TULAP - An Accessible and Sustainable Platform for Turkish Natural Language Processing Resources

Susan Üsküdarlı
suzan.uskudarli@boun.edu.tr

Muhammet Şen
muhammet.sen@boun.edu.tr

Furkan Akkurt
furkan.akkurt@boun.edu.tr

Merve Gürbüz
merve.gurbuz@boun.edu.tr

Onur Güngör
onurgu@boun.edu.tr

Arzucan Özgür
arzucan.ozgur@boun.edu.tr

Tunga Güngör
gungort@boun.edu.tr

Abstract

Access to natural language processing resources is essential for their continuous improvement. This can be especially challenging in educational institutions where the software development effort required to package and release research outcomes may be overwhelming and under-recognized. Access to well-prepared and reliable research outcomes is important both for their developers as well as the greater research community. This paper presents an approach to address this concern with two main goals: (1) to create an open-source easily deployable platform where resources can be easily shared and explored, and (2) to use this platform to publish open-source Turkish NLP resources (datasets and tools) created by a research lab. The Turkish Natural Language Processing (TULAP) was designed and developed as an easy-to-use platform to share dataset and tool resources which supports interactive tool demos. Numerous open access Turkish NLP resources have been shared on TULAP. All tools are containerized to support portability for custom use. This paper describes the design, implementation, and deployment of TULAP with use cases (available at https://tulap.cmpe.boun.edu.tr/). A short video demonstrating our system is available at https://figshare.com/articles/media/TULAP_Demo/22179047.

1 Introduction and Motivation

There is a growing interest in the field of natural language processing (NLP) due to the recent advances in deep learning-based approaches, such as transformer architectures and pretrained language models. This interest shows itself in the demand both for applications solving NLP tasks and for high-quality corpora that can be used in machine learning models. Although there are plenty of such resources for well-studied languages such as English, there is a scarcity of these resources in most of the other languages. In addition to developing such resources, there is a need for building software environments that collect these resources within a platform and offer them to the NLP community with easy-to-use interfaces.

This paper introduces TULAP (Turkish Language Processing Platform) that aims to provide a variety of Turkish NLP resources (datasets and tools). TULAP provides a user-friendly user interface (UI) where users can easily discover and examine NLP resources. Tools can be explored without the hassle of having to build and install them. All tools are containerized APIs (Application Programming Interface) to facilitate easy installation and use. The tools served on TULAP are provided as open-source to facilitate further research and development. This is particularly important in educational institutions where students tend not to package their code appropriately, leading to to loss of time and effort for those who wish to use their work.

The main motivation of TULAP is to develop a platform that supports the continuous contribution of NLP resources generated by our research lab, TABILab[1]. For the initial deployment, we gathered previously developed resources, created APIs, and containerized them. Hereafter, the resource developers will be able to make their contributions with an easy-to-use interface. The aim is to provide an up-to-date NLP platform with resources that are easy to discover and explore as well as to contribute new resources produced by researchers.

The main contributions of this work are summarized as follows:

- The creation of an NLP platform (TULAP)[2] that supports access to Turkish NLP datasets and tools, provides interactive demos, enables easy end-user contribution of datasets and tools, and open access to all resources;

[1] Text Analytics and BioInformatics Lab (TABILab) is a research lab of the Computer Engineering Department of Boğaziçi University. https://tabilab.cmpe.boun.edu.tr/

[2] https://tulap.cmpe.boun.edu.tr/

Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics System Demonstrations, pages 219–227
May 2-4, 2023 ©2023 Association for Computational Linguistics

- Systematic documentation of the Turkish NLP datasets and tools with access information [3];
- Creation of APIs and containerized versions of the tools to facilitate accessibility;
- Introduction of a general archival process to support continuous contributions by developing a research output platform[4];
- A monitoring system to identify problems and track how the platform is being used, and
- A deployment that aggregates Turkish NLP resources developed at Boğaziçi University.

The remainder of this paper is organized as follows: Section 2 provides an overview of NLP platforms; Section 3 describes the platform requirements, design, and implementation; Section 4 describes the resources on TULAP with use-cases in Section 5. Finally, Section 6 discusses the current state and future work with concluding remarks.

2 Related Work

The goal of collecting NLP resources under a unified framework and thus making implementing various types of applications easier has led to the development of NLP platforms in several languages. In this section, we briefly review a few of the widely-used NLP platforms with different functionalities including those specific to the Turkish language.

NLTK (Bird, 2006) is one of the earliest NLP libraries consisting of tools as a hierarchy of modules. It supports a wide range of tasks as well as corpora that can be used in various NLP tasks. AllenNLP (Gardner et al., 2018) is a library for deep learning-based NLP research. The users can build their own models using common deep learning operations. Stanford CoreNLP toolkit (Manning et al., 2014) provides a pipeline of preprocessing operations and downstream NLP tasks in various languages. It is one of the most widely used NLP platforms partly due to its simple design. Users can also integrate new NLP tasks to the pipeline.

The Hugging Face platform (Hugging Face, Inc., 2023) is one of the most popular platforms that is actively used in current NLP research. It includes a wide range of libraries that consist of different types of software such as machine learning models, datasets, demos, and evaluation tools. The `transformers` library (Wolf et al., 2020) includes open-source implementations of state-of-the-art transformer-based models. The users can easily share the resources they have developed in the platform via the Hugging Face Hub.

Among the platforms specific to Turkish, İTÜ Turkish NLP Web Service (Eryiğit, 2014) offers a variety of tools that can be used in Turkish NLP studies. The tools in the platform can be used either as a pipeline or as stand-alone components. Since the platform has been designed for Turkish, it includes tools that are specific to Turkish such as diacritic restorer. A recently-developed platform is Mukayese (Safaya et al., 2022), which is a benchmarking platform that provides a set of datasets and benchmarks for seven different types of Turkish NLP tasks. The Mukayese platform is also a part of the Turkish Data Depository (TDD) project[5] for building a repository of Turkish NLP resources.

The platform that we introduce in this work, TULAP, bears commonalities with these earlier works as well as different functionalities. Unlike most other platforms, we offer both NLP datasets and software tools. The tools can be used as ready-to-use stand-alone applications, while the datasets can be utilized for training machine learning models or for other purposes. With respect to the tool set, TULAP is similar to the Stanford CoreNLP framework in the sense that both preprocessing and downstream operations are supported, and the platform is extendable with new tools. There are several differences of TULAP from the currently available Turkish platforms. While these platforms mostly provide preprocessing operations, TULAP also supports a wide range of more complex tasks like text summarization and question answering. Furthermore, all the resources are made accessible with open-source or Creative Commons licenses and all tools are containerized for portability purposes. Finally, the platform is extensible with support for end-user contribution of new tools and datasets.

3 Platform Design & Implementation

3.1 Requirements

The main goals of the platform are to: (1) provide information and access to NLP resources (datasets and tools); (2) demonstrate the use of tools; (3) facilitate acquisition and use of resources; and (4) contribute new resources. There are three types of users: end users, resource providers, and system administrators. The main requirements are:

[3] https://github.com/BOUN-TABILab-TULAP

[4] https://github.com/BOUN-TABILab-TULAP/tabi-rop

[5] https://tdd.ai/?language=en

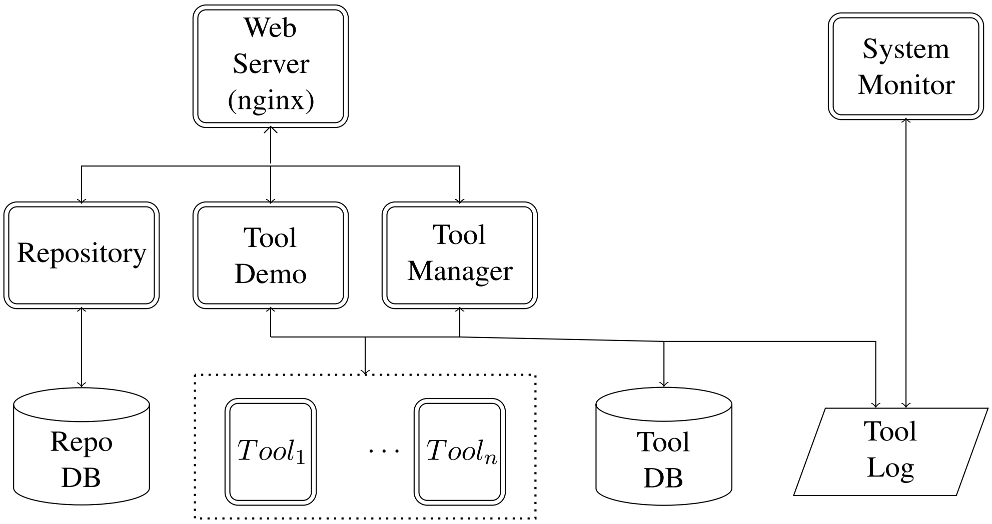


Figure 1: An overview of TULAP. The containers (docker) are indicated with double borders.

- A user should be able to browse and search the resources by keywords, tags, and authors.
- Resource providers should be able to specify or modify information (authors, language, etc.) related to the resources they contribute.
- The system should provide a demo interface that processes the input and returns the results.
- The system should provide installation and tool usage instructions.
- Containerization should be used to support portability, executability and scalability.
- All resources should be easy to obtain and use to support accessibility and extensibility.
- System administrators should be able to control and monitor the use of the tools.

3.2 Design and Implementation

The design of TULAP addresses the description, discovery and access to NLP-related dataset and tool resources. To support the portability and ease of building we chose to implement the platform with containerized components using Docker (Boettiger, 2015). Furthermore, all tools that are hosted on the platform must also be containerized. Thus, the platform as well as the tools it hosts are easily reusable.

The TULAP platform consists of several containers (Figure 1): An Nginx (NGINX, 2022) container serves as a reverse proxy for TULAP services and static files. There are two main functions that TULAP serves: (1) the specification and the access to information regarding NLP-related resources, and (2) demonstration of how tools function. The first part is addressed with the *Repository*, based on CLARIN-DSPACE (UFAL, 2022) (Common Language Resources and Technology Infrastructure) which is widely used for repositories. Users may browse and search about structured information related to datasets and tools, including references to corresponding academic articles. A PostgreSQL (PostgreSQL, 2022) container is used to persist data related to these resources.

Tool-related functionality is handled with the *Tool Manager* and *Tool Demo* containers. The *Tool Manager* supports the addition of new tool demos which involves specifying project source code (dockerized), information about the input & output types, and a user guide (see Section 5). *Tool Demo* enables the interactive exploration of tool demos using predefined or user-provided input. Figure 2 shows a sequence diagram of how a user interacts with a specific tool demo. First, the tool specification is fetched to generate a user interface (form). The input provided by the user is used to generate an API request which is sent to the tool. The response is processed according to the output specification and presented to the user.

Tools must be open-source, dockerized, and provide an application programming interface (API). Tool specifications are used to generate user interfaces to support interaction with demos. The front-end of tool demos is handled with a Node container[6] and the back-end (API calls) is handled with a Python container[7]. A MongoDB container[8] stores tool specifications and their usage information. The aim to provide ease of access and use to

[6] https://hub.docker.com/_/node
[7] https://hub.docker.com/_/python
[8] https://hub.docker.com/_/mongo

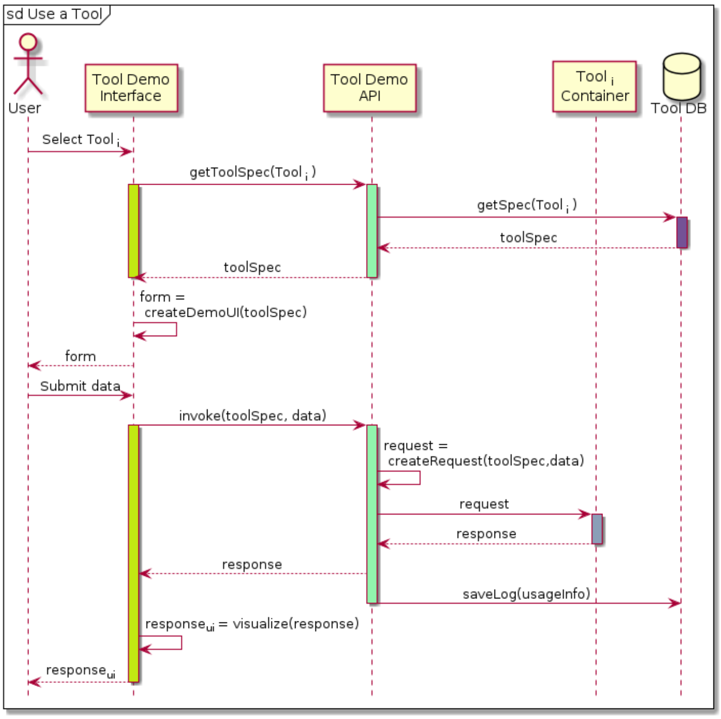


Figure 2: The sequence diagram for using a specific tool.

tools is the reason for imposing these criteria. As tools are dockerized, they can be fetched, built, and used for custom purposes.

Finally, the *System Monitor* allows the system administrators to gain insight into which, when, and how tools have been accessed. The system health is tracked to ensure that tools are up and running and to mitigate any problems.

3.3 Adding New Tools to the Demo Platform

Tool developers and system administrators are authorized to add new tools to the platform. Tools can be contributed by specifying four kinds of information: (1) *general* information which includes name & description (in Turkish and English), Git repository link, contact email, and API endpoint; (2) *input* specification which includes a name and type for each field type, at least one sample input, and the corresponding parameter name used by the tool; (3) *output* specification which includes the name and type of fields in the response provided by the tool; and (4) *user guide* which describes the input and output expected by the tool demo. This tool specification is posted to the *Tool Demo* which clones the Git repository and builds a Docker image. Then it creates a container from the image and stores the container details for future invocations. Once the container starts, it can handle requests.

4 TULAP Deployment

At the time of the writing of this article, TULAP includes 12 datasets and 16 tools which are shown in Table 1 and Table 2, respectively.

NLP Datasets
BOUN Dependency Treebank **v2.8** (Türk et al., 2019; Türk et al., 2021); **v2.11** (Marşan et al., 2022; Marşan, 2022)
Question Answering Corpus (Derici et al., 2018b,a)
Scientific Abstracts Corpus (Öztürk et al., 2014b,a)
Sentiment Analysis Corpus (Köksal and Özgür, 2021c,b)
Sign Language Corpus (Buz and Güngör, 2019a,b)
Türkiye Büyük Millet Meclisi (Grand National Assembly of Turkey) Corpus (Güngör et al., 2018a,b)
Turkish-English Parallel Corpus (Taşçı et al., 2006a,b)
Turkish Multi-document Summarization Corpus (Nuzumlalı and Özgür, 2014a,b)
Turkish Question Answering Dataset (SQuAD-TR) (Budur et al., 2020)
Web Corpus (Sak et al., 2011, 2010)
Word Embeddings (Güngör and Yıldız, 2017a,b)

Table 1: The datasets provided on TULAP

The tools on the platform are dynamically updated with new contributions. The platform itself is open-source and accessible at: https://github.com/BOUN-TABILab-TULAP/tabi-rop.

5 Using TULAP

This section describes how TULAP can be used through use cases. The landing page of TULAP

NLP Tools
Tokenizer (Ak and Güngör, 2022c)
Sentence Splitter (Ak and Güngör, 2022b)
Deasciifier (Ak and Güngör, 2022a)
Lemmatizer (Köksal, 2018)
Morphological Analyzer (Sak et al., 2007a,b)
Morphological Disambiguator (Sak et al., 2007a,c)
Dependency Parser (Özateş et al., 2018, 2020)
Verbal Multiword Expression Identifier (Yirmibeşoğlu and Güngör, 2020a,b)
Named Entity Recognizer (Güngör et al., 2018c,d)
Question Answerer (Derici et al., 2018b,c)
Relation Extractor (Köksal and Özgür, 2020, 2021a)
Sentiment Analyzer. i) Binary (Aydın and Güngör, 2020; Aydın et al., 2021); ii) Ternary (Köksal and Özgür, 2021c,b)
Text Summarizer (Baykara and Güngör, 2022b,a)
Grammar Annotation Tool. i) Standalone (Türk et al., 2022; Berk and Köksal, 2021); ii) Web (Akkurt et al., 2022; Akkurt and Uskudarli, 2022)

Table 2: The tools provided on TULAP

allows users to browse and search for datasets and tools. Dataset resources can be easily searched, inspected, and downloaded. Due to space limitations, in this section we focus on describing tool handling as it is significantly more complicated. The following use cases demonstrate how users: (1) discover and inspect tools; (2) contribute new resources; (3) fetch, build, and use tools independently; and (4) monitor the use of TULAP:

(1) Tool Discovery and Demo Figure 3a shows the repository where users can browse and search for resources. The user has searched for "named entity" for which resources that include either keyword are shown. Figure 3b shows the details pages when the "Named Entity Recognition" tool is selected. Figure 3c [9] shows the use of the demo with the Turkish sentence *İstanbul Barosu'ndaki Yapay Zekâ, Robotlar ve Hukuk Konferansı'nda pirimiz Alan Turing'i anmadan olmazdı*[10]. The tool returns the entity tags for tokens, for which the demo provides two alternative presentations (BRAT (Stenetorp et al., 2012) and JSON).

(2) Adding new tools Adding a new tool to the repository consists of providing information about the source-code (Git) and executable demo (URL) of the tool as well as academic information such as related papers, authors, and funding.

(3) Executing tools TULAP provides all the resources related to the tools along with an interface to demonstrate their functionality. All the tools provided in TULAP are open-source and are also independently available at TULAP repository[11]. For those who wish to utilize a tool beyond the demo interface (e.g. to recognize named entities in a large dataset using the NER tool), we have dockerized their APIs for easy deployment. TULAP itself utilizes the dockerized tools along with an interface we generate to demonstrate their functionality. Anyone can easily build the dockerized tools in their environment to use the APIs to their specific purposes. Listing 1 shows how easily the *Named Entity Recognition* tool can be built and used. Note that the API call is the one for the example shown in Figure 3c.

(4) Monitoring Tool Demos System administrators use the *System Monitor* to observe metrics like the number of requests and response times for the tools. These metrics are collected and visualized using Grafana (Grafana, 2022).

6 Discussion and Conclusions

This work's primary focus was to create an accessible and sustainable platform for Turkish NLP resources produced at Boğaziçi University. For this, we developed a *research output platform* (*tabi-rop*) where open access to datasets and tools that result from research activities related to NLP or other fields can be shared. TULAP was developed using *tabi-rop* and hosts numerous resources produced by our research lab. One of the most challenging tasks was the collection and packaging of previously developed resources. This fact validated the need for such a platform – a sentiment also expressed by the research lab alumni when reviewing TULAP. We continuously monitor TULAP as well as maintain *tabi-rop*[12]. Currently, we are upgrading the *Repository* component to benefit from DSpace v7 (Lyrasis, 2022) improvements.

Moving forward, we plan to include preprocessing tasks, new tasks, and improved versions of present tools. We are pleased that new resources have been contributed subsequently to the initial release. Other future work includes platform maintenance and improvements to the presentation of and interaction with resources.

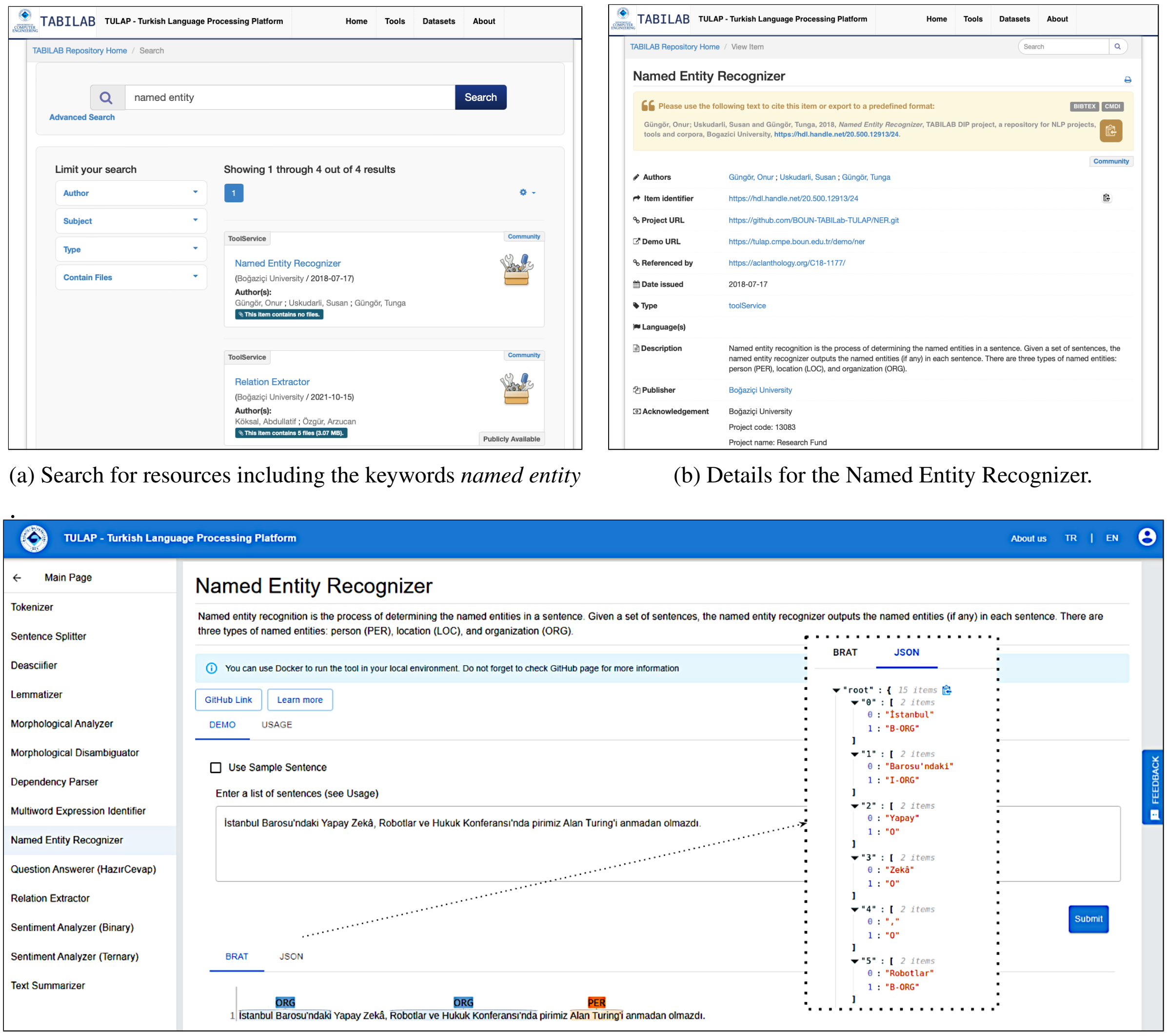
While many platforms have been emerging to

[9] For higher resolution of the images see https://figshare.com/articles/figure/TULAP_NER/21629549

[10] Translation: *It is impossible not to remember our sage Alan Turing during the Artificial Intelligence, Robots, and Law Conference held at the Bar Association of Istanbul.*

[11] URL: https://github.com/BOUN-TABILab-TULAP

[12] https://github.com/BOUN-TABILab-TULAP/tabi-rop



named entity

Showing 1 through 4 out of 4 results

Named Entity Recognizer
(Boğaziçi University / 2018-07-17)
Author(s):
Güngör, Onur ; Uskudarli, Susan ; Güngör, Tunga
This item contains no files.

Relation Extractor
(Boğaziçi University / 2021-10-15)
Author(s):
Köksal, Abdullatif ; Özgür, Arzucan
This item contains 5 files (3.07 MB).

(a) Search for resources including the keywords *named entity*

TABILAB
TULAP - Turkish Language Processing Platform
Home
Tools
Datasets
About
TABILAB Repository Home / View Item
Search
BIBTEX
CMDI
Community

Named Entity Recognizer

Please use the following text to cite this item or export to a predefined format:

Güngör, Onur; Uskudarli, Susan and Güngör, Tunga, 2018, *Named Entity Recognizer*, TABILAB DIP project, a repository for NLP projects, tools and corpora, Bogazici University, https://hdl.handle.net/20.500.12913/24.

Authors	Güngör, Onur ; Uskudarli, Susan ; Güngör, Tunga
Item identifier	https://hdl.handle.net/20.500.12913/24
Project URL	https://github.com/BOUN-TABILab-TULAP/NER.git
Demo URL	https://tulap.cmpe.boun.edu.tr/demo/ner
Referenced by	https://aclanthology.org/C18-1177/
Date issued	2018-07-17
Type	toolService
Language(s)	
Description	Named entity recognition is the process of determining the named entities in a sentence. Given a set of sentences, the named entity recognizer outputs the named entities (if any) in each sentence. There are three types of named entities: person (PER), location (LOC), and organization (ORG).
Publisher	Boğaziçi University
Acknowledgement	Boğaziçi University Project code: 13083 Project name: Research Fund

(b) Details for the Named Entity Recognizer.

Named Entity Recognizer

Named entity recognition is the process of determining the named entities in a sentence. Given a set of sentences, the named entity recognizer outputs the named entities (if any) in each sentence. There are three types of named entities: person (PER), location (LOC), and organization (ORG).

You can use Docker to run the tool in your local environment. Do not forget to check GitHub page for more information

□ Use Sample Sentence

Enter a list of sentences (see Usage)

İstanbul Barosu'ndaki Yapay Zekâ, Robotlar ve Hukuk Konferansı'nda pirimiz Alan Turing'i anmadan olmazdı.

1 İstanbul Barosu'ndaki Yapay Zekâ, Robotlar ve Hukuk Konferansı'nda pirimiz Alan Turing'i anmadan olmazdı.

```
"root" : { 15 items
  "0" : [ 2 items
    0 : "İstanbul"
    1 : "B-ORG"
  ]
  "1" : [ 2 items
    0 : "Barosu'ndaki"
    1 : "I-ORG"
  ]
  "2" : [ 2 items
    0 : "Yapay"
    1 : "O"
  ]
  "3" : [ 2 items
    0 : "Zekâ"
    1 : "O"
  ]
  "4" : [ 2 items
    0 : ","
    1 : "O"
  ]
  "5" : [ 2 items
    0 : "Robotlar"
    1 : "B-ORG"
  ]
```

(c) NER for sentence: *İstanbul Barosu'ndaki Yapay Zekâ, Robotlar ve Hukuk Konferansı'nda pirimiz Alan Turing'i anmadan olmazdı*. Response is in BRAT format. The JSON serialization is shown in the dotted frame.

Figure 3: Searching (a), inspecting (b), and using a tool (c) in TULAP.

```
$ git clone https://github.com/BOUN-TABILab-TULAP/NER.git
$ docker build -t ner .
// docker console feedback omitted
$ docker run -d -p 8080:8080 ner
$ curl -X POST http://localhost:8080/ner/predict/ -H 'Content-Type: application/json' -d '{"text":"İstanbul
    Barosu'ndaki Yapay Zekâ, Robotlar ve Hukuk Konferansı'nda pirimiz Alan Turing'i anmadan olmazdı."}'

{'tagger_output': {'0': ['İstanbul', 'B-ORG'], '1': ["Barosu'ndaki", 'I-ORG'], '2': ['Yapay', 'O'], '3':
    ['Zekâ', 'O'],...}
```

Listing 1: The commands to acquire, build, run the NER tool and an API request. The output of the *docker build* and API response are truncated due to space limitation.

submit various resources, we expect that research teams will strongly benefit from a repository where reference versions of their work are collected in a single platform. The systematic publication of research outcomes supports the sustainability and continuity of research. Having control over the body of work makes it easier to track and include resources valued by the team that may not meet the criteria of external repositories. Similarly, it reduced the risks of relying solely on other repositories. This is not to imply that we do not support contributing our work on all relevant platforms. We expect that consistent effort in preparing reference material will result in following good practices that enhance the preparation of resources that can also be shared on all relevant platforms.

We approached the development of this platform as a software project starting from requirements elicitation, design, implementation, deployment, and maintenance. The project was managed with weekly meetings, version management, and issue-tracking tools. We believe that this approach was instrumental to achieving our goals. We note that the research output platform (*tabi-rop*) which we designed and developed for TULAP supports sharing computational research outputs in general and stands as a valuable contribution in its own right. It is not restricted to the deployment of TULAP and it can serve as an underlying platform in other domains and languages.

Our hope is that TULAP facilitates research and development efforts in Turkish NLP with information, demos, open-source resources, and easily accessible/usable reference versions of data and tools that we have provided.

Acknowledgements

This work was supported by Boğaziçi University Research Fund Grant Number 16909. We are grateful to all resource providers and the TABILab alumni who provided valuable feedback.

References

Buse Ak and Tunga Güngör. 2022a. Deasciifier. [Online; https://hdl.handle.net/20.500.12913/28; last accessed 19 July 2022].

Buse Ak and Tunga Güngör. 2022b. Sentence Splitter. [Online; https://hdl.handle.net/20.500.12913/26; last accessed 19 July 2022].

Buse Ak and Tunga Güngör. 2022c. Tokenizer. [Online; https://hdl.handle.net/20.500.12913/25; last accessed 19 July 2022].

Salih Furkan Akkurt, Büşra Marşan, and Susan Uskudarli. 2022. Boat v2 – a web-based dependency annotation tool with focus on agglutinative languages. This paper was presented at The International Conference and Workshop On Agglutinative Language Technologies as a Challenge of Natural Language Processing (ALTNLP) 2022.

Salih Furkan Akkurt and Susan Uskudarli. 2022. Bogazici Annotation Tool (BoAT) v2 - Web based Grammar Annotation tool for Morphologically Rich Languages. [Online; https://hdl.handle.net/20.500.12913/31; last accessed 19 July 2022].

Cem Rıfkı Aydın and Tunga Güngör. 2020. Combination of recursive and recurrent neural networks for aspect-based sentiment analysis using inter-aspect relations. *IEEE Access*, 8:77820–77832.

Cem Rıfkı Aydın, Tunga Güngör, and Ali Erkan. 2021. Sentiment Analyzer. [Online; https://hdl.handle.net/20.500.12913/9; last accessed 19 July 2022].

Batuhan Baykara and Tunga Güngör. 2022a. Text Summarizer. [Online; https://hdl.handle.net/20.500.12913/30; last accessed 19 July 2022].

Batuhan Baykara and Tunga Güngör. 2022b. Turkish abstractive text summarization using pretrained sequence-to-sequence models. *Natural Language Engineering*, page 1–30.

Gözde Berk and Abdüllatif Köksal. 2021. Bogazici Annotation Tool (BoAT) v1. [Online; https://hdl.handle.net/99999/38; last accessed 19 July 2022].

Steven Bird. 2006. NLTK: the natural language toolkit. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, pages 69–72.

Carl Boettiger. 2015. An introduction to Docker for reproducible research. *ACM SIGOPS Operating Systems Review*, 49(1):71–79.

Emrah Budur, Rıza Özçelik, Tunga Gungor, and Christopher Potts. 2020. Data and Representation for Turkish Natural Language Inference. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8253–8267, Online. Association for Computational Linguistics.

Buse Buz and Tunga Güngör. 2019a. Developing a statistical Turkish Sign Language translation system for primary school students. In *2019 IEEE International Symposium on INnovations in Intelligent SysTems and Applications (INISTA)*, pages 1–6.

Buse Buz and Tunga Güngör. 2019b. Sign Language Corpus. [Online; https://hdl.handle.net/20.500.12913/23; last accessed 19 July 2022].

Caner Derici, Yiğit Aydın, Çiğdem Yenialaca, and Nihal Yağmur Aydın. 2018a. Question Answering Corpus. [Online; https://hdl.handle.net/20.500.12913/22; last accessed 19 July 2022].

Caner Derici, Yiğit Aydın, Çiğdem Yenialaca, Nihal Yağmur Aydın, Günizi Kartal, Arzucan Özgür, and Tunga Güngör. 2018b. A closed-domain question answering framework using reliable resources to assist students. *Natural Language Engineering*, 24(5):725–762.

Caner Derici, Yiğit Aydın, Çiğdem Yenialaca, Nihal Yağmur Aydın, Günizi Kartal, Arzucan Özgür, and Tunga Güngör. 2018c. Question Answerer. [Online; https://hdl.handle.net/20.500.12913/29; last accessed 19 July 2022].

Gülşen Eryiğit. 2014. ITU Turkish NLP web service. In *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1–4.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A Deep Semantic Natural Language Processing Platform.

Grafana. 2022. Grafana. [Online; https://grafana.com/; last accessed 30 July 2022].

Onur Güngör, Mert Tiftikçi, and Çağıl Sönmez. 2018a. A corpus of Grand National Assembly of Turkish Parliament's transcripts. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Paris, France. European Language Resources Association (ELRA).

Onur Güngör, Mert Tiftikçi, and Çağıl Sönmez. 2018b. TBMM Corpus. [Online; https://hdl.handle.net/20.500.12913/20; last accessed 19 July 2022].

Onur Güngör and Eray Yıldız. 2017a. Linguistic features in Turkish word representations. In *2017 25th Signal Processing and Communications Applications Conference (SIU)*, pages 1–4.

Onur Güngör and Eray Yıldız. 2017b. Word Embeddings. [Online; https://hdl.handle.net/20.500.12913/18; last accessed 19 July 2022].

Onur Güngör, Suzan Üsküdarlı, and Tunga Güngör. 2018c. Improving named entity recognition by jointly learning to disambiguate morphological tags. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING)*, pages 2082–2092, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Onur Güngör, Suzan Üsküdarlı, and Tunga Güngör. 2018d. Named Entity Recognizer. [Online; https://hdl.handle.net/20.500.12913/24; last accessed 19 July 2022].

Hugging Face, Inc. 2023. Hugging Face. [Online; https://huggingface.co; last accessed 22 February 2023].

Abdullatif Köksal. 2018. Lemmatizer. [Online; https://hdl.handle.net/20.500.12913/27; last accessed 19 July 2022].

Abdullatif Köksal and Arzucan Özgür. 2020. The RELX dataset and matching the multilingual blanks for cross-lingual relation classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 340–350, Online. Association for Computational Linguistics.

Abdullatif Köksal and Arzucan Özgür. 2021a. Relation Extractor. [Online; https://hdl.handle.net/20.500.12913/5; last accessed 19 July 2022].

Abdullatif Köksal and Arzucan Özgür. 2021b. Sentiment Analyzer. [Online; https://hdl.handle.net/20.500.12913/7; last accessed 19 July 2022].

Abdullatif Köksal and Arzucan Özgür. 2021c. Twitter dataset and evaluation of transformers for Turkish sentiment analysis. In *2021 29th Signal Processing and Communications Applications Conference (SIU)*, pages 1–4.

Lyrasis. 2022. DSpace 7.x Documentation . [Online; https://wiki.lyrasis.org/display/DSDOC7x; last accessed 1 December 2022].

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60.

Büşra Marşan. 2022. BOUN Treebank v2. [Online; https://hdl.handle.net/99999/39; last accessed 28 July 2022].

Büşra Marşan, Salih Furkan Akkurt, Muhammet Şen, Merve Gürbüz, Onur Güngör, Şaziye Betül Özateş, Suzan Üsküdarlı, Arzucan Özgür, Tunga Güngör, and Balkız Öztürk. 2022. Enhancements to the BOUN Treebank Reflecting the Agglutinative Nature of Turkish. This paper was presented at The International Conference and Workshop On Agglutinative Language Technologies as a Challenge of Natural Language Processing (ALTNLP) 2022.

NGINX. 2022. NGINX. [Online; https://www.nginx.com; last accessed 1 Aug 2022].

Muhammed Yavuz Nuzumlalı and Arzucan Özgür. 2014a. Analyzing stemming approaches for Turkish multi-document summarization. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 702–706, Doha, Qatar. Association for Computational Linguistics.

Muhammed Yavuz Nuzumlalı and Arzucan Özgür. 2014b. Turkish MDS Data Set. [Online; https://hdl.handle.net/20.500.12913/17; last accessed 19 July 2022].

PostgreSQL. 2022. PostgreSQL. [Online; https://www.postgresql.org/; last accessed 1 Aug 2022].

Ali Safaya, Emirhan Kurtuluş, Arda Goktogan, and Deniz Yuret. 2022. Mukayese: Turkish NLP Strikes Back. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 846–863, Dublin, Ireland. Association for Computational Linguistics.

Haşim Sak, Tunga Güngör, and Murat Saraçlar. 2007a. Morphological disambiguation of Turkish text with perceptron algorithm. In *Computational Linguistics and Intelligent Text Processing*, pages 107–118, Berlin, Heidelberg. Springer Berlin Heidelberg.

Haşim Sak, Tunga Güngör, and Murat Saraçlar. 2011. Resources for Turkish morphological processing. *Language Resources and Evaluation*, 45(2):249–261.

Haşim Sak, Tunga Güngör, and Murat Saraçlar. 2007b. Morphological Analyser. [Online; https://hdl.handle.net/20.500.12913/4; last accessed 19 July 2022].

Haşim Sak, Tunga Güngör, and Murat Saraçlar. 2007c. Morphological Disambiguator. [Online; https://hdl.handle.net/20.500.12913/8; last accessed 19 July 2022].

Haşim Sak, Tunga Güngör, and Murat Saraçlar. 2010. Web Corpus. [Online; https://hdl.handle.net/20.500.12913/16; last accessed 19 July 2022].

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. BRAT: a web-based tool for NLP-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107.

Şerafettin Taşçı, A. Mustafa Güngör, and Tunga Güngör. 2006a. Compiling a Turkish-English bilingual corpus and developing an algorithm for sentence alignment. In *Proceedings of the Third International Bulgarian-Turkish Conference on ComputerScience*, pages 291–296, Istanbul.

Şerafettin Taşçı, Mustafa Güngör, and Tunga Güngör. 2006b. Turkish-English Parallel Corpus. [Online; https://hdl.handle.net/20.500.12913/19; last accessed 19 July 2022].

Utku Türk, Furkan Atmaca, Şaziye Betül Özateş, Gözde Berk, Seyyit Talha Bedir, Abdullatif Köksal, Balkız Öztürk Başaran, Tunga Güngör, and Arzucan Özgür. 2022. Resources for Turkish dependency parsing: Introducing the BOUN treebank and the BoAT annotation tool. *Language Resources and Evaluation*, 56(1):259–307.

Utku Türk, Furkan Atmaca, Şaziye Betül Özateş, Abdullatif Köksal, Balkiz Ozturk Basaran, Tunga Gungor, and Arzucan Özgür. 2019. Turkish treebanking: Unifying and constructing efforts. In *Proceedings of the 13th Linguistic Annotation Workshop*, pages 166–177.

Utku Türk, Furkan Atmaca, Şaziye Betül Özateş, Gözde Berk, Seyyit Talha Bedir, Abdullatif Köksal, Balkız Öztürk Başaran, Tunga Güngör, and Arzucan Özgür. 2021. BOUN Treebank. [Online; https://hdl.handle.net/20.500.12913/14; last accessed 19 July 2022].

UFAL. 2022. clarin-dspace digital repository. [Online; https://github.com/ufal/clarin-dspace; last accessed 31 July 2022].

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Zeynep Yirmibeşoğlu and Tunga Güngör. 2020a. ERMI at PARSEME shared task 2020: Embedding-rich multiword expression identification. In *Proceedings of the Joint Workshop on Multiword Expressions and Electronic Lexicons*, pages 130–135, online. Association for Computational Linguistics.

Zeynep Yirmibeşoğlu and Tunga Güngör. 2020b. Verbal MWE Identifier. [Online; https://hdl.handle.net/20.500.12913/10; last accessed 19 July 2022].

Şaziye Betül Özateş, Tunga Güngör, Arzucan Özgür, and Balkız Öztürk. 2018. A morphology-based representation model for lstm-based dependency parsing of agglutinative languages. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 238–247.

Şaziye Betül Özateş, Tunga Güngör, Arzucan Özgür, and Balkız Öztürk. 2020. Dependency Parser. [Online; https://hdl.handle.net/20.500.12913/11; last accessed 19 July 2022].

Seçil Öztürk, Bülent Sankur, Tunga Güngör, Mustafa Berkay Yılmaz, Bilge Köroglu, Onur Ağın, Mustafa İşbilen, Çağdaş Ulaş, and Mehmet Ahat. 2014a. Scientific Abstracts Corpus. [Online; https://hdl.handle.net/20.500.12913/15; last accessed 19 July 2022].

Seçil Öztürk, Bülent Sankur, Tunga Güngör, Mustafa Berkay Yılmaz, Bilge Köroglu, Onur Ağın, Mustafa İşbilen, Çağdaş Ulaş, and Mehmet Ahat. 2014b. Turkish labeled text corpus. In *2014 22nd Signal Processing and Communications Applications Conference (SIU)*, pages 1395–1398.