Submitted: 13/09/2025 - Accepted: 11/11/2025 - Published: 29/12/2025

TS Corpus Project: An online Turkish Dictionary and TS DIY Corpus

Taner Sezer

Mersin University, Turkey

DOI: 10.26417/ythnjj06

Abstract

TS Corpus is a free and independent project that aims building Turkish corpora, NLP tools and linguistic datasets. Since 2011, 10 corpora, various NLP tools, a large dataset and an online dictionary has been released. This paper focuses on the "online dictionary" and " TS do-it-yourself corpus" released by the project. The dictionary data is based on TDK (Turkish Language Society) Contemporary Dictionary. However, the dictionary published serves many enhanced functions at user interface level. But, the main importance of the study is about the results presented to the users upon their queries; the presentation of collocations and tri-grams of the key word searched for. The collocations are harvested from a large Turkish corpus, +760 million tokens and the tri-grams were generated from Turkish Wikipedia pages. The do-it-yourself corpus (TS DIY Corpus), allows users to build their own corpora, modify or delete the uploaded texts and run queries. Users may run queries in different modes, such as "as is", "starting/ending with" or including; besides advanced query option allows users to run queries with part-of-speech tags and lemmas. The results are given in KWIC (keyword in context) format. Various text classification options such as pubdate, author, domain, genre etc. could be selected during corpus creation. As the number of available Turkish corpora is limited, TS DIY Corpus is applicant to be a useful, well-known and largely used software for the scholars and researchers who wants to use a Turkish corpus or study over Turkish texts of their own.

Keywords: TS Corpus, Turkish Corpus, Corpus Linguistics, Collocation, Part-of-Speech Tagging, Turkish Dictionary

Introduction

TS Corpus Project

TS Corpus is a free and independent project that aims building Turkish corpora, NLP tools and linguistic data sets (Sezer, Sezer, 2013). The project started in 2010 and released the first corpus in 2011. Since then, 10 other corpora, NLP tools and a dataset had released by the project.

This paper will focus on two new software recently released by the project. First, an online dictionary that features collocations, tri-grams and hyphenation of the query input. And a do-it-yourself corpus software that allows users to build their own corpora, edit or add content freely, with various search options, and running simple queries and advanced queries with part-of-speech tags and lemmas.

Online Turkish Dictionary

Turkish Language Society (TDK), Contemporary Turkish Dictionary was first published in 1945. In 2006, the dictionary was carried to digital platform and released via TDK official website. Today, the online interface for this dictionary is available at the main page of the official TDK website.

The dictionary fetches word class, definition of the word, samples retrieved from Turkish literature and the representation of the query word in Turkish Sign Language. If exists, idioms and proverbs, and compound words are presented to the user.

Our dictionary software uses TDK Contemporary Turkish Dictionary data but serves new features.

The Design

The user interface served by TDK is not mobile friendly. As the numbers of users reaching Internet via mobile devices increased, mobile friendly websites, in other words, websites with responsive design became more popular. These websites can transform their design, which is based on a grid system, according to the clients screen size and resolution. Therefore, they can fit to any screen size of a mobile device and serve better usability. So, one of the major points we focused on is to serve the dictionary with a responsive design in order to support mobile devices.

We used Bootstrap framework by Twitter. The official Bootstrap page defines this framework as “HTML, CSS, and JS framework for developing responsive, mobile first projects on the web.” We also used Ajax. Ajax is an asynchronous communication architecture that fetches data from a database or a server without the refreshment of the page. The combination of these two allowed us to build a mobile friendly web interface with practical usage.

As another design feature different from the dictionary served by TDK, thanks to Ajax, as the user began typing in the search box, possible hits are shown immediately. This design, as well as supporting usage of the arrow keys on a desktop PC or a laptop,

allows users to scroll by using touch screens in a touch-device to navigate through or select one of those possible hits.

The possible hits also involve similar words even the word has a character with a caret or one of the characters peculiar to Turkish alphabet like "*ş*" or "*ğ*". This feature is very useful for the users whose operating system does not support Turkish or with the keyboards lack of these letters.

Tri-Grams

An n-gram is the consecutive sequence of n number of items in a text. In this definition n represents the number of items that forms the set; a bi-gram is consist of two items and a tri-gram is consists of three items.

In 2015, TS Corpus released Turkish Wikipedia Corpus. Later in same year, the data set of this corpus *(including raw texts in XML format, a tokenized version of this data, a part-of-speech tagged version and bi-grams and tri-gram harvested from this dataset)* was released by Linguistic Data Consortium (LDC). The tri-gram set generated from this corpus has 12.5 million units. As a word is searched, with the results, the most frequent ten tri-grams are presented to the user if exists in the tri-gram database.

For computational linguistics, statistics and language teaching n-grams introduce valuable data. In a dictionary, a tri-gram is useful for representing the authentic usage of the query word.

Syllabication

Syllabication is the process of splitting a word into syllables. One of the recent corpus we released was TS Syllable Corpus. The corpus was build as a part of another study to define the syllable inventory of Turkish. We designed a script, named TS syllable tagger. The syllable tagger takes advantage of TRMorph (Çöltekin, 2010), an open sourced morphological analyzer developed by Çöltekin, for hyphenation. After the word hyphenated, the script attaches a tag to each syllable. The used tags are V *(a)*, CV *(ve)*, VC *(al)*, VCC *(ilk)*, CVC *(tür)* and CVCC *(ters)* where V refers to vowel and C refers to consonant. These 6 types of syllables are known as valid syllables for Turkish harmony.

For every result shown in dictionary, the syllables that form the word is presented with the syllable tag-set mentioned above.

Collocations

The most important and "*state of the art* feature" of the online dictionary we present is the presentation of the collocations for the query key.

Evert (Evert, S. 2007:4) defines a collocation as "a combination of two words that exhibit a tendency to occur near each other in natural language", upon the well-know idea by Firth (Firth, J. R. 1957:179) "you shall know a word by the company it keeps".

The collocations are valuable references for understanding a word, extracting multi-word expressions and provide useful hints for language teaching.

So, we build a corpus that is consisted of +764 million tokens. The corpus is powered by CQPWeb and CWB. We prepared a Python script that automatically runs a query in the corpus for each word in the dictionary database one-by-one and saves results in a MySQL table.

If the query word appears in the corpus then the script runs collocation function of the CQPWeb. Among many others, we choose log-likelihood statistical method for calculating collocations as it stands on collocations by significance following Dunning (Dunning, 1993). This method is efficient as it is specially designed to surpass the association of the words by chance.

We also take advantage of CQPWeb to classify the collocations as the right and left side ones. This means, the collocations for the query word are harvested in two groups, the preceding and the following according to the node.

Spell-book

A module added to the online interface is the spell-book. "Zemberek" is an open-sourced Turkish NLP library, written in Java, by Akın&Akın (2007), featuring a morphological parser and a spell checker; not only for Turkish but also for other Turkic languages. We wrapped Zemberek spell-checker service within our module with AJAX and run Bash scripts at background level. Spell-book relies on a lexicon and processes the user input at the time of typing. The suggestions are listed below the text input area. Furthermore, using Zemberek helped us to fetch suggestions for the words that might not be in a hard-copied spelling book. As the user goes on typing, no matter how long the word is or what fixes are added to the word, our module produce suggestions or shows if the input is correct or false.

TS DIY Corpus

Building a corpus is a hard and time consuming task that also requires advanced computer literacy. Corpora are served as ready-made to the users. For building such a corpus, the data should collected, prepared and processed before compiling as a corpus. Therefore, corpora presented to the users in this form are composed of pre-defined and constant data sets.

However, users may need to use different datasets or want to build a specialized corpus by using data they own data. To accomplish this task and full-fit flexible desires of the scholars and researchers, we designed a "do-it-yourself corpus" software.

This is the second, freshly designed and improved version of the software. The very first release of TS DIY Corpus was in 2014. It used to rely on PHP-MySQL frame that we later noticed not enough to run the tasks we desired efficiently and accurately.

There are many software, that can be used online or locally is available for text processing and corpus building; TXM, AntConc, CQPWeb, Sketch Engine etc. However, we aimed to design a software that involves and combine all of the following features:

Online available: Users should have access their corpora anywhere they are connected to the Internet.

Easy to use: *N*o requirement for advanced computer literacy about text processing, tokenization, database management, regular expressions, part-of-speech tagging etc.

Flexible Corpus Design: Users are allowed to tag meta information about the texts upon their design features.

Automatic Processing: The texts submitted to the software are tokenized and tagged automatically. Users do not need to have or know how to use a tokenizer or a part-of-speech tagger.

Below, we will explain the features of the TS DIY Corpus software and sample it's usage.

Under the hood

TS DIY Corpus software is consisted of many different tools. The main framework is based on Python Django. The corpus interface that users interact with is also designed by Django, of course by taking advantage of HTML, CSS, JS and AJAX.

Django includes many libraries that help us to reduce required work force and save time during coding. The user registration and authentication, interface design, database connection, preparing query statements are the very first points pop in mind. Compared to a PHP-MySQL based software, frequently used for similar projects, these tasks are run with ease by Django.

Besides, the existence of TS DIY Corpus software relies on the scripts we previously prepared and released. Titles 3.2 and 3.3 will detail these scripts.

Tokenization

Basically, an electronic text is a sequence of letters, punctuation marks, numerals, white-spaces, end-of-line markers and other characters that could be generated via keyboard. In order to process these texts with a computer, these sequences should be segmented (Mitkov, 2005:201).

Forst and Kaplan (2006) underlines the importance of the precise tokenization of texts over the overall accuracy of the following processes, such as part-of-speech tagging, syntactic parsing etc.

"Word" and therefore "token" are vague terms in NLP studies. Evert (2008) defines "*word"* as an "*entirely generic term which may refer to any kind of lexical item, depending on the underlying theory or intended application.*" Similarly, Grefensette (Grefensette et al. 1994) underlines, "*there are many ways to decide what will be*

considered as unit for a computational approach to text." So, we may say that, "there is not one and absolute correct way" for tokenization, but clearly, tokenization is a vital process that should be handled carefully and run by following the distinctive features of the target language. The search ability of the programming languages are dramatically raised with the tokenized texts, compared to non-tokenized texts.

We build TS Tokenizer, based on utf8-tokenizer, which is a part of TreeTagger and prepared by Sharoff and Schmid. However, we enhanced the tokenizer to full-fit the requirements of Turkish. Each piece of text submitted by the user is tokenized automatically and then stored in the database. As the processing completed (including part-of-speech tagging), the submitted texts will be ready-to-use in the corpus.

Part-of-speech Tagging

Another script we prepared previously and used when building TS DIY Corpus is the part-of-speech tagger. The tagger stands on morphological disambiguator (MD) designed by Sak, Güngör and Saraçlar (2008). Since we need enhancements, enlarging the lexicon, extend the tagset and re-design the generated output we made required improvements.

As well as tokenization, the part-of-speech tagging also run automatically as the users submits a piece of text.

Using TS DIY Corpus

Registration

Users should register to the TS DIY Corpus in order to use it. The registration form has 3 obligatory fields to fill in; user name, password and e-mail address. The registration is necessary, as each user will build his or her own corpus with the data he or she owns.

Creating a new corpus

As registration is completed, the user can reach to TS DIY Corpus main screen. This screen includes the main menu. The very first thing a user should do is creating a new corpus by clicking "C*reate New Corpus*" button. A corpus name and desired text classification titles are asked to user. Users are allowed to select any number of text classification items from a pre-defined list. These options are automatically added to search restriction options in the query screen for the corpus created. The corpus is created immediately, with the name provided and selected text classifiers as the user clicks on "*Create Corpus*" button.

Text Classification Items

During corpus creation, users can select various text classification items from the offered list. The list contains text restriction ranges for written and spoken data offered by BNC mainly. It is also possible to extend or narrow these restrictions.

Corpora List

Corpora List menu in the main screen allows users to run three different actions. "Contents" link fetches a list of texts added to the relevant corpus. Editing or deleting processes are managed in this screen and details of this action will be given in title 4.5.

"*Edit*" link is for editing text classification items as mentioned before and "*Delete*" link is for deleting the corpus.

Corpus Content

Users are allowed to add unlimited number of texts to each corpora they create. Each text in a corpus can be edited or deleted at any time. In order to edit or delete a text, users should follow "*Corpora List*" menu and then click on the "*Content*" link for the relevant corpus.

"*Add New Content*" link directs users to a new screen where users can add a new text easily just like filling an online form. The fields are automatically derived from the options set during corpus creation.

As the user submits a text, a notification will appear explaining the status of the process. This process refers to tokenization and part-of-speech tagging. The progress could be followed in the contents list with a check-box. If the check-box is checked, it means the submitted text is processed and ready to use.

Editing option allows both editing *text classification items* and the *text itself*.

Query History

Users can reach the queries previously run in every corpus using "*Query History*" link. The query history table contains search term (the query key), corpus name, date/time of the query and a "search again" button that runs the same query immediately by a simple click.

The query history table also allows running a search within the table.

Running a Query

The query screen includes defined text classifiers, a drop-down corpus selection menu and another drop-down menu for query mode.

The double dash in the text classifier drop-down menu means no restriction is selected and all the texts will be included in the query. Users can select multiple options from the text classification items drop-down menu by holding down Ctrl key in Linux and Windows and Command key on Mac OS.

The query mode supports "as is", "starts with", "ends with", "including" modes both case-sensitive and case-insensitive as well as "regular expression", and "advanced" modes. *As is* refers that the exact match given as the query key will be searched. *Starts*

with, *ends with* and *including* modes are self-explanatory. Regular expression queries allow users to use wild cards within query key.

Advanced query mode allow users to run queries by using the query language provided. In order to use advanced queries, users should know the tag-set we used during part-of-speech tagging. The tag set has the following tags (Sak et al, 2008) (Sezer, 2016):

Noun (*noun*), Verb (*verb*), Adj (*adjective*), Conj (*conjunction*), Det (*determiner*), Adv (*adverb*), Postp (*postposition*), Pron (*pronoun*), Num (*number*), Ques (*question suffix*), Interj (*interjection*), Punc (*punctuation*), Dup (*duplication*), abbr (*abbreviation*), intAbbr (*internet abbreviation*), YY (*misspelling*), emoticon (*smileys*), intEmphasis (*internet emphasis*), intSlang (*internet slang*) and UnDef (*undefinite*).

Each advanced query must be written in square brackets and should involve one these three structural attribute: PoS, Word or Lemma.

For instance, if the user want to search for a PoS tag, lets say a noun, query key should be formed up in the following form:

[Pos="Noun"]

The advanced queries allow using multiple annotations in one query statement. For example, the following query will fetch all the occurences of any adjectives followed by the word "araba" *(car)*:

[PoS="Adj"] [Lemma="Araba"]

In advanced mode, also lemma is an annotation that could be used. The following query will fetch any word in the corpus that is inflected from ev (house) lemma.

[Lemma="ev"]

Lemma and Pos annotations could also be used together.

[Lemma="gül" & PoS="Verb"]

This query will fetch all the occurences of the verb gül.

Saving Results

Users are allowed to save results of the queries in many different formats, including CSV (comma-seperated values), tables to use with spreadsheet software and PDF. Results also could be send to printer directly or copied to clipboard with a single click.

Results

The two software we introduced in this paper are serving unique features for their kind which we think useful and make contributions to Turkish NLP studies.

The representation of collocations and tri-grams are the first appearance of these information in a Turkish dictionary. This information could help students learning

Turkish as a second language. The collocations provided may contribute building a word-net for Turkish and supply valuable information for statistical NLP studies.

The TS DIY Corpus software, will help users who want to build their own corpus but lack of adequate computer literacy or hardware. The software features enhanced features like automatic tokenization and part-of-speech tagging which are hard to deal with.

Also, as users are free to design their own corpus privately, the could use texts which could never appear in a publicly available corpus due to copyright fees.

The following versions of TS DIY Corpus is planned to include language selection during corpus creation. So, users may created tagged corpora for various languages. Also improvements with the served statistical results is in the schedule for next versions.

References

[1] Akın, A. A., & Akın, M. D. (2007). Zemberek, an open source NLP framework for Turkic Languages. *Structure*, *10*.

[2] Çağrı Çöltekin (2010). A Freely Available Morphological Analyzer for Turkish In Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC2010), Valletta, Malta, May 2010.

[3] Dunning, Ted E. (1993). Accurate methods for the statistics of surprise and coincidence. Computational Linguistics , 19 (1), (pp: 61-74)

[4] Evert, S. (2005). The statistics of word cooccurrences (Doctoral dissertation, Dissertation, Stuttgart University).

[5] Evert, Stefan. "Corpora and collocations." Corpus linguistics. An international handbook 2 (2008): 1212-1248.

[6] Firth, J. R. (1957). A synopsis of linguistic theory 1930-55. In Studies in linguistic analysis, pages 1–32. The Philological Society, Oxford. Reprinted in Palmer (1968), pages 168-24

[7] Forst, M., & Kaplan, R. M. (2006). The importance of precise tokenizing for deep grammars. In Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006) (pp. 369-372).

[8] Mitkov, R. (2005). The Oxford handbook of computational linguistics. Oxford University Press.

[9] Sak, H., Güngör, T., & Saraçlar, M. (2008). Turkish language resources: Morphological parser, morphological disambiguator and web corpus. In Advances in natural language processing (pp. 417-427). Springer Berlin Heidelberg.

[10] Sezer, B., Sezer, T. 2013. TS Corpus: Herkes İçin Türkçe Derlem. Proceedings 27th National Linguistics Conference. May, 3-4 Mayıs 2013. Antalya, Kemer: Hacettepe University, English Linguistics Department. (pp: 217-225)

[11] Sezer, T, and Sezer, T. TS Wikipedia LDC2015T15. Web Download. Philadelphia: Linguistic Data Consortium, 2015.
[12] Sezer, T. 2016. Tweets Corpus: Building a Corpus by Social Media. *Journal of National Education and Social Sciences*. Spring 2016, 210, ss: 621-633