; Brown et al., 2020) for the task of Named Entity Recognition (NER). Our extended CoNLL

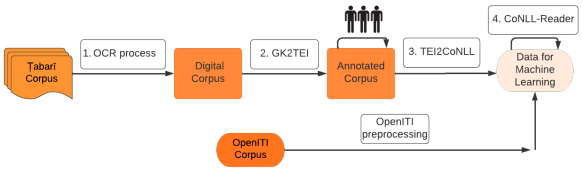

Figure 1: Processing steps for preparing the raw data from CA literature

format also contains a matrix with extracted information on Topic Modeling (TM) data. Additional columns can be added for syntactical and analytical information of a word or a sentence. Finally, this whole dataset can be used for ML evaluations (see Figure 1), either by training the heavyweight large language models with *MaChAmP* (van der

[1]https://gk.islamweb.net

OSACT 2024 Workshop, pages 67–73

Goot et al., 2021) (even in multi-task learning scenarios), or by lightweight embedding-based models (Mikolov et al., 2013; Ling et al., 2015) trained on the unlabeled OpenITI corpus (Miller et al., 2018) from scratch.

Thus, our work allows us to accelerate the process of digitizing information from the historical literature of theological studies. We lay the technical foundations for our ongoing research work on Natural Language Processing (NLP) for CA literature allowing the training and fine-tuning of current ML models for higher-level NLP tasks such as NER or TM (Ahmed et al., 2022b). Until now there was no software tool available that solved these respective tasks for CA. The tool developed by us is freely available along its data[2], so that the reproducible and further development of both the methodology and the results is enabled for the open source community.

2. Related Work

Various approaches have been utilized to create software tools that address preprocessing challenges in different languages. These tools enable the faster analysis of substantial amounts of text data, even for historical low-resource languages and their literature.

Recent research in Arabic NLP has produced new tools that provide different functions for text, sentence, word, pre- and suffix analysis, e.g. *CAMeL* tools (Obeid et al., 2020), *Stanza* (Qi et al., 2020), *MADAMIRA* (Pasha et al., 2014) and *Farasa* (Abdelali et al., 2016). CAMeL tools is a comprehensive NLP package specialized for Arabic language. Stanza offers various preprocessing methods for many languages including Arabic. MADAMIRA and Farasa are tools for Morphological Analysis and Disambiguation of Arabic. These tools mainly focusing on Modern Standard Arabic, rather than CA. Besides, these prior works cannot convert out-of-the-box a given piece of text in XML/TEI format into a specific format (e.g. CoNLL). In our research work, this target format is needed for ML downstream-task evaluations. Even with a combination of existing tools just mentioned above, the target format cannot be reached. Therefore, we solve this challenge by developing a comprehensive modular pipeline which, once started, automatically solves the required tasks.

Preprocessing Arabic, especially Classical Arabic, is challenging due to its complex morphology. Tasks include word evaluation, categorizing sentence components, and segmenting sentences. Analyzing words and parsing sentence components for specific topics or subtopics is challenging due to contextual dependencies. Context-aware strategies are needed to prevent misinterpretations, as words can have multiple meanings depending on context, such as "madina" meaning both a city and a personal name, and "mansur" meaning assistance or a personal name.

This field is relatively new and specialized. Our literature review revealed that to the best of our knowledge, there are no prior studies available. Hence, we are among the first to deal with this genre of literature from the perspective of computational linguistics. Despite that, we discovered that existing tools are helpful for certain aspects in our pipeline, while working on preparing ML training data.

To accomplish the tasks just mentioned and to meet the specific requirements of our research, we developed in this paper our own functions and methods, which are not available in any previous work, and introduced our very specific approach for textual preprocessing of CA literature for ML applications.

3. Preprocessing Approach

As previously illustrated, our preprocessing pipeline comprises four distinct stages (see Figure 1). We provide details for each module and put our emphasis on the most complex part, namely the *TEI2CoNLL* module.

OCR process The initial step is defining and extracting CA literature data through an *OCR process* which involves in our case the digitization of the Al-Tabari corpus. This is a foundational step of transformation in our preprocessing pipeline, which allows us to take any raw OCR text from the vast collection of CA literature and digitize it so that it can be used for further processing.

GK2TEI Afterwards our *GK2TEI module* diligently transforms the digitized data of the *OCR process* available from its very specific markup language into XML files applying the TEI format, leveraging a myriad of functions we embedded within this module. This format enables the structured coding and annotation of the data. Consequently, the data is ready to be used for manual annotation and further processing by our tool *TEI2CoNLL*.

TEI2CoNLL The digitization process of the Tabari text into XML format includes the annotation of NER, topics, and subtopics by experts in the field of theological studies. The annotation of the text data is carried out manually whereby the rest of the transformations are automated by our pipeline. Afterwards, the data is exported into XML files, serving as the base for *TEI2CoNLL*, which is the core

[2] https://github.com/sa-j/ArabicNLP

of our processing pipeline. The program provides versatile filtering options for generating specific outputs, including choices for NEs, topics, subtopics, either with or without *Isnad* (chain of transmitters). Users can customize data extraction using flags, and the order of functions can be adjusted, offering flexibility in data processing. The resulting output is presented in a specific matrix format (see Figure 2).

TEI2CoNLL reads preprocessed XML files, merges them into one, and extracts NEs, topics, subtopics, and Isnads. It then converts the merged file into CoNLL format, crucial for data analysis and training ML models. In these XML files, annotated data is identified using specific keys such as ‹*persName*› for *Isnad*, ‹*name ..*› for NEs, ‹*p..ana=..*› for topics, ‹*seg ana=..*› for subtopics, and ‹*said*› for the *Matn* (text of a transmission). The XML files are processed, creating a three-column matrix shown in Figure 2. Very long sentences are split based on heuristics considering factors like length and specific factors, as described in detail in Schweter and Ahmed (2019). Users can customize the inclusion NEs and Isnad into the sentences. NE tags are converted to the BIO scheme, marking start with B-[NE] and subsequent ones with I-[NE].

```
# adyan: 1
# asbab: 0
# fiqh: 0
# kalam: 1
# lugha: 1
# mushkilat: 0
# mutashabih: 0
# naskh: 0
# qiraat: 0
# science: 1
# sirah: 0
# sufism: 1
# takhsis: 0
# tikrar: 0
# israeliyat: 1
الْجِنِّ    O   B-kalam
وَهُوَ  O   O
إِبْلِيسُ   B-OTH   B-kalam
،   O   O
وَالْأَخَرُ   O   O
فِرْعَوْنُ  B-PER   O
.   O   O
قَالَ   O   O
:   O   O
أَنَا   O   B-kalam
رَبُّكُمُ   O   I-kalam
الْأَعْلَى   O   I-kalam
.   O   O
```

Figure 2: Matrix representation of NEs, topics, and subtopics

CoNLL-Reader In the final phase of the pipeline, our *CoNLL-Reader* module empowers the modification of pre-existing ML training datasets, already strictly structured in CoNLL format. This flexibility enables the application of advanced grammatical, morphological, and script-dependent preprocessing techniques, thereby enhancing the depth of analysis for historical languages and their ancient writing systems. Our *CoNLL-Reader* module has specific filtering options to modify punctuation, diacritic marks (*tashkeel*) and even letters in the CoNLL files. This allows us to customize the CoNLL files to generate different preprocessed versions of ML training data, allowing us to develop our novel method of *script-compression* as part of our ongoing work on NLP for the CA language.

Preparation of data for language model training: OpenITI extraction Beside *TEI2CoNLL*, we analogously apply a specific preprocessing technique on the OpenITI corpus in order to extract data for training the language model from scratch. To get CA text data, we crawl the platform of *OpenITI*, which contains one of the largest collections of online available historical books for CA. The final data is stored in one large text data file in which per line one sentence is saved. To actually generate this format, we apply our sentence splitting heuristics along tokenization from CAMeL tools. This additional data helps us to train a lightweight model with state-of-the-art performance for NER or related tasks without relying on pre-trained language models.

4. Results

The preprocessing pipeline *Tafsir Extractor* produces text data for different stages of our ML analysis. In the following sections, we present the major results after the *Tafsir Extractor* has been applied on the input data set consisting of the entire Tabari corpus.

GK2TEI: data for human annotation The *GK2TEI module* standardizes the raw CA text from its very specific markup language by automatically generating the TEI files. This allows the usage of various tools which are based on the popular TEI format, such as the *Oxygen XML Editor*[3]. Hence,

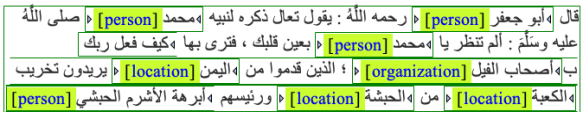

Figure 3: Screenshot of the annotation working environment in *Oxygen XML Editor* (figure taken from Ahmed et al. (2022b)).

this crucial step of conversion generates the data which enables the manual annotation of raw CA texts with NEs and Topics by experts and its further analysis by ML models. Figure 3 provides a view of the annotation environment.

[3] https://www.oxygenxml.com/

TEI2CoNLL: data for task-specific ML training (NER and TM) The output in Figure 2 presents a matrix displaying sentences with listed topics. Each sentence begins with topics marked as 1 or 0. Untagged sentences are denoted with 0, and undefined topics as 'nil.' Subtopics follow a BIO format akin to NE tags. Data extraction includes NEs, sentence-based topics, and span-based subtopics. Extracting NE tags involves boundary recognition and categorization into semantic types: persons (PER), organizations (ORG), locations (LOC), times (TME), and others (OTH), leveraging annotation data for concept analysis in theology. Sentence-based topics, in total 15, encompass a range of categories including topics like topics of juridical rulings (*fiqh*), theological topics (*kalam*), and linguistics (*lugha*). Span-based subtopics further refine these topics and include themes like specific historical topics (e.g. *tareekh*). Finally, the processed data is saved into three files (dev.conll, test.conll, train.conll).

Table 1 provides the results for NEs. We can see that there are twice as many NE tokens in the data with Isnad compared to the data without Isnad. Especially for the NE category PER, the amount is increased significantly, but not for the other four NE categories. This is not surprising, since by definition an Isnad consists predominantly of transmitters (i.e. PER). Therefore, the inclusion of Isnad has a greater impact on the number of PER tokens, rather than any other NE category and their tokens. Furthermore, according to our results presented in Table 1, the total count of tokens is 1,793,315. However, when Isnad is excluded from the calculation, the count drops to 913,749 tokens. This suggests that approximately half of the text consists of Isnad data.

NE w. Isnad	NE w.o. Isnad
1,409,334 O	775,010 O
176,105 B-PER	47,746 B-PER
149,292 I-PER	31,991 I-PER
22,026 B-ORG	21,459 B-ORG
12,453 B-OTH	12,142 B-OTH
8,456 I-ORG	8,122 I-ORG
4,160 B-TME	6,610 B-TME
5,583 B-LOC	4,990 B-LOC
4,087 I-TME	3,912 I-TME
1,032 I-OTH	1,008 I-OTH
787 I-LOC	759 I-LOC

Table 1: Results for NE tokens with/without Isnad

For topics, the picture is more dynamic while looking from the perspective of Isnad inclusion (see Table 2). For some topics (fiqh, sufism, adyan) the number depends highly on the Isnad inclusion, whereas for some other topics (qiraat, tikrar, takhsis) the number seems to be not strongly influenced by this inclusion. Further investigation is required to determine the reason for this pattern.

Topic	w. Isnad	w.o. Isnad
adyan (non-Islamic relig.)	31,931	20,536
asbab (occas. of revelation)	9,143	6,268
fiqh (jurisprudence)	21,381	9,753
israiliyat (Judeo-Christian)	7,795	4,533
kalam (Islamic theology)	36,133	19,384
lugha (linguistics)	29,573	15,776
mushkilat (problem)	59	28
mutashabih (allegorical)	360	175
naskh (abrogation)	1,257	727
qiraat (recitation style)	4,957	4,238
sirah (prophetic biography)	3,960	2,729
sufism (mysticism)	15,570	7,553
takhsis (specification)	400	317
tikrar (repetition)	405	381
ulum (science)	5,028	2,262

Table 2: Results for Topic tokens with/without Isnad

OpenITI: data for language model training (task-independent) Our results for the OpenITI corpus data are 134.17 Mio. sentences, extracted from 17 GB of raw text data, which is the largest amount yet to be used for CA. Thus, this allows the training of lightweight ML models for CA-NER and CA-TM without relying on pre-trained language models which are not made with regard to the domain of historical theology. We plan to upload this corpus data along its text generation module for the research community. This will give rise to the possibility of using the strengths of current heavyweight ML models (such as BERT, XLNet, GPT-3 (Devlin et al., 2019; Yang et al., 2019; Brown et al., 2020)) and training domain-specific versions of them as well, even when new historical text collections are added to the growing platform of OpenITI.

5. Conclusion

In this paper, we introduced the *Tafsir Extractor*, a comprehensive preprocessing tool designed for extracting raw text data from CA literature and converting it into a specific format (e.g. CoNLL and its extensions), to facilitate downstream-task evaluations for fundamental NLP tasks, such as NER and TM. The absence of a similar tools for CA literature prior to our work prompted the development of *Tafsir Extractor*. Consequently, our work paves the way for a large-scaled generation and analysis of historical CA literature with modern ML methods.

Our work highlights the challenge of sentence segmentation and word recognition in CA texts due to the absence of punctuation and the context-dependant changes in the semantics of words. To overcome these challenges, we have employed a specialized heuristics in our program, which considers word counts, customizable through a filter in our program, and takes into account sub/topic annotatations for segmentation. Determining contextual meanings of words still poses a formidable challenge for NLP methods in prospective projects. Despite minor deviations, the cleanliness of the data

enables its utilization for subsequent downstream-task evaluations without hindrance.

In future work, we propose improving sentence segmentation by developing a domain-specific neural network model which identifies sentence boundaries based on semantics rather than syntax of text. This approach holds promise for addressing the major limitations encountered in our current work.

Acknowledgments

This interdisciplinary work has been conducted as part of the research on NLP for Classical Arabic literature[4]. This work was supported by a fellowship of the German Academic Exchange Service (DAAD). Special thanks go to Prof. M. Syed (University of California, Davis) and Prof. G. Roig (Goethe University Frankfurt) for the resources made available for conducting the research presented in this paper.

6. Bibliographical References

Ahmed Abdelali, Kareem Darwish, Nadir Durrani, and Hamdy Mubarak. 2016. Farasa: A fast and furious segmenter for Arabic. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 11–16, San Diego, California. Association for Computational Linguistics.

Sajawel Ahmed, Misbahur Rehman, Joshua Tischlik, Carl Kruse, Edin Mahmutovic, and Ömer Özsoy. 2022a. Linked Open Tafsir—Rekonstruktion der Entstehungsdynamik (en) des Korans mithilfe der Netzwerkmodellierung früher islamischer Überlieferungen. In *8. Jahrestagung des Verbandes Digital Humanities im deutschsprachigen Raum (DHd)*.

Sajawel Ahmed, Rob van der Goot, Misbahur Rehman, Carl Kruse, Ömer Özsoy, Alexander Mehler, and Gemma Roig. 2022b. Tafsir dataset: A novel multi-task benchmark for named entity recognition and topic modeling in classical Arabic literature. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3753–3768, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Jalal Al-Din Al-Suyuti. 1505. *Al-itqan Fi 'ulum Al-Qur'an (The Perfect Guide to the Sciences of the Qu'ran)*. Garnet Publishing; Bilingual edition (May 1, 2012).

[4] https://tafsirtabari.com/about

Emily M Bender and Batya Friedman. 2018. Data statements for natural language processing: Toward mitigating system bias and enabling better science. *Transactions of the Association for Computational Linguistics*, 6:587–604.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901, NA. Curran Associates, Inc.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural Architectures for Named Entity Recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270. Association for Computational Linguistics.

Wang Ling, Chris Dyer, Alan Black, and Isabel Trancoso. 2015. Two/Too Simple Adaptations of word2vec for Syntax Problems. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Matthew Thomas Miller, Maxim G. Romanov, and Sarah Bowen Savant. 2018. Digitizing the textual heritage of the premodern islamicate world: Principles and plans. *International Journal of Middle East Studies*, 50(1):103–109.

Ossama Obeid, Nasser Zalmout, Salam Khalifa, Dima Taji, Mai Oudah, Bashar Alhafni, Go Inoue, Fadhl Eryani, Alexander Erdmann, and Nizar Habash. 2020. CAMeL tools: An open source python toolkit for Arabic natural language processing. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 7022–7032, Marseille, France. European Language Resources Association.

Arfath Pasha, Mohamed Al-Badrashiny, Mona Diab, Ahmed El Kholy, Ramy Eskander, Nizar Habash, Manoj Pooleery, Owen Rambow, and Ryan Roth. 2014. MADAMIRA: A fast, comprehensive tool for morphological analysis and disambiguation of Arabic. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1094–1101, Reykjavik, Iceland. European Language Resources Association (ELRA).

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Stefan Schweter and Sajawel Ahmed. 2019. Deep-EOS: General-Purpose Neural Networks for Sentence Boundary Detection. In *Proceedings of the 15th Conference on Natural Language Processing (KONVENS)*.

Abdulhadi Shoufan and Sumaya Alameri. 2015. Natural language processing for dialectical Arabic: A survey. In *Proceedings of the Second Workshop on Arabic Natural Language Processing*, pages 36–48, Beijing, China. Association for Computational Linguistics.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Rob van der Goot, Ahmet Üstün, Alan Ramponi, Ibrahim Sharaf, and Barbara Plank. 2021. Massive choice, ample tasks (MaChAmp): A toolkit for multi-task learning in NLP. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 176–197, Online. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, volume 32, NA. Curran Associates, Inc.

A. Data and Code Availability Statement

The project aims to promote cooperation and progress in the field of NLP. To ensure transparency and reproducibility, all datasets used in our experiments, along with the corresponding codebase, will be made readily available to the public through GitHub repositories (https://github.com/sa-j/ArabicNLP). The datasets will be provided in commonly used formats, accompanied by comprehensive documentation detailing their sources, preprocessing procedures, and any relevant licensing information. The codebase will be structured in a modular and well-documented manner. The aim is to offer researchers precise instructions for accessing and using the data, which will facilitate their understanding, extension, and adaptation of our algorithms and methodologies. The NLP community is encouraged to explore, critique, and build upon the contributions, promoting a culture of open collaboration and accelerating progress in the field.