

**Citation:** Piasecki, M., Walkowiak, T., Rudnicka, E., & Bond, F. (2018). Lexical platform — the first step towards user-centred integration of lexical resources. *Cognitive Studies / Études cognitives*, 2018(18). <https://doi.org/10.11649/cs.1811>

MACIEJ PIASECKI<sup>1,A</sup>, TOMASZ WALKOWIAK<sup>2,B</sup>, EWA RUDNICKA<sup>1,C</sup>  
& FRANCIS BOND<sup>4,D</sup>

<sup>1</sup> Faculty of Informatics and Management, Wrocław University of Science and Technology, Wrocław, Poland

<sup>2</sup> Faculty of Electronics, Wrocław University of Science and Technology, Wrocław, Poland

<sup>3</sup> School of Humanities, Nanyang Technological University, Singapore

<sup>A</sup>[maciej.piasecki@pwr.edu.pl](mailto:maciej.piasecki@pwr.edu.pl) ; <sup>B</sup>[tomasz.walkowiak@pwr.edu.pl](mailto:tomasz.walkowiak@pwr.edu.pl)

<sup>C</sup>[ewa.rudnicka@pwr.edu.pl](mailto:ewa.rudnicka@pwr.edu.pl) ; <sup>D</sup>[bond@ieee.org](mailto:bond@ieee.org)

# LEXICAL PLATFORM — THE FIRST STEP TOWARDS USER-CENTRED INTEGRATION OF LEXICAL RESOURCES

## Abstract

The paper describes the Lexical Platform — a means for lightweight integration of independent lexical resources. Lexical resources (LRs) are represented as web components that implement a minimal set of predefined programming interfaces. These provide functionality for querying and generate a simple, common presentation format. Therefore, a common data format is not needed and the identity of component LRs is preserved. Users can search, browse and navigate via resources on the basis of a limited set of anchor elements such as base form, word form and synset id.

**Keywords:** lexical resources; wordnet; interoperability of lexical resources

## 1 Introduction

With the advent of the digital age, more and more lexical resources (LRs) are built for natural languages. They describe such aspects of lexical systems as lexico-semantic relations, valency frames, collocations, and inter-lingual equivalences, among others. Contrary to traditional paper dictionaries, they have no size limitations and offer new possibilities of data presentation. Thus, their potential of use is extremely broad subsuming linguistic studies, language teaching, human and machine translation, and natural language processing. Surprisingly, the growth in number and coverage of LRs has not yet resulted in their widespread use in research, commercial or popular applications. There are a couple of reasons for such state of affairs. LRs are usually products of different research projects, hence based on different models and encoded in different formats. They are spread across the web. Even if the resources can be found in some virtual catalogues like

CLARIN VLO,<sup>1</sup> usually every individual LR has to be accessed separately and via a dedicated browsing and searching system. Still, for such uses, we need only limited knowledge about an LR: what kind of elements can be searched and how to present the query of search results to users. Some LRs are inter-connected (e.g. many wordnets within the Open Multilingual Wordnet: Bond & Foster, 2013), but many are only available via their home interface in a native format. The reasons for that are often lack of funds and goodwill to cooperate, sometimes restricted licences. LR developers are afraid that their resource may lose part of its visibility once it becomes a member of some common platform (or a conglomerate of resources). Despite these reservations, there is clearly a need for some kind of integration of the already existing LRs that would guarantee keeping the identity and visibility of an individual resource. Our answer to this need is the Lexical Platform (LexP).

The Lexical Platform is meant to be a virtual (storage) place for aggregating different types of LRs as separate individual components in an inter-connected system, a kind of complex LR. We assume that the knowledge provided for LRs must be minimal and no common format should be required to make the construction of the Platform feasible. The Platform should be open to all types of LRs, but wordnets are in focus since they are usually very large resources, providing rich description, but are not so easily accessible to users.

## 2 Related work

There are three main problems in linking different types of LRs: no common format (even for wordnets), different models (also for wordnets) requiring different interpretation from the point of view of applications, and, finally, different solutions for technical aspects of storing, accessing and linking the data within LRs.

### 2.1 Formats and standards

One commonly accepted database format for all LRs could solve the problem of their inter-linking. Several standards were proposed, but none has gained an overwhelming coverage so far. Implementations of RDF for wordnets were proposed, but used only for single wordnets, e.g. Princeton WordNet (PWN) (Fellbaum, 1998) and DanNet (Pedersen, Nimb, Asmussen, Sørensen, Trap-Jensen, & Lorentzen, 2009). Several implementations of Lexical Markup Framework (LMF), a generic ISO standard, were proposed for wordnets, e.g. WordNet LMF from the KYOTO project, henceforth called KYOTO LMF (Vossen, Soria, & Monachini, 2013), GermaNet LMF (Henrich & Hinrichs, 2010), UBY LMF (Eckle-Kohler, Gurevych, Hartmann, Matuschek, & Meyer, 2013). However, KYOTO LMF focuses only on the representation of synsets, while the two other formats also do not allow for the full representation of the data a wordnet can provide, e.g. plWordNet (Maziarz, Piasecki, Rudnicka, Szpakowicz, & Kędzia, 2016). In addition, these formats are not designed for morphological or valency information.

Lemon (McCrae, Montiel-Ponsoda, & Cimiano, 2012) was proposed as an ontology-based representation for lexicons and machine-readable dictionaries and as a means of linking them to the Semantic Web and the Linked Open Data cloud. A Lemon-based representation is still too much focused on PWN, it cannot represent many elements present in different wordnets, but its various applications show its potential as a candidate for the future ‘common format’. The main obstacle for the existing formats is the lack of effective means for expanding them with new elements of the data format in a way which does not hamper existing applications.

In human-oriented lexicography, lexicons are usually encoded as a tree, a hierarchical data structure of parent-child relations (Měchura, 2016). Many authors, such as Aguado-de Cea, Montiel-Ponsoda, Kernerman, and Ordan (2016), Klimek and Brümmer (2015), and Měchura (2016), propose to use graph representation.

---

<sup>1</sup><https://www.clarin.eu/content/virtual-language-observatory-vlo>

## 2.2 Platforms

UBY LMF<sup>2</sup> Eckle-Kohler et al. (2013) platform was built to integrate LRs on both structural and semantic level. Twelve LRs<sup>3</sup> were combined and interlinked into a complex system. However, all these LRs were first converted to one common implementation of LFM as a necessary common format. In fact, a set of new LRs has been created as a complex resource and stored locally inside the UBY LMF platform. This can be done only in the case of LRs on open licences or with a licence granted to the platform. There is only one type of anchoring element, namely *word senses*.

*CILI* that is *Collaborative Interlingual Index* is described as “a flat list of concepts” and is currently based on Princeton WordNet 3.0 set of synsets (Bond, Vossen, McCrae, & Fellbaum, 2016). It is intended to serve as an intermediary reference resource between wordnets of different languages within the *Open Multilingual WordNet* (OMW) (Bond & Paik, 2012). Currently, CILI has been initialised with a set of concepts corresponding to Princeton WordNet 3.0 and should be gradually expanded with concepts lexicalised in languages other than English. Every concept is described textually by a short definition in English. CILI will require consistency in the understanding of lexical and semantic relations among different languages. There will be persistent identifiers for CILI entries. Concepts will never be deleted, only deprecated or superseded. Candidates for new CILI concepts must be linked to a concept in its ‘mother’ wordnet by one of the well-known relations (hypernymy, meronymy, antonymy) and indirectly linked via this concept to the already existing CILI concept. CILI is available on an open licence. CILI can become a primary resource for linking other semantic resources, but it does not solve the problem of navigation across different resources. Moreover, it is mainly focused on linking lexico-semantic networks.

*OMW*<sup>4</sup> is an open platform aggregating wordnets of different languages indirectly linked via Princeton WordNet 3.0 (Bond & Foster, 2013). All wordnets are first converted to a common database format, the so called CILI LMF format. For some wordnets the conversion to CILI LMF is unidirectional, i.e. it does not allow to reconstruct the original structure of a wordnet due to the flattening of relation structure during the conversion and the impossibility to reconstruct them back. However, this problem is gradually disappearing with the inclusion of larger number of wordnets and the evolution of CILI LMF. In addition, some other non-relational information stored in many wordnets cannot be expressed in CILI LMF and this problem needs further investigation and discussion. OMW is intrinsically focused on wordnets as resources to be integrated and the incorporation model in which one single complex resource is built from the individual wordnets to be merged. In such a model only wordnets available on an open licence can be encompassed in practice.

*PANACEA*<sup>5</sup> (Bel, 2010) is an EU FP7 project aimed at building a system of language resources for the purposes of Machine Translation. The system of LRs was enhanced with a handful of tools. A wide range of resources for several European languages was developed and integrated, with one LMF-based common format chosen as the data format for dictionaries.<sup>footnote</sup>[http://www.panacea-lr.eu/system/deliverables/PANACEA\\_D3.4.pdf](http://www.panacea-lr.eu/system/deliverables/PANACEA_D3.4.pdf)

*The Language Grid*<sup>6</sup> is a multilingual service platform which enables registration and sharing of language services such as online dictionaries, bilingual corpora, and machine translators, with a mixture of restricted and open resources (Ishida, 2011). Users can construct a multilingual environment to support their activities by combining language services on the Language Grid and providers must write a wrapper around their resource. It is not widely used, perhaps because of the complexity of the interface and licensing.

*LEAP* (*Lexical Engine and Platform*)<sup>7</sup> is a commercial product, focused on multilingual dictionary data that are semantically combined with asymmetrical translation memory. It offers a REST

---

<sup>2</sup><https://dkpro.github.io/dkpro-uby/>

<sup>3</sup><https://www.ukp.tu-darmstadt.de/data/lexical-resources/uby/>

<sup>4</sup><http://compling.hss.ntu.edu.sg/omw/>

<sup>5</sup><http://www.panacea-lr.eu/>

<sup>6</sup><http://langrid.org>

<sup>7</sup><https://www.oxforddictionaries.com/our-story/leap>

API for developers. All the data come from the same single vendor and are encoded according to the same format.

*Léacslann* (Měchura, 2012) is a platform for working with sets of lexical entries of arbitrary structures. A collection of entries, called stocks, can be monolingual, bilingual, terminology database, a collection of proverbs or a set of references to other resources. However, it is assumed that all the entries were uploaded to Léacslann and are stored locally as a single resource.

*Lexonomy*<sup>8</sup> — a direct descendant of Léacslann — is a tool designed for writing and publishing dictionaries. Its entry consists of a lemma (word form), PoS, word sense defined by a simple textual description and sense usage examples. An entry description can be a mixture of text and marked elements (in-line XML markup) corresponding to different elements of its structure. The dictionary has a structure of a graph (Měchura, 2016). Nevertheless, in the system such graph is edited as a single resource, so the integration of several resources can only be done by merging.

To sum up, all existing solutions for the integration of lexical resources go towards merging them on the basis of a single common format of data representation. Such approach has two serious limitations. Firstly, we need to define a common format which is a very challenging task bearing in mind a large variety of resources. Secondly, only open resources can be merged, and even for the authors of open resources the idea of having their work dissolved in a new super resource may be daunting, as citations and recognition are important aspects in acquiring funding.

Obviously, a common data format is crucial for natural language processing applications. However, it is much less important for human users who treat LRs as dictionaries. Most LRs are intrinsically equipped with a presentation format comprehensible for non-technical users. We are going to capitalise on this fact in our proposal.

### 3 Basic assumptions

It is not easy to combine many heterogeneous LRs into one complex LR without a common format of data representation. Still, many LRs are already inter-linked on the basis of their content, and we can show them combined to the users. Taking this basic goal as a starting point, the idea of Lexical Platform has evolved from a handful of intuitions and observations.

Lexical Platform should group together different LRs as independent components, implemented as software modules, in order to minimise the dependency of the whole Platform on peculiarities of individual LRs. Only a minimal set of requirements should be imposed on LRs developers to make joining the Platform easier. Moreover, the identity of any single LR must be visible and preserved inside the Platform. It is important for LR developers that their LRs gain full recognition. It is crucial that Lexical Platform is not meant to become a ‘super-resource’, because that would discourage LRs developers from joining in.

Lexical Platform will promote the use of a limited set of common formats, but it will not enforce any specific data format on its components, even during the process of data exchange. Joining the Platform should be possible without the need of constructing format converters. Any component may be located in a freely selected network location. Neither the component, nor its LR data need to be physically copied to Lexical Platform. This can be a crucial feature from the point of view of the management of IPR and data protection. Some LR cannot be transferred outside their home institutions.

Thus, a component will be accessible via a limited set of Programming Interfaces (PIs). They can be implemented, e.g. as traditional Web Services (WSs) communicating by HTTP/HTTPS or as micro-services (4.1) communicating through the AMQP protocol (4.2). PI can be implemented as one separate WS, or several PIs can be provided by a single WS — this is a matter of detailed design decisions for Lexical Platform. Still, some minimal set of PIs must be specified and are required to be implemented by every component, to make the Platform operational and provide some basic level of usability. At the first approximation, a component’s PI (including PIs allowing

---

<sup>8</sup>[http://www.lexonomy.eu/\\_info/](http://www.lexonomy.eu/_info/)

to obtain the description of a component, access an element of the resource or get the visual representation of a resource element). However, all components can provide any number of additional PIs.

Lexical Platform is not intended to be a tool for changing the content of individual resources, or the links between them. It is meant only to be used for accessing a complex system of linked resources from a single access point, i.e. a kind of ‘meta-user-interface’. Lexical Platform will not be a system supporting the development of LRs, at least this will not be its prominent role.

The access to language data of the component LRs will be constrained in order to make the construction of Lexical Platform feasible. The components will encapsulate the data, i.e. the access to the data — the content of an LR — will be possible only via the PIs of the given components. Every component can provide data in any format, but some formats, e.g. *Lemon* (McCrae et al., 2012) (or its expansions), may be suggested as preferred ones. The construction of converters from native formats to a limited number of common formats will be promoted. Every component will be required to support addressing elements of Lexical Platform via *anchor elements* of limited and predefined types. The set of types will be specified by an ontology. Still, every component which is compliant with Lexical Platform can offer expanded methods of addressing LR elements.

The inter-linking of LRs via Lexical Platform components is a key issue. It will be exclusively based on the content of LRs and exploring the already existing possibilities. It is not realistic to expect extensive work on the side of LRs creators just for the needs of linking their components to the Platform. Each component should recognise references to elements of some limited set of types. Such elements serve as selected points by which the data from different components are anchored to the Platform and mutually inter-linked. Such selected data elements will be called *anchor elements*. Anchor elements should intrinsically originate from the construction of a given LR. They should be its characteristic elements by which users browse it or which users usually search for. In addition, anchor elements should also be those data elements that provide natural mapping to other LRs (or knowledge resources). The selection of anchor element types can be left to resource creators. However, if an anchor element is to be used by other resources, especially for inter-linking, the way of naming it must be known to the creators of those resources. Anchor elements of the following types are expected to be provided by different components<sup>9</sup>:

- *orth* (word, word form) — an inflected form, it can also be a multi-word expression
- *lemma* (also called literal in wordnets, canonical form, entry form, basic form) — a basic morphological form representing a set of inflected forms; it can also be a basic form of a multi-word lexical unit
- *lexical unit* (word sense) — a triple: ⟨lemma, Part of Speech, sense id⟩
- *synset*, represented by a synset identifier, (e.g. CILI identifier or the internal wordnet key)
- *frame* (syntactic and/or semantic), represented by an identifier
- *domain* (semantic field) represented by its name, (e.g. lexicographer file names from Princeton WordNet or WordNet Domains (Bentivogli, Forner, Magnini, & Pianta, 2004))
- *concept* — represented by an identifier, (e.g. concept identifiers from SUMO ontology (Pease, 2011) which several wordnets are linked to (Pease & Fellbaum, 2010; Kędzia & Piasecki, 2014))

On request, via (the function of) its PI, every component will provide a list of anchor elements it can recognise. The ontology of anchor element types will be created (or, preferably, selected from the existing ones) and maintained as the only central knowledge resource of Lexical Platform. Still, it must be a small-size ontology to keep the Platform open and flexible.

---

<sup>9</sup>This is an initial list to be (gradually) expanded during the development of the Platform.

The primary functionality of Lexical Platform will be aimed at non-technological users and will be close to the idea of *Federated Content Search* (FCS) of CLARIN<sup>10</sup> that allows to browse (search across) many corpora from a single access point. The query language of FCS is very limited in comparison to those of the majority of search engines of its component corpora. Nevertheless, FCS users are able to quickly check what kind of (text) material is available in hundreds of CLARIN network corpora.

(At the minimum), Lexical Platform should allow users to:

- learn about the component LRs and the range of information provided by them, especially those corresponding to some initial query;
- search across combined LRs on the basis of anchor elements supported by different components and browse LRs by lists of anchor elements retrieved from the components;
- browse and manually navigate across linked LRs on the basis of anchor elements;
- finally, find out how to access and download original resources and obtain information on how to browse different LRs in their native browsers.

To enable browsing LRs via Lexical Platform, all its components need to provide PIs that will generate presentation format for each anchor element. The presentation format will depend on a given component, but it should visually highlight anchor elements and make them respond to users' actions. When a user clicks on the anchor elements this action together with the anchor element name should be reported to Lexical Platform in order to facilitate interactive browsing. A range of formats can be considered as acceptable presentation formats: HTML, XML, SVG, etc. A selected format (or formats, as the Platform can support several of them) should be as simple and as popular as possible in order to simplify the construction of components. After the analysis of the existing systems, especially web-based systems, for browsing LRs, we decided to concentrate in the first prototype of Lexical Platform on HTML which is simple and commonly used. However, it has also one very serious drawback: it is used in very different ways by LR browsers.

Users should be able to list anchor elements described by a component and the whole Lexical Platform. To this end, we need PIs that implement a listing facility together with some kind of filtering e.g. by PoS, UPOS (Universal PoS), natural language, supertype/hypernym, or semantic domain etc. To support some of the search filters listed, an LR component has to have access to the meta-data of LR elements beyond the anchor types, but such information is available and used in browsers for most LR components. Lists retrieved from the components will be collected by Lexical Platform and presented to users as merged lists.

It would be hard to follow the changing versions of individual LR elements, so LR versions will be reported by the description PI of a component. In the first prototype of Lexical Platform we deliberately neglect this problem.

Without mapping to a common format, or at least to a limited number of formats, the support offered by Lexical Platform to technological users will be naturally limited. However, some functions can be already envisaged. Lexical Platform can be used for collecting data sub-structures describing specified anchor elements in native formats or some formats for which converters from the native formats are available. PIs for calculating similarity measures between anchor elements can be introduced. Other possible functions could provide, e.g., some statistics, clustering of elements, mapping texts onto substructures extracted from Lexical Platform components. Finally, we expect that with its growing popularity, Lexical Platform can create an environment stimulating the integration of LRs, also in terms of practical actions aimed at the convergence of formats.

---

<sup>10</sup><https://www.clarin.eu/content/federated-content-search-clarin-fcs>

## 4 Platform architecture

### 4.1 Lexical micro-services

Lexical Platform links diversified LRs in a flexible and autonomous way, i.e. each resource is preserved as a separate module and keeps its identity (the linked resources are not merged in one big ‘super-resource’). Such a strategy should help to convince a large group of resource creators to link their resources to the Platform.

The existing PIs for LRs are developed in different languages (Java, C++ , Python). Moreover, many of them (e.g. APIs for plWordNet *Walenty* (Przepiórkowski, et al., 2014)) store very large datasets. Therefore, the time of loading such a component is much longer than processing a single task. The solution is to run an LR component as a service with data loaded into memory. Each service runs its own process. The usage of services communicating with one another by lightweight mechanisms also solves the problem of a variety of technologies used by LR components since there is no need for tight integration. It results in a set of “cohesive, independent processes interacting via messages” (Dragoni et al., 2017). This is a definition of micro-services (Wolff, 2016), an architecture style following service-oriented (Bell, 2012) ideas that has recently started gaining wide popularity. The micro-service architecture will enable continuous development/deployment (Richardson, 2018) of Lexical Platform.

Each LR is represented inside the Platform as a separate micro-service. It is possible to run several instances of each LR micro-service to achieve larger throughput. This is important for LR PIs with large response time. A queuing system is used to distribute requests among micro-services. Each LR component is assigned its own queue. Lexical Platform micro-service will collect tasks from a given queue and send back messages when results are available. Such a solution facilitates effective scalability capabilities since a queuing system acts as a load balancer.

Every Lexical Platform component implemented as a micro-service provides a set of required PIs. The minimal set of required (obligatory) PIs encompasses the following functions:

- **getInfo** — delivers the resource and component meta-data (including license) and informs about the PIs provided by a component, facilitates component registration in Lexical Platform (the information can be provided in several languages, but obligatorily in English)
- **exist(*element*)** — checks if an anchor element exists in a given LR, returns the format of the element (native or HTML) and links to the element on the resource web page (if such exists), several ways of filtering are possible.
- **getNative(*element*)** — returns all possible descriptions of a specified resource anchor element
- **getHtml(*element*)** — generates a simple visualisation of a specified resource anchor element in HTML format that can be easily rendered in a web browser without the need of data interpretation,

The above list can be extended to cater for a specific need of an LR, e.g. it would be good to add to it the following function:

- **getResource** — returns URL/URLs to the zipped resource (with the data in a resource in a specific format/or formats).

Some of the above LR PIs functions require an element anchor as an argument. To allow different methods of addressing (as discussed in § 3) and achieve maximum flexibility a file system path like the method of LR element addressing is proposed. The Lexical Platform address is composed of elements separated by right slashes: i.e. */a/b/c/d/e*, where *a* is a name of an anchor type, while the remaining elements are subtypes, parameters and element value. For each type of an element (for example lemma, orth, synset id) specific subtypes and parameters are allowed. To give an example:

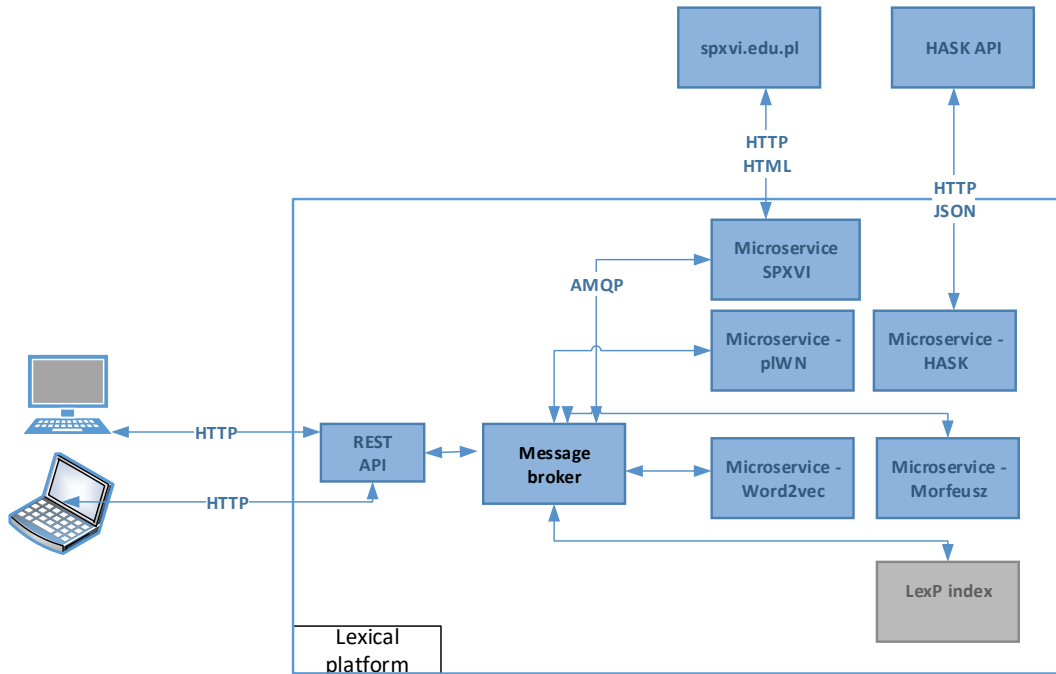


Figure 1: Lexical Platform architecture.

- */lemma/en/house* – refers to "house", a lemma in English;
- */lemma/pl/dom* – refers to "dom" (house in English), a lemma in Polish;
- */orth/pl/domu* – refers to "domu", a word form in Polish;
- */synset/plwordnet3.0/4782* – refers to a synset in plWordNet with an id equal to 4782;

## 4.2 System architecture

The architecture of Lexical Platform is presented in Fig. 1. We used the AMQP (Vinoski, 2006) protocol for lightweight communication with lexical micro-services and open source RabbitMQ (Videla & Williams, 2012) broker for a queuing system. AMQP protocol has clients for a large number of different software platforms as required by technologies used by LR PIs. In the proposed architecture an additional server grants the access from the Internet. It works as a proxy for the core system delivering synchronous, HTTP based REST API. Such approach allows for an easy integration with almost any kind of application including JavaScript ones.

We assume that all data (such as requests and responses) will be sent in JSON format. In the case when a given LR is not able to serialize a resource into JSON, the results in other formats (for example XML) will be encapsulated in JSON strings.

In addition, a Lexical Platform orchestrator is planned to be developed. It is meant to process all incoming requests to the Platform. No external application will have a direct access to any lexical micro-service. The orchestrator is aimed to:

- filter all wrong requests,
- add mapping between an external resource name and an internal micro-service name,



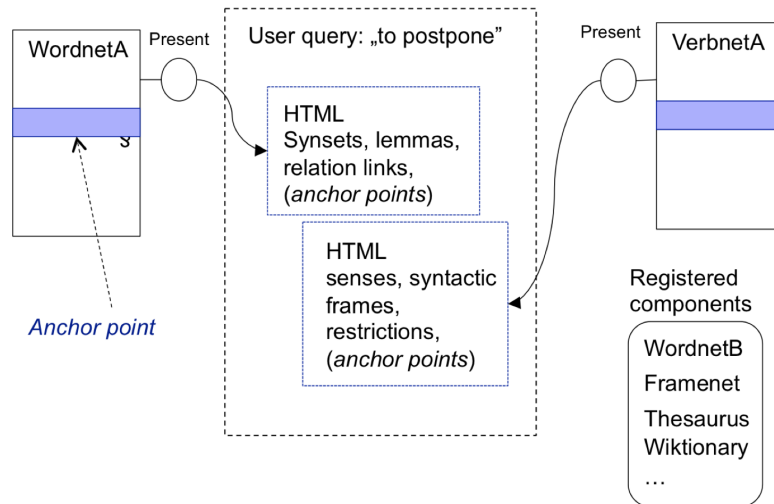


Figure 2: The illustration of the working of Lexical Platform. Modules that make available specific resources provide a presentation widget (here called HTML) for an element chosen by a user.

- send simple requests to a given type of a micro-service (lexical resource),
- process complex tasks (built on a sequence of calls to lexical resources):
  - results for a list of anchor elements,
  - results for all types of resources for a single element,
  - selected combination of resources or their parts in the form of a graph;
- process other tasks, for example:
  - listing of available resource types,
  - conversion of output formats,
  - access to the whole resource in a given, resource-specific format,
  - logging of external tasks and users data (IP, user names) for the Platform usage analysis;
- add prioritisation of tasks:
  - for example, a simple task will be performed faster than a request for a huge set of elements.

The Platform’s micro-services can be deployed on the central server of Lexical Platform or on servers of their suppliers (or authors, owners, etc.). If the external micro-service is not able to follow AMQP protocol, a specific, resource-oriented adapter (see. Fig. 2) may be developed and connect any external resource to the Platform. A resource adapter could include cache capabilities to speed up the resource access.

The proposed architecture also includes a service index. It functions as a simple database of micro-services names. At the start-up micro-service instances register themselves in a given queue and de-register themselves during the shutdown. Moreover, the RabbitMQ broker may invoke a micro-service health check to verify if an instance is able to handle requests (if not, the instance is removed from a list of queue consumers). The service index monitors the number of clients of each queue and provides a list of working components.

The RabbitMQ is able to work in a distributed way. Several instances of RabbitMQ could cooperate in different manners (via clustering, through federation, and by the use of the shovel). Therefore, it will be easy to distribute Lexical Platform among different data centres.

### 4.3 Central web application

The architecture described in the previous chapter focuses on programming access to LRs. To make the Platform accessible for individual (non-technical) users, a central web application needs to be developed. It will communicate with the core of the Platform by an HTTP and JSON based REST API. The user will be able to access any functionality provided by the Platform API. At present, the prototype presented in Section 5 offers only a simple browsing functionality. The results are displayed as interactive widgets. Each LR has a specific, JavaScript widget that graphically represents the requested resource. For example, for plWordNet, it can be an interactive graph showing the requested synset and its main relations. When a specific resource widget is not available (not yet developed), the generic one will be used. It will use a basic HTML result from a lexical micro-service (the result of `getHTML` function of lexical a micro-service). The widgets will not only display the content of an element, but they will also provide a set of hyper-links that will allow to select other elements and browse their content. Most LRs have references to themselves, but in Lexical Platform anchor elements refer to other resources. It is important to have LRs that will allow to link elements of different types or subtypes. For example, to allow to select a lemma in a different language or link a lemma with synsets. Such functionality is provided by wordnets. There is also a need to have some extra resources that will allow to map elements from a modern language to its old or middle version. The large number of anchor elements will allow to browse through different LRs in a similar way as we can browse through internet resources, for example Wikipedia. Moreover, the user will be able to easily browse through the Platform resources.

First of all, we plan to extend the path like method of addressing elements (4.1) with a functionality similar to wildcard functionalities (also known as "globbing") on file paths in Linux and POSIX operating systems. The Lexical Platform should interpret `*` (match one of more characters), `?` (match a single character) and `[` (begin a character range with `!` to state excluding information). To give an example:

- `/lemma/en/h*` — refers to lemmas in English that start with h;
- `/lemma/p?/d*m` — refers to lemmas that start with d, finish with m in languages whose acronyms start with p (i.e. Polish or Portuguese);
- `/orth/en/[cb]at` — refers to word forms in English such as cat or bat;
- `/lemma/en/[!d]ow` — refers to English lemmas whose first letter is not d and others

Moreover, it is planned to extend element anchoring by a mechanism that allows to filter out search results. The mechanism will utilise a query similarly to the approach taken in the uniform resource locator (URL). Strictly speaking, an anchor element with Lexical Platform server name forms the correct URL. The mechanism will be based on adding the question mark at the end of an anchor element. It will be followed by a query string, with filters separated by the ampersand, `&`. We assume that each LR will have a set of tags assigned to it. The tags will describe LR features such as the type of an LR (for example, dictionary) and its content (for example, technical vocabulary). The query-like mechanism will allow to select elements only from an LR that includes or does not include a given tag. To give an example:

- `?tag=historical&tag!=technical` — will show the results only from LRs with a tag 'historical' and without the tag 'technical'

Lexical Platform GUI will allow to form queries by a set of buttons that will list available tags for currently selected LRs. We also plan to define a set of recommended tags that could be assigned to LRs. In this way the idea of Federated Corpus Search will be expanded to a kind of a *federated search for lexical resources*. Technical users, apart from the easier download of all resources in accordance with their licences, will be able to download a combination of selected resources, or their parts, as a graph.

## 5 Prototype

The very idea of Lexical Platform arose from the need to offer users access to a set of superficially inter-linked lexical resources without actually integrating them. Thus, we want to follow an agile approach and move directly from the level of concept to the level of implementation. By building the first prototype we wanted to learn how many potential pitfalls are hidden in the already existing technical solutions for LRs. For the first prototype we focused on LRs for Polish (presented in Sec. 5.1) and selected LRs directly linked to them. Most of them are very large, represent different types of LRs and have been electronically published using a range of different solutions. The level of their technical support also varies a lot, from a very advanced one, as in the case of plWordNet and OMW to almost none.

### 5.1 Scope: lexical resources related to Polish

For a start, we took the largest lexico-semantic resource, namely plWordNet 3.1 emo (called *Stowosieć* in Polish), manually mapped onto Princeton WordNet 3.1. However, since Princeton WordNet 3.0 has been linked to wordnets for many languages and for many wordnets these mappings have been utilised in OMW to show a large mesh of multilingual sense connections, we used OMW, instead of Princeton WordNet alone, as a component of the Platform. Among many other LRs for Polish, we chose only those accessible via web-based browsers and large enough to respond to many user queries and whose creators agreed to make them available via the Platform. As a side effect, we obtained a collection of comprehensive, but heterogeneous and bilingual LRs for Polish.

The set of LRs covered by the first prototype of Lexical Platform includes:

- plWordNet 3.1 emo (*Stowosieć*)<sup>11</sup> (Maziarz et al., 2016), — a very large wordnet of Polish<sup>12</sup>, substantial manual mapping to Princeton WordNet 3.1 and partial manual emotive annotation<sup>13</sup> (for 76k lexical units) (Zaśko-Zielińska & Piasecki, 2018) — three types of anchor elements: lemmas (191k), lexical units (286k) and synsets (220k),
- *Open Multilingual Wordnet* — a complex multilingual resource built by combining material from many wordnets, includes a significant portion of plWordNet 3.0 material extracted on the basis of mappings onto Princeton WordNet 3.1, so OMW functions as a bridge linking plWordNet to many other languages; OMW includes also Princeton WordNet expanded with *enWordNet* 1.0 — a significant manually built expansion ([20]) — a very large wordnet for English, plWordNet has been manually mapped onto it and vice versa — anchor elements: lemma (163k), lexical unit (215k), synset (124k),
- Walenty<sup>14</sup> (Przepiórkowski et al., 2014) — Polish valence dictionary, describing predicate-argument structures on the level of syntax and semantics, valency frames defined both for lemmas and lexical units, the latter correspond to a large extent to lexical units of plWordNet — anchor elements: lemmas (15k), lexical units, frames,
- Polimorf<sup>15</sup> — a very comprehensive morphological dictionary of Polish combined with SGJP<sup>16</sup> (Saloni, Woliński, Wołosz, Gruszczyński, & Skowrońska, 2012) — a grammatical dictionary of Polish — anchor elements: word forms ( $\approx 4M$ ), lemmas,

---

<sup>11</sup><http://plwordnet.pwr.edu.pl>

<sup>12</sup>In fact, currently the largest wordnet in the world and the largest dictionary of contemporary Polish in terms of lexical coverage.

<sup>13</sup>The annotations encompass sentiment polarity, basic emotions and fundamental human values, plus usage examples assigned to lexical units (senses).

<sup>14</sup><http://zil.ipipan.waw.pl/Walenty>

<sup>15</sup><http://zil.ipipan.waw.pl/Polimorf>

<sup>16</sup><http://sgjp.pl/leksemy/>

- NELexicon 2.0<sup>17</sup> — a lexicon of Polish Proper Names described by semantic categories — anchor elements: lemma ( $\approx 2.4\text{M}$ ), synsets (representing semantic classes of Proper Names),
- MWELexicon<sup>18</sup> — a lexicon of Polish Multi-Word Expressions described by their lexico-syntactic structures, all MWEs are treated as lemmas in plWordNet 3.1 emo — anchor elements: word form, lemma (54k),
- Hask<sup>19</sup> (Pęzik, 2014) — a set of collocation databases (i.e. collocation dictionaries extracted from large Polish and English corpora) — anchor elements: word form, lemma (150k for English).

In addition, the set of manually built LR components have been expanded with a kind of statistical *Similarity* thesaurus built on the basis of the very large plWordNetCorpus 10.0 (Piasecki, Czachor, & Kędzia, 2018).

## 5.2 Techniques of connecting LRs to the Platform

There are four methods of connecting LRs to the Platform:

- LR access by its own web page (for example the Dictionary of XVI-the Century Polish)
- LR with API inside Lexical Platform (for example plWordNet, Open Multilingual Wordnet)
- LR with its own REST API available via Internet (for example, Hask)
- LR following Lexical Platform REST API hosted by an LR owner

In the first case, Lexical Platform is responsible for hosting an LR. In the case of a new resource, there is a need to develop a micro-service that transforms an LR API to the Lexical Platform API. There is also a need to develop a JavaScript module that visualises the native LR representation in the Web Browser. The second method requires to build a micro-service that transforms an LR REST API to the Lexical Platform internal API. As in the previous case, we need to build a visualisation JavaScript module.

## 5.3 Technology

All LRs listed in the previous section can be viewed as a huge system, but they are not interconnected and users are forced to consult several different specialised browsers to obtain information provided by different LRs.

Lexical Platform web-based interface is a Single Page Application (Crane, Pascarello, & James, 2005) consisting of a main HTML page and a JavaScript library to access the Lexical Platform engine. When the user selects an element (by an URL address or through select boxes on the main page), the main web page asynchronously calls the index micro-service by the REST API. As a result, a list of LRs is shown on the screen (Fig. 3). The checked data form an array that includes information from each of referred LRs. Each LR is described by:

- link to the logo,
- short name,
- full name,
- short description (in HTML),

---

<sup>17</sup><http://hdl.handle.net/11321/247>

<sup>18</sup><http://hdl.handle.net/11321/274>

<sup>19</sup>[http://pelcra.pl/hask\\_en/](http://pelcra.pl/hask_en/)

- copyright info (in HTML),
- link to the element on the resource web page.

All textual information can be provided in different languages. The data form a map with a language as a key and text in a given language as values. The selected GUI language data is used, or English ones, if the data in the GUI language are not available. Any of the displayed LRs (Fig. 3) can be selected. When a user clicks on a resource its LR micro-service is called asynchronously. When the REST API responds, the JavaScript class specific for the resource is called that shows its content. Therefore, there is a need to develop a specific JavaScript class for each LR to show LR content.

However, there are dedicated web applications for each of these LRs. Moreover, for many of them, web services are provided, too. The applications already offer some presentation formats on the web. This technological background was a good starting point for the construction of components of the Lexical Platform prototype. In the case of LRs connected to Lexical Platform by fetching, the resource web page (the fourth type presented in 5.2), there is a need to parse the resource web page inside the micro-service. At first, the resource web page has to provide the access to elements by an HTTP address. There are LRs, mostly AJAX-oriented, which have no direct URLs to their elements (for example: <http://walenty.clarin-pl.eu/>). But even if there is a direct access, very often the element value (for example, a lemma) is not mapped directly to the URL. For example, the lemma "AARON" in the Dictionary of XVI-the Century Polish web page is accessed by the following URL: <https://spxvi.edu.pl/indeks/haslo/5142>. Therefore, there is also a need to have an index that would map LR internal links to Lexical Platform anchoring schemas (4.1). Having an index, we could download the referred element web page. The page (HTML code) has to be parsed and required div tags have to be selected. Headers, footers and menus included in the HTML page have to be removed. Moreover, CSS, fonts, figures, etc. used have to be properly linked (or copied and linked) to the Lexical Platform web page. When an LR web page uses JavaScript to perform actions they have to be mimicked. Usually, they have to be rewritten and added to the Lexical Platform web page code. Finally, all internal links available on an LR web page (links that refer to other LR elements) have to be translated into Lexical Platform links using the mentioned index. Such modified HTML is fetched by an LR micro-service to the web-based user interface.

As for other ways of connecting LRs to Lexical Platform (the first three listed in 5.2), there is a need to develop an LR visualisation JavaScript class. It has to implement a *showHTML* method that generates the HTML code showing the element based on the data (passed to the class constructor) received from an LR micro-service. It is important to present all resource links as anchors that follow the Lexical Platform addressing schema (4.1). The visualisations are mostly text-based and very often consist of tables of data (Fig. 6). However, other techniques could be used, too, for example, word clouds (Fig. 4).

## 5.4 Web-based user interface

As shown in Sec. 5.1, all LRs selected for the prototype are connected to a minimal set of anchor element types that provide links between all the resources. This fact is capitalised on in the prototype presented in Fig. 3.

First, the user selects the type and language<sup>20</sup> of the element they are going to search for. After the query has been processed, the Platform presents the list of components that support the selected type of anchor element and include information matching the provided query.

For instance, in Fig. 3, the Platform presents the list of LRs including descriptions of the lemma: *dom* ‘a house’ or ‘a home’. However, a user can also come up with an inflected *word form*

---

<sup>20</sup>Both could be guessed from a user query, but the specification makes the response of the Platform more accurate from the user’s perspective.

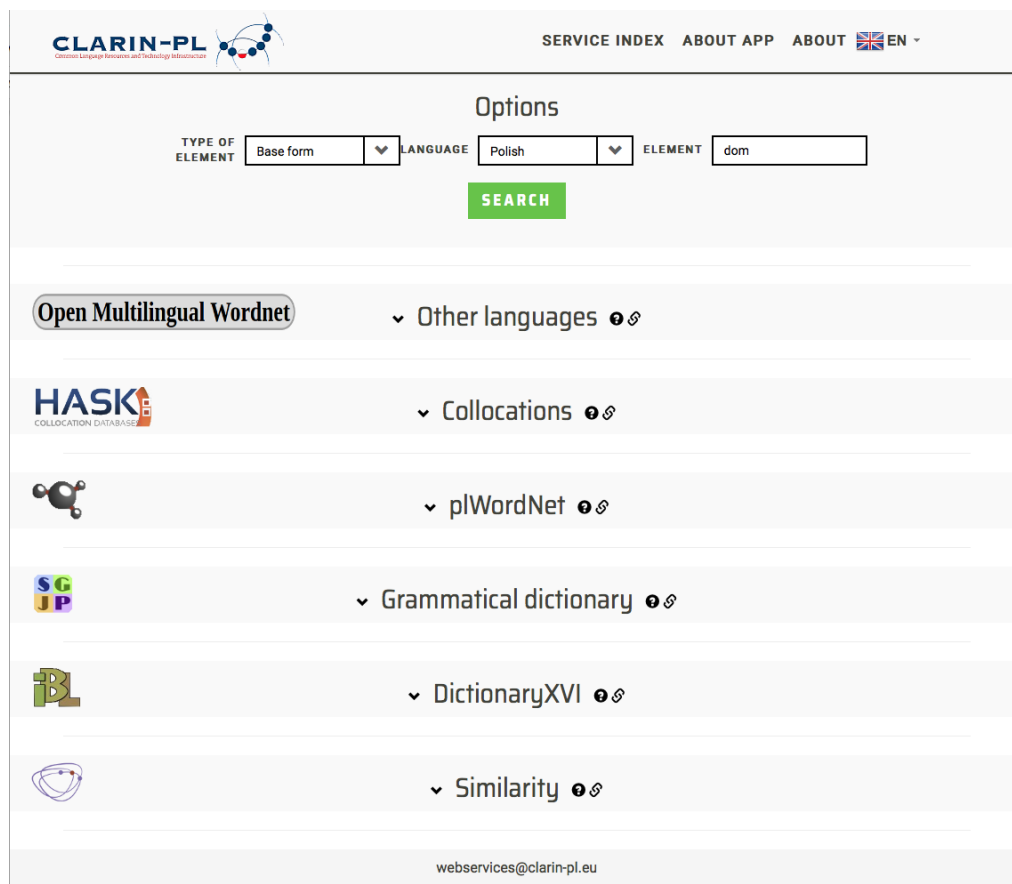


Figure 3: Lexical Platform prototype: User Interface.



Figure 4: Lexical Platform presentation of the results in the form of a word cloud.

The screenshot shows the CLARIN-PL Dictionary XVI entry for the Polish word "dom". At the top, there are navigation links for "SERVICE INDEX", "ABOUT APP", and "ABOUT" with a language selector set to "EN". The main title is "Dictionary XVI".

The entry for "DOM (7018) sB m" includes a list of source abbreviations and their corresponding counts: *dóm* (191), *dom* (84), *dom-* (182), *dóm-* (15); *dóm ForChr*, *BialKat* (6), *Strum*, *KochMarsz*, *KochPieł* (4), *KochW* (2), *SiebRozmysł* (2), *GoslCast* (2); *dóm KlerPow*, *KochTr*, *KochFr* (5), *KochDz* (2), *OstrEpit* (2), *JanNKar*, *JanNKarKoch*; *dóm*: *dom OpoczŻyw* (4 : 28), *MurzNT* (1 : 3), *Mqcz* (139 : 1), *OrzQuin* (1 : 3), *KochPs* (1 : 10), *KochWz* (1 : 2), *KochMBot* (1 : 1), *ZawJefi* (2 : 2), *KochFrag* (2 : 5), *SarnStat* (20:14); *dóm-* *Strum*; *dóm-* *dom-*, *BialKat* (9 : 6), *KochPieł* (12 : 1), *PudlFr* (8 : 2), *GórnTroas* (14 : 3), *KochFrag* (11 : 1), *SarnStat* (130 : 1).

The "Flexja" section contains a table with three columns: "sg", "pl", and "du".

	sg	pl	du
N	dóm	domy	
G	domu, dom	domów	domu
D	domowi, domu	domóm	
A	dóm	domy	
I	domem	domy, domami	
L	domu, domie	domiech, domach, domoch	
V	domie, domu		

Below the table is a long list of source abbreviations and their corresponding counts for the word "dom".

The "Znaczenia" section lists 7 meanings of the word "dom":

1. Budynek, najczęściej budynek mieszkalny; mieszkanie, stałe miejsce zamieszkania
  - Przez:
    - a) Niebo, raj jako mieszkanie Boga, przysłałe mieszkanie wieczne człowieka
    - b) Świeca, duma, ciało człowieka jako mieszkanie Boga lub Ducha Świętego; ciało jako mieszkanie duszy
2. Budynek przeznaczony na jakiś cel; zakład, instytucja
  - a. Świątynia, miejsce modlitwy
    - Przez: Kościół jako instytucja, ogół wiernych
3. Rodzina, mieszkańcy domu, ognisko domowe; gospodarstwo; majątkość, posiadłość
4. Ród, dynastia; pochodzenie
5. Strony rodzinne; kraj, ojczyzna, siedziba
6. Klasztor, dom zakonny
7. W dawnej astrologii pewna część nieba w stosunku do odpowiedniej części Ziemi

8. s. loc.  
\*\*\* Bez wystarczającego kontekstu

Synonimy: 1. *bud*, *buda*, *budowanie*, *chromina*, *gmach*, *kamienica*, *komora*, *mieszkanie*; 2. *scelita* *komora*, *praszownia*, *ratuz*, *warsztat*, *zbrojnica*, *zest.* »dom barwierski: *postrzygalsnia*, »dom chorujących: *szpital*, »dom ciemnicze: *zamknięcie*, »dom gołniczy: *gospoda*, *gościniec*, *kurczma*, »dom kurewskie: *zantai*, »dom nauki: *szkola*, »dom nierządny: *zantai*, »dom pastuszy: *pastyrnia*, »dom pielgrzymy: *szpital*, »dom polpolic: *gospoda*, »dom radny: *ratuz*, »dom robotny: *warsztat*, »dom sądny: *dawr*, *ratuz*, *urząd*, »dom sirotny: *szpital* *dziecinny*, »dom trędowatych: *szpital*, »dom wojtowski: *ratuz*, »zbojczy morskich *dom*: *kurczma*, »schadzka *lotrowska*; a. *budowanie*, *kaplica*, *kościół*, *kremlow*, *mieszkanie*, *pałac*, *przysłubek*, *zebranie*; 3. *familija*, *gminio*, *gospodarstwo*, *ojczyzna*, *pokolenie*, *potomstwo*, *rodzaj*, *rodzina*, *zebranie*; 4. *familija*, *herb*, *imię*, *narod*, *pokolenie*, *potomstwo*, *przewisko*, *rod*, *rodzaj*, *zawołanie*; 5. *ojczyzna*.

CYDOMA, [DOMOGOSPODARNY], DOMORÓDAK, DOMORODNY, DOMOSTWO

Figure 5: Lexical Platform prototype: Polish *dom* ‘a house’ or ‘home’ found in the Dictionary of XVI-th Century Polish linked to the platform.

*domu* ‘a house/home<sub>sng,loc.</sub>’, e.g. taken from a text, and then they can learn its potential *lemmas* – highlighted in the presentation format – from the SGJP component. Next, they can directly navigate, by clicking a link, to the description of lexical units (senses of the lemma) provided by plWordNet component, translations from OMW, collocations from HASK, and valency frames from Walenty, or the list of the most similar lemmas from the statistical Similarity thesaurus. In addition, the user can also check the elaborated description of *dom* in the Dictionary of XVI-th Polish, see Fig. 5. The dictionary entry provides different variants, descriptions of meanings, synonyms, references to sources, etc. Many lemmas occurring in the entry are marked on the fly in the presentation format as anchor elements of the lemma type and can be clicked to navigate across the Platform.

Wordnets are directly linked on the level of synsets and indirectly linked on the level of lexical units.<sup>21</sup> In Fig. 6 we can see the presentation (still limited in this prototype) of interlinked data coming from OMW and plWordNet.

<sup>21</sup>Every lexical unit (a word sense) belongs to exactly one synset, so from a lexical unit in one language via the synset link one can go to a lexical unit in another language. However, not all of lexical units from the target synset are good translation equivalents from the source lexical unit, cf Rudnicka, Bond, Grabowski, Piasecki, and Piotrowski (2018).

TYPE OF ELEMENT

LANGUAGE

ELEMENT

SEARCH

---

Open Multilingual Wordnet

^ Other languages
 ⓘ ⓘ

Meaning	List of words
a dwelling that serves as living quarters for one or more families	en : <a href="#">house</a> es : <a href="#">casa</a>
aristocratic family line	en : <a href="#">house</a> es : <a href="#">casa</a>
where you live at a particular time	en : <a href="#">home, place</a> es : <a href="#">casa</a>

---

∨ Collocations
 ⓘ ⓘ

---

^ plWordNet
 ⓘ ⓘ

Synset ID	Entity	Qualifier	Gloss	Example of use
4782	<a href="#">chałupa.3</a>	pot.	mieszkanie, dom	Dzisiaj nie mam czasu, muszę posprzątać chałupę, bo wieczorem mają przyjść do mnie znajomi. #P
	<a href="#">dom.2</a>	og.	mieszkanie, miejsce, w którym ktoś mieszka	Marzę o tym, żeby już być w domu. #P
	<a href="#">lokum.1</a>	og.	mieszkanie	Później, w wyniku jednej ze spraw Holmesa, Watson ożenił się, założył praktykę lekarską i wyprowadził się ze wspólnego lokum. #W

Figure 6: Lexical Platform prototype: browsing heterogeneous lexical resources for Polish *dom* ‘a house’ or ‘a home’.



## 6 Conclusions

The first prototype of Lexical Platform is available at <http://lexp.clarin-pl.eu/>. It currently includes six interlinked lexical resources, namely: Open Multilingual Wordnet — a resource built on the basis of data imported from many wordnets, plWordNet — a very large relational semantic dictionary of Polish (a wordnet of Polish), Grammatical Dictionary of Polish Language (SGJP), Dictionary of XVI-th century Polish, Hask Polish and English Collocations dictionaries, and a statistical Similarity thesaurus of Polish (built with the help of word2vec). We are working on linking Walenty — Polish valence frames dictionary, and negotiating permission for linking several other dictionaries.

The experience gained during the construction of Lexical Platform prototype allowed us to confront the general idea and initial assumptions with the reality of the existing LRs, technology for providing access to them and policy in managing them. The situation of LRs from the point of view of the technological support is very diversified. Some of them have excellent support with well-developed and maintained systems. In this case it is not difficult to construct wrapping microservice for a LR as a component for the platform. However, in many cases web browser application for a LR has a ‘static’ level of development, i.e. the application is kept running on the server, but as a kind of ‘black-box’, without any technical support for it. In the latter case the only possible option is to build a component installed on Lexical Platform, which is able to catch the HTML output of the dictionary application, parse it and transform in a way making anchor elements clickable. Nevertheless, this less elegant solution works according to the main assumptions for the Platform.

Lexical Platform aims at better promotion and accessibility of the existing LRs. Non-technical users can discover, browse and utilise LRs from a single access point. The Platform saves users time, enhances their use of LRs as a complex system that provides more comprehensive view and increases the outcome from the investment in the construction of the resources. It must be emphasised that Lexical Platform is not a new ‘super-resource’ collecting credits for the work of original LRs, it does not present itself as a new resource, so users can clearly see the identity of an LR they are using. Instead, all LRs linked to LexP preserve their identity. They can be stored in their original sites which also gives a possibility to provide restricted access protected by authentication.

Lexical Platform has something to offer to technical users, too. They cannot download all its component LRs in one common format, but they can view their content in a single place spending less time on the first manual inspection of individual LRs and learning about their usefulness. Lexical Platform may also serve as a good tool for the promotion of the need to develop one common format for LRs and converge descriptions of their models.

Linking new resources to Lexical Platform is relatively easy. One of the biggest problems is how to deal with different or new versions of the stored LRs. For instance, lexical units from *Walenty* valence dictionary are linked to lexical units from the older version of plWordNet, while Lexical Platform in its simplest form will present the latest version of plWordNet. There are several potential ways of handling this problem. The final solution needs to be in line with the simplicity of Lexical Platform design. All LR developers are kindly invited to join Lexical Platform as its co-developers and bring in their LRs. The works on Lexical Platform are to be carried out within CLARIN-PL — an open research infrastructure, a part of the Pan-European CLARIN research infrastructure.

## References

- Aguado-de Cea, G., Montiel-Ponsoda, E., Kernerman, I., & Ordan, N. (2016). From dictionaries to cross-lingual lexical resources. *Kernerman Dictionary News*, 24, 25–31.
- Bel, N. (2010). Platform for automatic, normalized annotation and cost-effective acquisition of language resources for human language technologies. PANACEA. *Procesamiento del Lenguaje Natural*,

- 45, 327–328. Retrieved from <http://journal.sepln.org/sepln/ojs/ojs/index.php/pln/article/view/826>
- Bell, M. (2012). *SOA Modeling Patterns for Service-oriented Discovery and Analysis*. Hoboken, NJ: John Wiley & Sons, Inc. <https://doi.org/10.1002/9781119202905.part1>
- Bentivogli, L., Forner, P., Magnini, B., & Pianta, E. (2004). Revising wordnet domains hierarchy: Semantics, coverage, and balancing. In *COLING 2004 Workshop on “Multilingual Linguistic Resources”, Geneva, Switzerland, August 28* (pp. 101–108). ACL. <https://doi.org/10.3115/1706238.1706254>
- Bond, F., & Foster, R. (2013). Linking and extending an open multilingual wordnet. In *51st Annual Meeting of the Association for Computational Linguistics: ACL-2013, Sofia* (pp. 1352–1362). Association for Computational Linguistics.
- Bond, F., & Paik, K. (2012). A survey of wordnets and their licenses. In C. Fellbaum, & P. T. J. M. Vossen (Eds.), *Proceedings of the 6th Global WordNet Conference, Matsue, Japan* (pp. 64–71). Brno: Tribun EU. The Global WordNet Association. Matsue. Retrieved from <http://web.mysites.ntu.edu.sg/fcbond/open/pubs/2012-gwc-wn-license.pdf>
- Bond, F., Vossen, P., McCrae, J., & Fellbaum, C. (2016). CILI: The collaborative interlingual index. In V. Barbu Mititelu, C. Forăscu, C. Fellbaum, & P. Vossen (Eds.), *Proceedings of the 8th Global Wordnet Conference* (pp. 280–289). Bucharest: Global Wordnet Association. <http://jiangbian.me/papers/2016/gwc2016.pdf>
- Calzolari, N., Choukri, K., Declerck, T., Loftsson, H., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., & Piperidis, S. (Eds.). *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, Reykjavík, Iceland, 2014*. ELRA. <http://www.lrec-conf.org/proceedings/lrec2014/index.html>
- Crane, D., Pascarello, E., & James, D. (2005). *Ajax in action*. Greenwich, CT: Manning Publications Co.
- Dragoni, N., Giallorenzo, S., Lafuente, A., Mazzara, M., Montesi, F., Mustafin, R., & Safina, L. (2017). Microservices: Yesterday, today, and tomorrow. In M. Mazzara & B. Meyer (Eds.), *Present and ulterior software engineering* (pp. 195–216). Springer. [https://doi.org/10.1007/978-3-319-67425-4\\_12](https://doi.org/10.1007/978-3-319-67425-4_12)
- Eckle-Kohler, J., Gurevych, I., Hartmann, S., Matuschek, M., & Meyer, C. M. (2013). UBY-LMF — Exploring the boundaries of language-independent lexicon models. In *LMF Lexical Markup Framework* (pp. 145–156). Hoboken, NJ: John Wiley & Sons, Inc. <https://doi.org/10.1002/9781118712696.ch10>
- Fellbaum, C. (Ed.). (1998). *WordNet: An electronic lexical database*. Cambridge, MA: MIT Press.
- Henrich, V., & Hinrichs, E. (2010). Standardizing wordnets in the ISO standard LMF: Wordnet-LMF for GermaNet. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)* (pp. 456–464). Beijing: Coling 2010 Organizing Committee. Retrieved from <http://www.aclweb.org/anthology/C10-1052>
- Ishida, T. (Ed.), (2011). *The language grid: Service-oriented collective intelligence for language resource interoperability*. Springer. <https://doi.org/10.1007/978-3-642-21178-2>
- Kędzia, P., & Piasecki, M. (2014). Ruled-based, interlingual motivated mapping of plWordNet onto SUMO ontology. In N. Calzolari, K. Choukri, T. Declerck, H. Loftsson, B. Maegaard, J. Mariani, A. Moreno, J. Odijk, & S. Piperidis (Eds.), *Proc. Ninth International Conference on Language Resources and Evaluation (LREC’14)* (pp. 4351–4358). European Language Resources Association. Retrieved from <http://www.lrec-conf.org/proceedings/lrec2014/index.html>
- Klimek, B., & Brümmer, M. (2015). Enhancing lexicography with semantic language databases. *Kernerman DICTIONARY News*, 23, 5–10. Retrieved from [http://svn.aksw.org/papers/2015/KDICT\\_Lexicography/public.pdf](http://svn.aksw.org/papers/2015/KDICT_Lexicography/public.pdf)
- Maziarz, M., Piasecki, M., Rudnicka, E., Szpakowicz, S., & Kędzia, P. (2016). plwordnet 3.0 — a comprehensive lexical-semantic resource. In N. Calzolari, Y. Matsumoto, & R. Prasad (Eds.), *COLING 2016: 26th International Conference on Computational Linguistics: Proceedings of the Conference: Technical Papers: December 11–16, 2016, Osaka, Japan* (pp. 2259–2268). Retrieved from <http://aclweb.org/anthology/C/C16/>
- McCrae, J., Montiel-Ponsoda, E., & Cimiano, P. (2012). Integrating WordNet and Wiktionary with lemon (pp. 25–34). Berlin: Springer. [https://doi.org/10.1007/978-3-642-28249-2\\_3](https://doi.org/10.1007/978-3-642-28249-2_3)
- Měchura, M. (2012). Léacslann: a platform for building dictionary writing systems. In R. V. Fjeld & J. M. Torjusen (Eds.), *Proceedings of the 15th EURALEX International Congress* (pp. 855–861). Oslo: Department of Linguistics and Scandinavian Studies, University of Oslo.
- Měchura, M. (2016). Data structures in lexicography: from trees to graphs. In A. Rambousek, A. Horák,

- & P. Rychlý (Eds.), *Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2016* (pp 97–104). Brno: Tribun EU. Retrieved from <http://nlp.fi.muni.cz/raslan/2016/paper04-Mechura.pdf>
- Pease, A. (2011). *Ontology: A practical guide*. Angwin, CA: Articulate Software Press.
- Pease, A., & Fellbaum, C. (2010). Formal ontology as interlingua: the SUMO and WordNet linking project and Global WordNet. In C.-R. Huang, N. Calzolari, A. Gangemi, A. Oltramari, & L. Prévot (Eds.), *Ontology and the lexicon: A natural language processing perspective, studies in natural language processing*. Cambridge: Cambridge University Press. <https://doi.org/10.1017/CB09780511676536.003>
- Pedersen, B. S., Nimb, S., Asmussen, J., Sørensen, N. H., Trap-Jensen, L., & Lorentzen, H. (2009). DanNet: The challenge of compiling a wordnet for Danish by reusing a monolingual dictionary. *Language Resources and Evaluation*, 43. <https://doi.org/10.1007/s10579-009-9092-1>
- Pęzik, P. (2014). Graph-based analysis of collocational profiles. In V. Jesensek & P. Grzybek (Eds.), *Phraseologie im Wörterbuch und Korpus (Phraseology in Dictionaries and Corpora)* (pp. 227–243). Maribor: Filozofska fakulteta.
- Piasecki, M., Czachor, G., & Kędzia, P. (2018). Wordnet-based evaluation of large distributional models for polish. In F. Bond, C. Fellbaum, & P. Vossen (Eds.), *Proceedings of the 9th Global Wordnet Conference (GWC 2018)*. Global WordNet Association.
- Przepiórkowski, A., Hajnicz, E., Patejuk, A., Woliński, M., Skwarski, F., & Świdziński, M. (2014). Walenty: Towards a comprehensive valence dictionary of Polish. In N. Calzolari, K. Choukri, T. Declerck, H. Loftsson, B. Maegaard, J. Mariani, A. Moreno, J. Odijk, & S. Piperidis (Eds.), *LREC 2014, Ninth International Conference on Language Resources and Evaluation* (pp. 2785–2792). Retrieved from <http://www.lrec-conf.org/proceedings/lrec2014/index.html>
- Richardson, C. (2018). *What are microservices?*. <https://microservices.io/>
- Rudnicka, E., Bond, F., Grabowski, L., Piasecki, M., & Piotrowski, T. (2018). Lexical perspective on wordnet to wordnet mapping. In F. Bond, T. Kuribayashi, C. Fellbaum, & P. Vossen (Eds.), *Proceedings of the 9th Global Wordnet Conference, Singapore, 8–12 January 2018* (pp. 210–219). Singapore: Global WordNet Association. Retrieved from <http://compling.hss.ntu.edu.sg/events/2018-gwc/pdfs/gwc-2018-proceedings.pdf>.
- Saloni, Z., Woliński, M., Wołosz, R., Gruszczynski, W., & Skowrońska, D. (2012). *Słownik gramatyczny języka polskiego* [A grammatical dictionary of Polish]. Warszawa: Warsaw University.
- Videla, A., & Williams, J. (2012). *RabbitMQ in action: Distributed messaging for everyone*. Shelter Island, NY: Manning.
- Vinoski, S. (2006). Advanced message queuing protocol. *IEEE Internet Computing*, 10(6). <https://doi.org/10.1109/MIC.2006.116>
- Vossen, P. T. J. M., Soria, C., & Monachini, M. (2013). Wordnet-LMF: A standard representation for multilingual wordnets. In G. Francopoulo, & P. Paroubek (Eds.), *LMF Lexical Markup Framework*. Hermes/Lavoisier. <https://doi.org/10.1002/9781118712696.ch4>
- Wolff, E. (2016). *Microservices: Flexible Software Architectures*. Boston, MA: Addison-Wesley Professional.
- Zaśko-Zielińska, M., & Piasecki, M. (2018). Towards emotive annotation in plWordNet 4.0. In F. Bond, T. Kuribayashi, C. Fellbaum, & P. Vossen (Eds.), *Proceedings of the 9th Global Wordnet Conference, Singapore, 8–12 January 2018* (pp. 154–163). Singapore: Global WordNet Association. Retrieved from <http://compling.hss.ntu.edu.sg/events/2018-gwc/pdfs/gwc-2018-proceedings.pdf>

---

## Acknowledgment

This work was financed as a part of the investment in the CLARIN-PL research infrastructure ([www.clarin-pl.eu](http://www.clarin-pl.eu)) funded by the Polish Ministry of Science and Higher Education.

The authors declare that they have no competing interests.

The authors' contribution was as follows: concept of the study: Tomasz Walkowiak, Maciej Piasecki, Francis Bond; data analyses: Tomasz Walkowiak, Maciej Piasecki, Ewa Rudnicka; prototype implementation: Tomasz Walkowiak; the writing: Tomasz Walkowiak, Maciej Piasecki, Ewa Rudnicka, Francis Bond.

This is an Open Access article distributed under the terms of the Creative Commons Attribution 3.0 PL License (<http://creativecommons.org/licenses/by/3.0/pl/>), which permits redistribution, commercial and non-commercial, provided that the article is properly cited.

