T-PAS Scraper: an Application for Linguistic Data Extraction and Analysis

Emma Romani[1], Valerio Gattero[1], Elisabetta Ježek[1]
1. Università degli Studi di Pavia, Pavia, Italy
emma.romani01@universitadipavia.it,
valerio.gattero01@universitadipavia.it, jezek@unipv.it

Abstract

In this paper we introduce T-PAS Scraper, a new online application for linguistic data extraction and analysis connected to the T-PAS resource, a corpus-based digital repository of Italian verbal patterns (Ježek et al., 2014). The application is conceived as a supplementation of the main functions of T-PAS and can be used concurrently with the resource, thus extending its accessibility. It consists of 25 different scripts which operate automatically on the database of the resource and can be useful for quantitative and qualitative studies of the linguistic data it contains.

1 Introduction

T-PAS Scraper is a new online application for linguistic data extraction and analysis. It is designed as an extension and supplementation of T-PAS resource (Ježek at al., 2014)[1], a corpus-based digital inventory of Italian verbal patterns, which provide syntactic and semantic information on the verb-argument structures (that is, the patterns)[2].

Initially, the project was not conceived as online application for the linguistic analysis of T-PAS resource data. The first idea was to find a way to speed up the revision of the resource: we needed a fast system that could facilitate the manual correction of annotators' mistakes within the patterns contained in the editor of the resource. As the dimension of the resource is considerable (T-PAS is a repository of 5326 patterns)[3], checking all of them manually while going through the correction and refinement phase would have been time-consuming. Annotators' mistakes are not widespread within the resource as most of the work performed on T-PAS was carried out manually, but isolated errors can occur.

As a solution, we developed a series of T-PAS-specific scripts running on the updated T-PAS resource database (a JSON-structured file containing all the patterns and the related information), which can extract lists of aggregated data, displayed in columns. By skimming the lists, one can easily notice data errors in the extracted data and therefore correct them by moving to the editor of the resource and editing the patterns which were wrongly annotated.

As the number of scripts that we developed was consistent, covering several aspects of the resource, we believed that they would have been useful for users and not only for annotators' revision. We decided to build an online application, which is called T-PAS Scraper, that extends T-PAS accessibility: it can be used by future T-PAS users for quantitative and qualitative studies of its linguistic data[4].

In this paper we describe the application in its components, how it is related to T-PAS resource and some possible uses.

The paper is structured as follows. In Section 2 we briefly describe the T-PAS resource, its main features and the online interface. In Section 3 we introduce T-PAS Scraper and provide a technical

[1] Link to the project: https://tpas.unipv.it/. Both T-PAS resource and T-PAS Scraper are accessible from this link.

[2] T-PAS resource has been developed within Sketch Engine (Kilgarriff et al., 2014); it will be available online by the end of 2021. T-PAS Scraper application has been developed at the University of Pavia during a curricular internship in which the first two authors were involved with the aim of refining, correcting, and improving the resource and its main features before its online publication.

[3] Last update: 25/08/2021.

[4] This is the primary and final purpose of the application, as the refinement, correction, and improvement phase is about to be concluded for the upcoming release of the resource.

explanation on how it was built; we also present the main functions of T-PAS Scraper, and in which sense they are complementary to T-PAS resource, as well as how it can be used from a user perspective. In Section 4 we discuss some future perspective on the project.

2 T-PAS Resource

T-PAS (Ježek et al., 2014) is a corpus-derived resource consisting of an inventory of Typed Predicate-Argument Structures (T-PAS) for Italian verbs. It is a gold standard for Italian verb-argument structures. The resource is being developed at the University of Pavia with the technical support of Lexical Computing Ltd. (CZ) and is intended to be used for linguistic analysis, language teaching, and computational applications. The resource consists of four fundamental components:

1. a repository of corpus-derived predicate argument structures (called *patterns*) with semantic specification of their argument slots, e.g. [Human] drinks [Beverage];
2. an inventory of ca. 200 corpus-derived semantic classes (called *Semantic Types*) organised in a hierarchy (called *System of Semantic Types*), used for the semantic specification of the arguments;
3. a corpus of manually annotated sentences that instantiate the different patterns of the verbs in the inventory. Corpus lines are tagged with their respective pattern numbers and anchored to the verb they feature, which is the lexical unit of analysis[5];
4. an editing system called Skema (Baisa et al., 2020), which allows the registration of patterns and all the syntactic and semantic information associated therewith and facilitates the manual annotation of corpus instances (directly linked to the patterns)[6].

Typed predicate-argument structures are patterns that display the semantic properties of verbs and their arguments: for each meaning of a verb, a specific pattern is provided. As referenced above, the patterns are corpus-derived, i.e., they are acquired through the manual clustering and annotation of corpus instances, following the CPA methodology (i.e., Corpus Pattern Analysis; Hanks, 2013). Currently, T-PAS contains 1165 implemented verbs, 5,326 patterns, and ca. 200,000 annotated corpus instances.

In the resource, each pattern is labelled with a pattern number and connected to a list of corpus instances realising that specific verb meaning. The Skema editor enables the registration of different semantic and lexical information in each pattern: the *verb*, which in T-PAS is generally in its infinitive form - e.g., *bere* (Eng., 'to drink'); the *Semantic Types* (e.g. [Human], [Beverage], always portrayed within square brackets), specifying the semantics of the arguments selected by the verb; argument positions[7], which are filled by the Semantic Types in the patterns; the *sense description*, i.e. a brief definition of the meaning of the verb in that specific pattern; a *lexical set* (optional) for each Semantic Type in the pattern, i.e. a selection of the most representative lexical items instantiating that Semantic Type (e.g. *vino* = 'wine' | *birra* = 'beer' | *aranciata* = 'orange juice' are good candidates for the lexical set of [Beverage]); the *roles* (optional) played by some specific Semantic Types in certain contexts: in particular, the Semantic Type [Human] can acquire the role of Athlete, Doctor, Musician, Host, Guest, Writer, etc., depending on the verb selecting it as an argument; the *features* (optional) associated with the Semantic Types, i.e. certain semantic characteristics required by the pattern syntax (e.g. Plural) or by the specific verb meaning (e.g. Female, Negative, Visible); *prepositions* (for prepositional complements); *particles* (for adverbials); *complementizers* (for clausals); *quantifiers*, and *determiners* (for lexical sets), which can be implemented according to the specific argument position in question.

The System of Semantic Types used to classify the semantics of arguments (Pustejovsky et al., 2004; Ježek, 2019) is a hierarchy of general semantic categories obtained by manual clustering of the lexical items found in the argument positions of corpus-derived valency structures. The System currently contains ca. 200 Semantic Types that are hierarchically organised

[5] The reference corpus for the resource is the web corpus ItWac (reduced), provided by Sketch Engine. It contains around 935 million tokens.

[6] Skema (Baisa et al., 2020) is a corpus pattern editor system implemented to facilitate the management of manual annotation of concordance lines with user-defined labels and the editing of the corresponding patterns in terms of slots, attributes and other features following the lexicographic technique of CPA (Hanks, 2013).

[7] I.e., subject, object, adverbial, clausals, prepositional complement, predicative complement. They can be optional, but yet registered in the pattern if they are relevant to the sense of the verb.

on the basis of the ‘is a’ (subsumption) relation (e.g. [Human] is an [Animate]).

T-PAS online version, which will be publicly available for the users by the end of 2021, will consist of:

1. the repository of predicate-argument structures (patterns);
2. five good corpus examples (GDEX; Kilgarriff et al., 2008) for each of the patterns (previously annotated);
3. the System of Semantic Types;
4. a search engine that allows to search Semantic Types and argument positions (subject, object, etc.) in combination.

T-PAS Scraper aims at completing and integrating T-PAS functionalities: the two interfaces can be used complementarily when visualizing the pattern and searching for specific linguistic phenomena.

3 T-PAS Scraper

3.1 Building T-PAS Scraper

T-PAS Scraper is, in its first release, constituted by two parts: the scripts to retrieve the data from T-PAS database, which were created using Python[8] and PyCharm[9], and a graphical interface that produces a cross-platform executable program.

The program can load a Sketch Engine-compatible database, select a script, and run it. Once clicked on the “Run script” button (see Figure 1), a JSON file is produced with the requested content (i.e., the list of extracted data). Every script is different and prints different complex data in its output, but the way in which data is structured is identical in all of them.

The online user application was created by programming a web application with Angular, a popular front-end framework, PrimeNG, a component library, and Express, a NodeJS server that allows the application to be loaded on Heroku, a hosting company. The procedure followed in order to develop this application consists of two steps: first of all, the scripts were formally defined using a pseudocode, the linguistic data that are the object of the extraction. In the second phase, each script was implemented and printed on a JSON file.

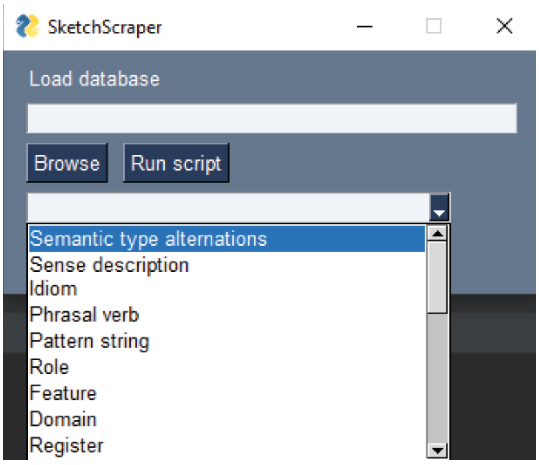


Figure 1. T-PAS Scraper program with the list of scripts that can be run on the JSON database

The online interface (see Figure 2) consists of several parts. First, a menu indicates the top categories of scripts that can been chosen. The top categories are Verbs, Semantic Types, Arguments and Lexical Sets; each of them contains a group of related scripts. For each top category there is a description page and a list of several scripts. Once a script has been selected, the data is shown in table format (generally long lists with different columns), with the possibility of filtering and paging the results in different ways. Data are displayed in alphabetical order, based the first column on the left (containing the verbs). There is also the option to export the current script in Excel format, which can be handy for studying the data externally.

3.2 Using T-PAS Scraper

T-PAS Scraper is useful to access semantic and syntactic information about verbs and their arguments, which cannot be accessed in T-PAS online in aggregate (see Section 2). The basic idea that underlies T-PAS Scraper usage is to have an aggregated overview of the lists (i.e., columns) of data being extracted[10].

As for the verbs, one can visualize the complete list of the 1165 verbs with different information: the number of patterns for each verb (and what is the average number of patterns, as well as which verbs have the highest number of patterns); the frequency and per-million frequency in the corpus (to check whether the frequency is somehow

[8] https://www.python.org/ (last access: 23/09/2021).

[9] https://www.jetbrains.com/pycharm/ (last access: 23/09/2021).

[10] It is possible to refine the research by filtering the columns (e.g., a specific verb and its related information in each script can searched typing the verb in the filter function of the first column of the list, which work as a search box, showing all the items corresponding to the typed query), but this kind of query better fits to T-PAS resource and its search engine (which allows to type a specific verb and open the list of patterns and the examples).

related to the number of patterns of that specific verb). The entire inventory of 5326 patterns contained in T-PAS online, together with the related sense description is provided, both separately and jointly. Verbs in patterns can be registered differently from their base form (i.e., the infinitive) or show for examples reflexive uses, and therefore the entire list of the verb forms can be filtered in order to obtain those forms (e.g., *lavarsi*, Eng. 'to wash yourself').

The complete inventory of the 212 phrasal verb patterns annotated in the resource (e.g., *buttare via*, Eng. 'to throw away') and 388 idiomatic uses (e.g., *bersi il cervello*, 'to go crazy') can also be searched, also in parallel to the patterns in T-PAS online for explanatory examples.

As for Semantic Types, one can search for the most frequent alternations of Semantic Type in argument positions (Ježek et al., 2021; see Figure 2 column 4 for examples) as well as the semantic roles and the features associated to the Semantic Types (see Section 2).

For what concerns the arguments, a list of the argument structures is provided (e.g., subject-object, subject-clausals, subject-prepositional complement) as well as those which are optional and obligatory. Syntactic alternations of arguments (e.g., *finire il pranzo* (object) vs. *finire di mangiare* (clausals) – Eng., 'to finish the meal' vs. 'to finish eating') are also listed.

Finally, complementizers, prepositions, adverbial particles, and obligatory determiners annotated within the patterns in T-PAS, as well as lexical sets, can be extracted through T-PAS Scraper and analysed by the researcher in their distribution.

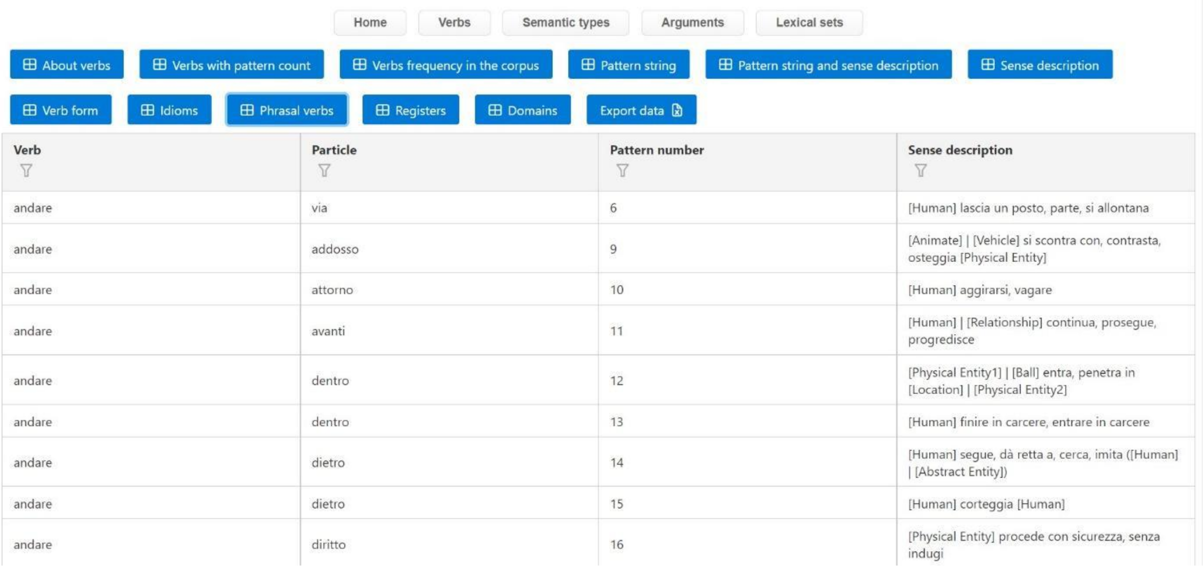
In Table 1 we provide some quantitative data regarding T-PAS resource, that can be extracted through T-PAS scraper.

script	**n. of items**
verbs	1165
patterns	5326
idiomatic patterns	388
phrasal verb patterns	212
semantic type alternations	3243
semantic types with roles	173
semantic types with features	228
optional arguments	1032
syntactic alternations of arguments	267
patterns with lexical sets	1109

Table 1. Summary of the quantitative data from the scripts

4 Conclusions and Future Perspectives

In this paper we introduced T-PAS Scraper, a new online application for linguistic data extraction and analysis specifically devised to retrieve the data contained in the T-PAS resource and make them available to users for purposes of linguistic analysis, thus extending T-PAS resource accessibility. We described why and how it was



Verb	Particle	Pattern number	Sense description
andare	via	6	[Human] lascia un posto, parte, si allontana
andare	addosso	9	[Animate] | [Vehicle] si scontra con, contrasta, osteggia [Physical Entity]
andare	attorno	10	[Human] aggirarsi, vagare
andare	avanti	11	[Human] | [Relationship] continua, prosegue, progredisce
andare	dentro	12	[Physical Entity1] | [Ball] entra, penetra in [Location] | [Physical Entity2]
andare	dentro	13	[Human] finire in carcere, entrare in carcere
andare	dietro	14	[Human] segue, dà retta a, cerca, imita ([Human] | [Abstract Entity])
andare	dietro	15	[Human] corteggia [Human]
andare	diritto	16	[Physical Entity] procede con sicurezza, senza indugi

Figure 2. Screenshot of T-PAS Scraper online application with the "phrasal verbs" script as an example (each script has its specific table format as different type of data can be extracted from the resource database). We can see the verb in the first column, the particle in the second, the number of the pattern in the list of available ones for that verb, and the description of the sense of the verb in that pattern.

built and its main functions related to T-PAS resource. We also suggested some possible uses in terms of qualitative and quantitative analysis, from a user perspective.

As a new-born project, T-PAS Scraper application is just at its initial stage and further work can be done. In particular, new scripts can be added to enrich the existing ones.

Currently, the data displayed on the application are in a static form: the updated database needs to be manually re-uploaded in case of some changes in T-PAS editor. The final goal would be to load the data from T-PAS database within Sketch Engine in real time, run the scripts and show the results: in a first phase the data will be displayed in this way. Loading data directly from Sketch Engine also requires coordination and the creation of a dedicated API with external authentication, which does not currently exist. A dynamic infrastructure has countless advantages: it allows to view data in real time, it is scalable and functional, and can also communicate with other systems.

T-PAS Scraper may eventually be extended to resources other than T-PAS whose structure is compatible with the database configuration of T-PAS and Sketch Engine.

References

Baisa, V., Tiberius, C., Ježek, E., Colman, L., Marini, C. & Romani, E. (2020). Skema: A New Tool for Corpus-driven Lexicography. In *Proceedings of the 19th EURALEX International Congress*.

Hanks, P. (2013). *Lexical Analysis: Norms and Exploitations*. Cambridge: The MIT Press.

Ježek, E. (2019). Sweetening Ontologies Cont'd: Aligning bottom-up with top-down ontologies. In A. Barton, S. Seppälä & D. Porello (eds.) *Proceedings of the Joint Ontology Workshops 2019*. Graz, Austria.

Ježek, E., Magnini, B., Feltracco, A., Bianchini, A. & Popescu, O. (2014). T-PAS: A resource of corpus-derived Types Predicate-Argument Structures for linguistic analysis and semantic processing. In *Proceedings of LREC*. pp. 890–895.

Ježek, E., Marini, C., Romani, E. (2021). Encoding semantic phenomena in verb-argument combinations. In Kosem, I., Cukr, M., Jakubíček, M., Kallas, J., Krek, S. & Tiberius, C. (eds.), *Electronic lexicography in the 21st century. Proceedings of the eLex 2021 conference*. 5–7 July 2021, virtual. Brno: Lexical Computing CZ, s.r.o.

Kilgarriff, A., Baisa, V., Bušta, J., Jakubíček, M., Kovář, V., Michelfeit, J., Rychlý, P. & Suchomel, V. (2014). The Sketch Engine: ten years on. Lexicography, 1(1), 7-36.

Kilgarriff, A., Husák, M., McAdam, K., Rundell, M. & Rychlý, P. (2008, July). GDEX: Automatically finding good dictionary examples in a corpus. In *Proceedings of the XIII EURALEX international congress*. Barcelona, Spain: Documenta Universitaria. pp. 425-432.

Pustejovsky, J., Hanks, P. & Rumshisky, A. (2004). Automated Induction of Sense in Context. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING)*. Geneva, Switzerland.