A collaborative system for building and maintaining wordnets

Tomasz Naskręt
G4.19 Research Group, Department of Computational Intelligence
Wrocław University of Science and Technology, Wrocław, Poland
tomasz.naskret@pwr.edu.pl

Abstract

A collaborative system for wordnet construction and maintenance is presented. Its key modules include WordnetLoom editor, Wordnet Tracker and JavaScript Graph. They offer a number of functionalities that allow solving problems on every stage of building, editing and aligning wordnets by teams of lexicographers working in parallel. The experience collected in recent years has allowed us to refine applications and add new modules to provide the best user experience in a reliable and easily maintainable way.

1 Introduction

Wordnet is not yet another electronic dictionary. It is a complex lexico-semantic network. Its construction, especially when done manually by a team of lexicographers, and its further editing and/or aligning with other resources requires very advanced and flexible tools. Such tools should offer the possibility of simultaneous work of many team members on the same lexicon (a wordnet for a particular language), simultaneous work of different teams on different lexicons, and the subsequent manual or semi-automated linking of the constructed resources.

Dictionary compiling tools are mostly designed as complex XML editors such as, for instance, Lexonomy (Měchura, 2017). This approach may not be beneficial in working with graph-like structures. Therefore, several dedicated tools have been designed and are currently used by different wordnet teams e.g DEBVisDic (Horák et al., 2006), sloWTool (Fišer and Novak, 2011). Visualisation of wordnet graphs in most tools follows a radial pattern: a synset in focus is presented in the middle and all links, irrespectively of their types are placed radially around the central element, e.g. *sloWTool* or *WordTies* (Pedersen et al., 2012). *GernEdiT* (Henrich and Hinrichs, 2010) offers visualisation of the wordnet structure in the range selected by the user, but it is hierarchical and focused mainly on hypernymy. Moreover, the visual presentation does not allow for direct editing of the structures. WordnetLoom stands out of the remaining tools, because it offers a graph-based visualisation of wordnet data and provides entirely different workflow based on the direct interaction with graph nodes. In this paper, we will present the most recent development of WordnetLoom and progress in relation to earlier releases. We have improved the graphic design for better user experience and implemented the lexical unit graph visualisation.

Both dictionary making and wordnet building are usually carried out by teams of lexicographers and/or developers. Collaborative work, especially in distributed teams working from , requires control tools to provide quality assurance and development progress. In-built auditing/change backlog feature is often absent in these systems and data versioning is handled by external VCS[1] software or done manually. The newest version of WordnetLoom is interconnected with the Wordnet Tracker module which provides additional feedback channel for lexicographers to enrich their workflow. Every activity of each lexicographer is registered and can be monitored by a senior lexicographer. This paper will showcase how auditing and monitoring can be handled.

We will also present a new web-related module, namely JavaScript Graph. JavaScript Graph module is an answer to user needs and provides the possibility of embedding graph visualization to existing websites or applications.

[1] VCS - Version Control System e.g. Git, Subversion

In this work, we will present the key modules that are part of a collaborative system for wordnet construction and maintenance including WordnetLoom editor, Wordnet Tracker and JavaScript Graph.

2 WordnetLoom Demo

Up to version 1.68, WordnetLoom was a standalone java fat client application directly connecting to its database with all logic contained on the client side. Such approach ensures that scaling of the application could only be possible by scaling the database server, in this case MySQL[2]. In order to meet the growing numbers of users and challenges in providing dedicated endpoints not only for the client editor application, but also for other external applications, web pages or mobile applications, all business logic was extracted to a separate application built on top of JEE8[3] framework. The application is responsible for data validation, data auditing, user activity monitoring, user management and data processing. It provides a communication channel via REST API (Fielding, 2000) in the form of Siren[4]-like hypermedia specification. Scaling of the application itself is done by docker-compose[5] replicas, while database scaling can be achieved by replication configurations where at least two databases are available. Master database configuration is optimized for writing and slave databases have configuration optimized for high performance reading. Further scaling can be ensured by introducing new slave database nodes for each distinct consumer such as a mobile application or a web page.

The main consumer of the API is a thick client in the form of WordnetLoom Editor java application (main application workspace presented at Fig. 1a) which has been slimmed down and does not contain essential business logic which reduces it to the role of a simple REST client. It enables advanced search functionalities and basic CRUD[6] operations on typical core objects being part of the semantic structure such as synset, sense (see sense editing properties Fig. 1b), sense relation, synset relation, and relation type. From the editor level, the user with administrator privileges can modify and add elements to dictionary entities such as: part of speech, domain, register (see editing dictionaries Fig. 1c) and adding or editing types of semantic relations (see editing relation types Fig. 1d). The main advantage of the application is the possibility of working with visualization in the form of a graph, which provides quick and easy navigation and simplifies the creative process. Due to the fact that the Editor has recently undergone a major architectural transformation, it has allowed for even simpler modifications and easier addition of new components, such as in the case of implementing the extended semantic description panel for Dictionary of Polish Borrowings in Yiddish[7] (see Fig. 1e). Also within this project we have created a graph visualization of lexical units which has become the part of a core application.

3 Wordnet Tracker Demo

An important aspect of the process of building and maintaining wordnet is the ability to monitor changes made by team members. It is made possible by the Wordnet Tracker module which provides tracking of user activity (see Wordnet Tracker dashboard Fig. 2) in terms of the number of lexical units, synsets and semantic relations entered (see Fig. 3 for synset relation changes). Through this application, the lexicographer has also access to the full history of changes that have been made within a given lexical unit (see Fig. 4 for current changes of lexical units). All changes in the synset structure are presented in Fig. 5 where the left side column displays the current synset state, while the right side column shows all changes in the synset elements. The user as well as the coordinator have access to current changes in real time for constant monitoring. This functionality turned out to be particularly valuable when working with new, inexperienced lexicographers. The application administrator has the possibility to create diagnostic queries within lexicons or even within the entire dataset, as well as to create statistic queries. In both cases the generated query results are available for download in the form of files. Wordnet Tracker also provides basic user management panel where the privileged user can add new users, reset passwords or restrict user access to chosen lexicons.

[2] https://www.mysql.com/

[3] Java Enterprise Edition 8 specification https://javaee.github.io/javaee-spec/

[4] https://github.com/kevinswiber/siren

[5] https://docs.docker.com/compose/overview/

[6] CRUD are four basic functions of persistent storage (such as create, read, update and delete)

[7] https://polonjid.wn.uw.edu.pl/?lang=en

4 JavaScript Graph Module Demo

Presenting work results in the form of a graph visualization outside WordNet Loom editor environment is possible now by a created javascript module. The module tries to faithfully preserve the navigation functions as in the WordNet editor, but at the same time gives the possibility to adjust the color scheme and nodes style to the host application/page design. The presentation data model is fetched from the WordnetLoom server via the REST endpoint and the D3.js[8] library with custom modifications handles graph visualization and user interaction. The module is constructed in such a way so as to allow easy embedding in other applications, such as a mobile application or a website. A very good example can be the main page of plWordNet[9] where the module is used in the form of a pop-up window or as a full scale central element of the website presented at the online Dictionary of Polish Borrowings in Yiddish[10](see Fig. 6). Simple library import and basic configuration will allow to present wordnet lexicon as graph visualization on every platform where JavaScript is supported.

5 Conclusions and Further Works

This concludes our brief description of each module. We have seen that the combination of presented tools offers solutions to common tasks and problems encountered while building wordnets particularly by distributed teams. We will continue to be open-source software licensed under GNU LGPL 3.0. The source code is hosted in GitHub repository(https://github.com/CLARIN-PL/wordnetloom).

We will continue to actively develop presented tools over the next years focused on adding new functionalities based on the needs of users. We will also direct our development towards a reliable, fully-flagged web-based system and we will strive to continue to simplify system deployment by an extensive use of docker[11] containers.

Acknowledgment

The work co-financed as part of the investment in the CLARIN-PL research infrastructure funded by the Polish Ministry of Science and Higher Education and the project funded by the National Science Centre, Poland under the grant agreement No UMO-2015/18/M/HS2/00100.

[8] D3.js is a JavaScript library for manipulating documents based on data

[9] http://plwordnet.pwr.edu.pl/wordnet/

[10] http://polonjid-dictionary.clarin-pl.eu

[11] https://www.docker.com/

References

Roy Thomas Fielding. *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, 2000. AAI9980887.

Darja Fišer and Jernej Novak. Visualizing sloWNet. In *Proceedings of eLex*, pages 76–82, 2011. URL `http://elex2011.trojina.si/Vsebine/proceedings/eLex2011-9.pdf`.

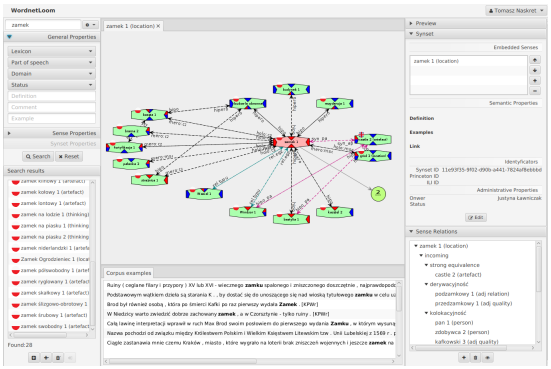
Verena Henrich and Erhard Hinrichs. GernEdiT – the GermaNet editing tool. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, May 2010. European Language Resources Association (ELRA). ISBN 2-9517408-6-7.

Aleš Horák, Karel Pala, Adam Rambousek, and Martin Povolný. DEBVisDic — first version of new client-server wordnet browsing and editing tool. In *Proceedings of the Third International WordNet Conference — GWC 2006*, pages 325–328. Masaryk University, 2006.

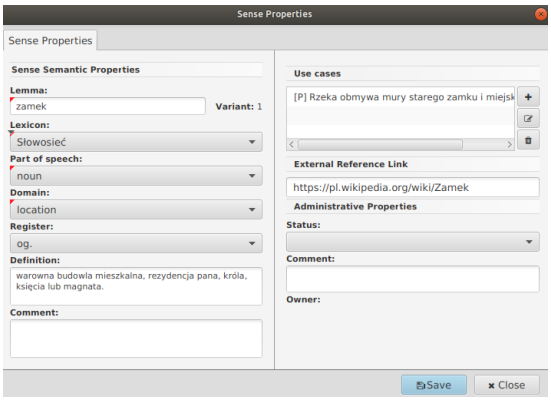
M. B Měchura. Introducing Lexonomy: an open-source dictionary writing and publishing system. In *Electronic Lexicography in the 21st Century: Lexicography from Scratch. Proceedings of the eLex 2017 conference, 19-21 September 2017, Leiden, The Netherlands.*, 2017. URL `https://www.lexonomy.eu/docs/elex2017.pdf`.

B.S. Pedersen, L. Borin, M. Forsberg, K. Lindén, H. Orav, and E. Rögnvalssson. Linking and validating nordic and baltic wordnets – a multilingual action in META-NORD. In *Proceedings of 6th International Global Wordnet Conference*, pages 254–260., Matsue, Japan., 2012.

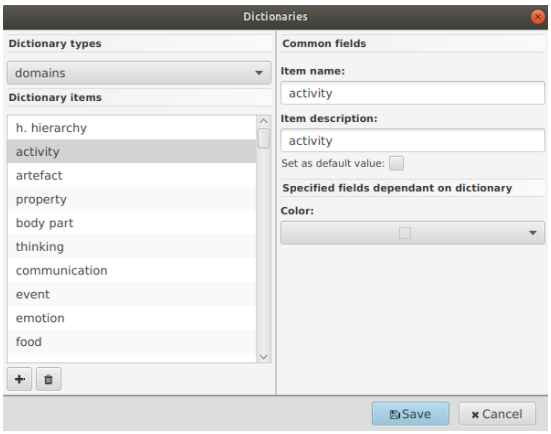
Figure 1: Key windows in WordnetLoom



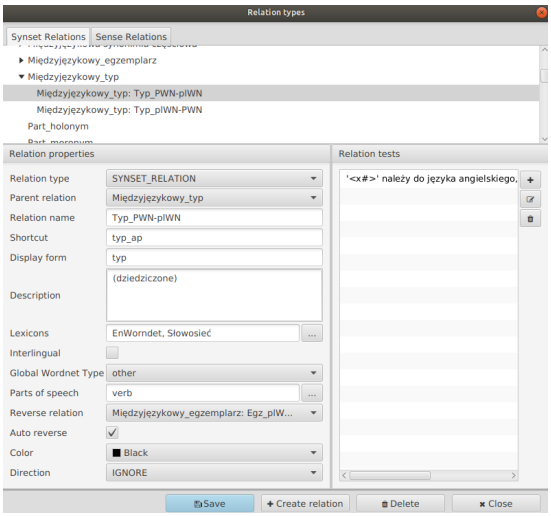
(a) Application main workspace.



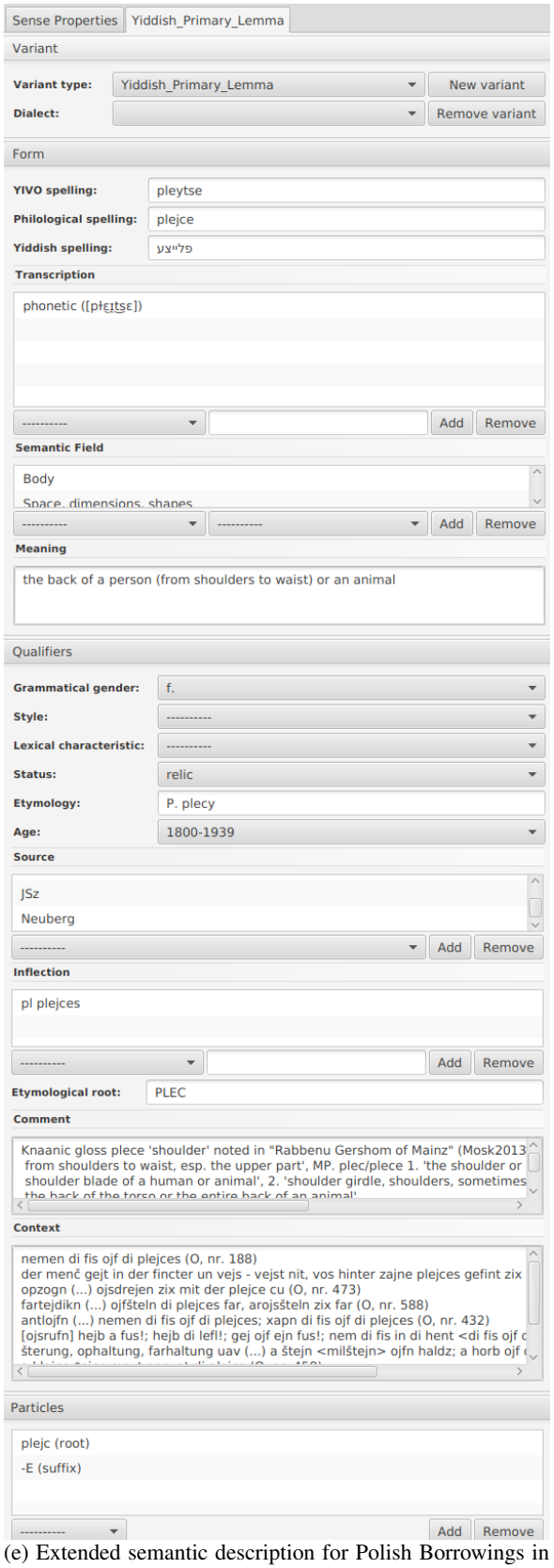
(b) Sense properties window.



(c) Dictionaries window.



(d) Relation types window.



(e) Extended semantic description for Polish Borrowings in Yiddish dictionary.

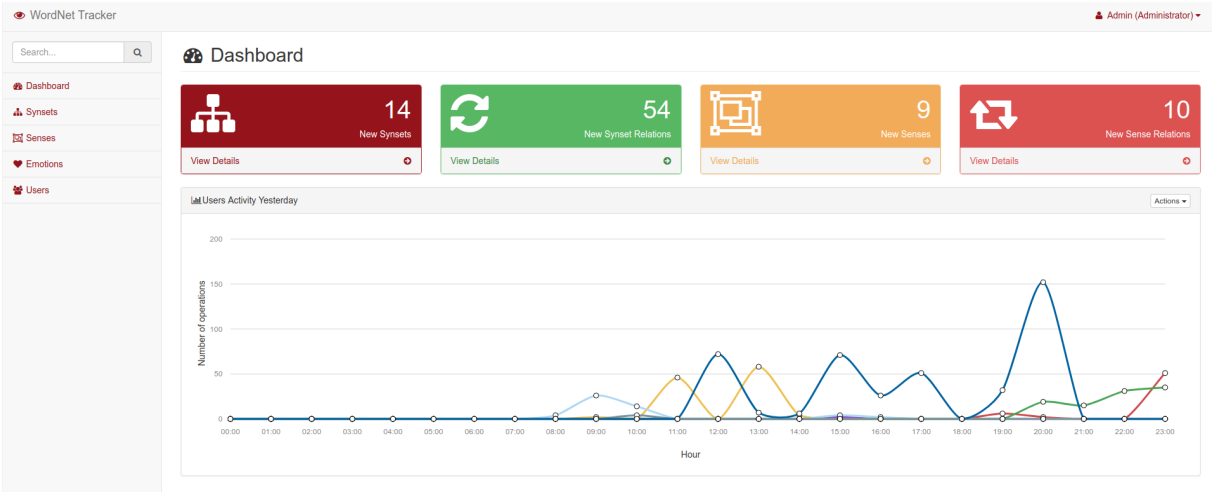
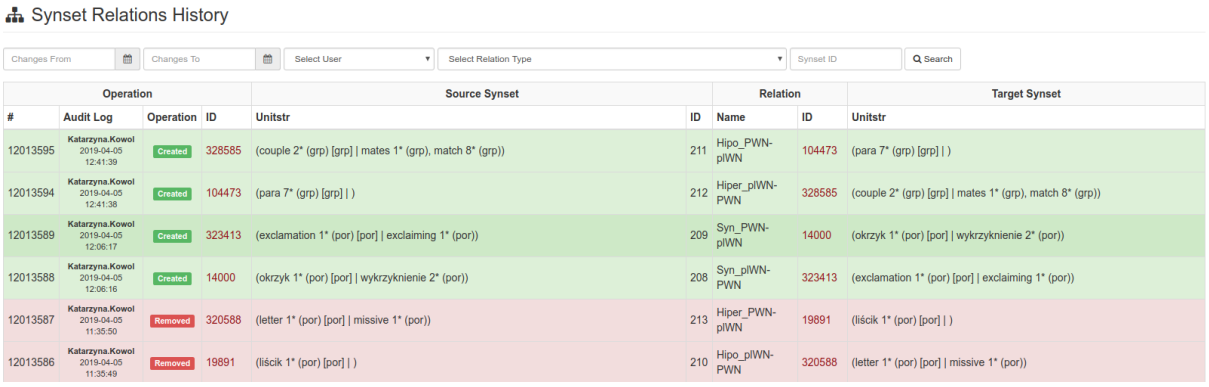
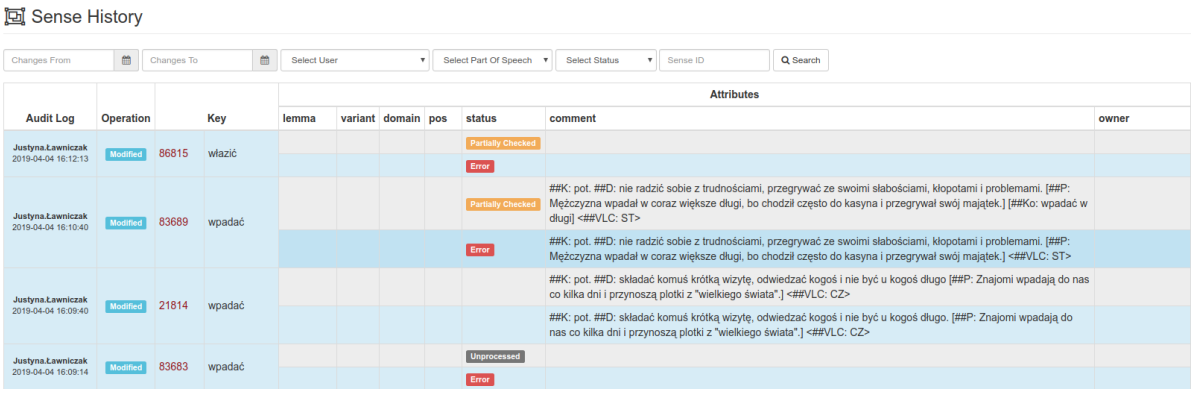


Figure 2: Tracker dashboard.



Operation				Source Synset		Relation		Target Synset
#	Audit Log	Operation	ID	Unitstr	ID	Name	ID	Unitstr
12013595	Katarzyna.Kowol 2019-04-05 12:41:39	Created	328585	(couple 2* (grp) [grp] | mates 1* (grp), match 8* (grp))	211	Hipo_PWN-plWN	104473	(para 7* (grp) [grp] |)
12013594	Katarzyna.Kowol 2019-04-05 12:41:38	Created	104473	(para 7* (grp) [grp] |)	212	Hiper_plWN-PWN	328585	(couple 2* (grp) [grp] | mates 1* (grp), match 8* (grp))
12013589	Katarzyna.Kowol 2019-04-05 12:06:17	Created	323413	(exclamation 1* (por) [por] | exclaiming 1* (por))	209	Syn_PWN-plWN	14000	(okrzyk 1* (por) [por] | wykrzyknienie 2* (por))
12013588	Katarzyna.Kowol 2019-04-05 12:06:16	Created	14000	(okrzyk 1* (por) [por] | wykrzyknienie 2* (por))	208	Syn_plWN-PWN	323413	(exclamation 1* (por) [por] | exclaiming 1* (por))
12013587	Katarzyna.Kowol 2019-04-05 11:35:50	Removed	320588	(letter 1* (por) [por] | missive 1* (por))	213	Hiper_PWN-plWN	19891	(liścik 1* (por) [por] |)
12013586	Katarzyna.Kowol 2019-04-05 11:35:49	Removed	19891	(liścik 1* (por) [por] |)	210	Hipo_plWN-PWN	320588	(letter 1* (por) [por] | missive 1* (por))

Figure 3: Synset relations changes history view.



Audit Log	Operation	Key		Attributes						
				lemma	variant	domain	pos	status	comment	owner
Justyna.Lawniczak 2019-04-04 16:12:13	Modified	86815	włazić					Partially Checked		
								Error		
Justyna.Lawniczak 2019-04-04 16:10:40	Modified	83689	wpadać					Partially Checked	##K: pot. ##D: nie radzić sobie z trudnościami, przegrywać ze swoimi słabościami, kłopotami i problemami. [##P: Mężczyzna wpadał w coraz większe długi, bo chodził często do kasyna i przegrywał swój majątek.] [##Ko: wpadać w długi] <##VLC: ST>	
								Error	##K: pot. ##D: nie radzić sobie z trudnościami, przegrywać ze swoimi słabościami, kłopotami i problemami. [##P: Mężczyzna wpadał w coraz większe długi, bo chodził często do kasyna i przegrywał swój majątek.] <##VLC: ST>	
Justyna.Lawniczak 2019-04-04 16:09:40	Modified	21814	wpadać						##K: pot. ##D: składać komuś krótką wizytę, odwiedzać kogoś i nie być u kogoś długo [##P: Znajomi wpadają do nas co kilka dni i przynoszą plotki z "wielkiego świata".] <##VLC: CZ>	
									##K: pot. ##D: składać komuś krótką wizytę, odwiedzać kogoś i nie być u kogoś długo. [##P: Znajomi wpadają do nas co kilka dni i przynoszą plotki z "wielkiego świata".] <##VLC: CZ>	
Justyna.Lawniczak 2019-04-04 16:09:14	Modified	83683	wpadać					Unprocessed		
								Error		

Figure 4: Senses changes view.

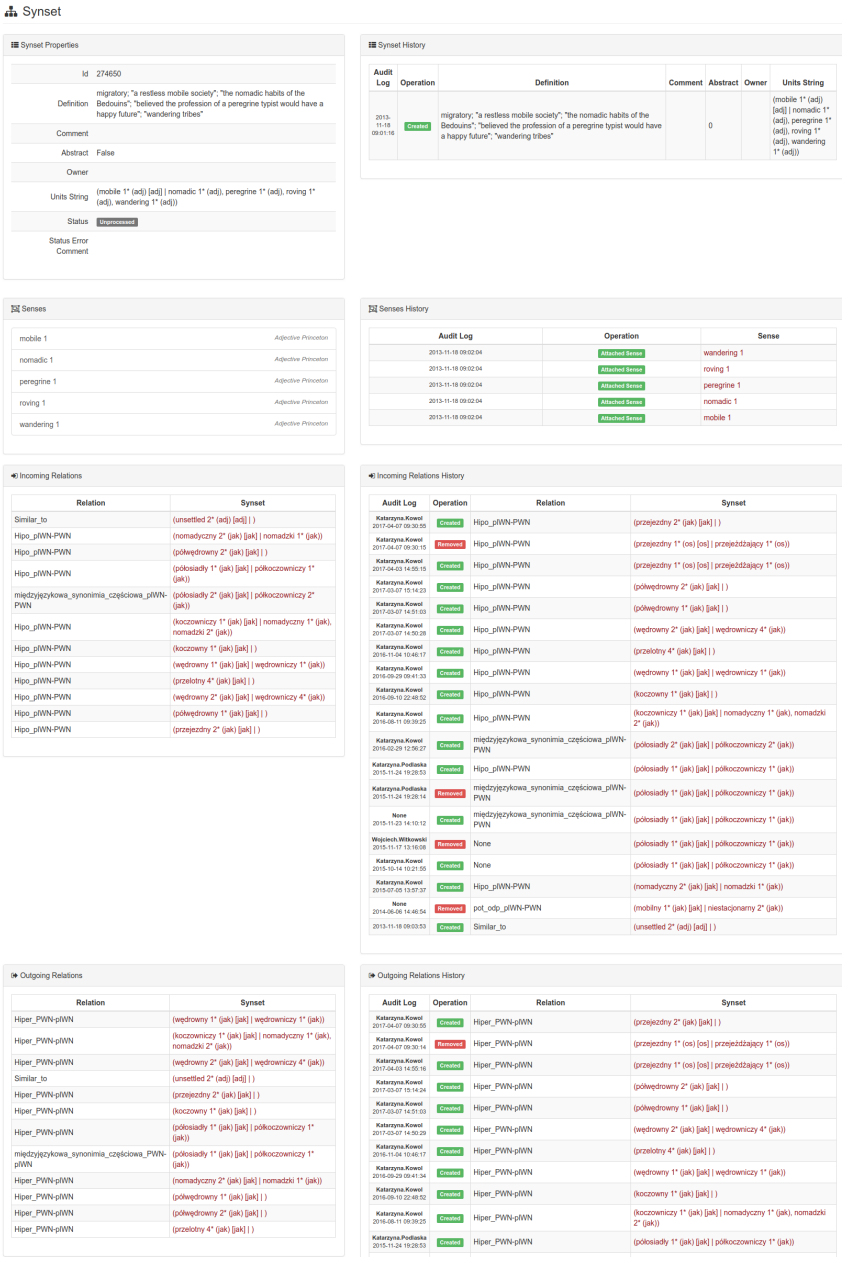


Figure 5: Selected synset history view.

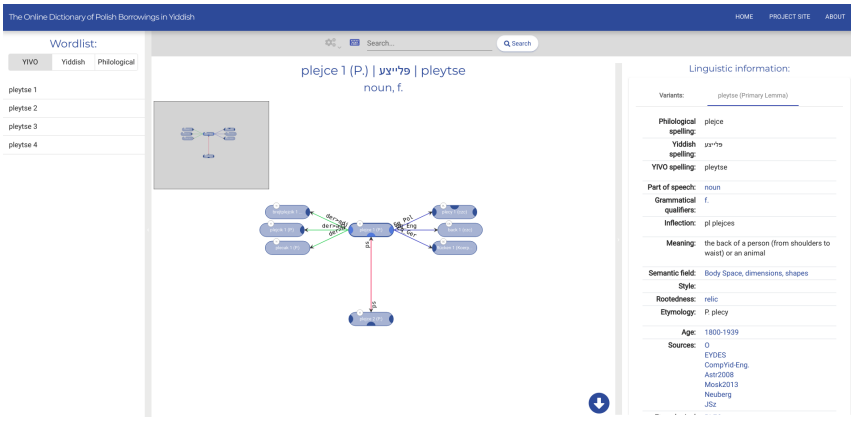


Figure 6: Example of embedded java script visualization module.