The Concept of Lexical Platform

Maciej Piasecki[1], Tomasz Walkowiak,[1]
Ewa Rudnicka,[1] Tomasz Naskręt,[1] Francis Bond[2]

[1]G4.19 Research Group, Wrocław University of Science and Technology, Wrocław, Poland
{maciej.piasecki,tomasz.walkowiak,
ewa.rudnicka,tomasz.naskret}@pwr.edu.pl

[2]Computational Linguistics Lab, Nanyang Technological University, Singapore
bond@ieee.org

Abstract. The paper presents an idea of Lexical Platform proposed as a means for a lightweight integration of various lexical resources into one complex (from the perspective of non-technical users). All LRs will be represented as software web components implementing a minimal set of predefined programming interfaces providing functionality for querying and generating simple common presentation format. A common data format for the resources will not be required. Users will be able to search, browse and navigate via resources on the basis of anchor elements of a limited set of types. Lexical resources linked to the platform via components will preserve their identity.

Keywords: lexical resources, wordnet, interoperability of lexical resources.

1 The need for the integration of lexical resources

Lexical resources (LRs) have recently become more numerous for many languages. They describe different aspects of their lexical systems. However, their impact on popular, commercial and even research applications is surprisingly limited. One reason for this is the fact that existing LRs often originate from a variety of research projects, are based on different models and are encoded in different formats. All these factors make combining them into one complex system a challenging task. From the point of view of text processing applications, there seems to be no other way than mapping all resources to one common model and a limited number of formats in order to be able to identify links between individual resources or even to augment them with appropriate links.

Non-technical users of LRs, interested in consulting and browsing them, also face challenges in accessing them. LRs are spread across the web. Even if different LRs can be found in some virtual catalogues like CLARIN VLO,[1] every individual LR usually has to be accessed separately via different dedicated browsing and searching systems. Still, for such uses, we need only limited knowledge about a LR: what kind of elements we can ask for and how to present the query and search results to the users.

[1] https://www.clarin.eu/content/virtual-language-observatory-vlo

Our goal is to present an idea of a lexical platform as a virtual place for aggregating different type of LRs as separate individual components in a way that they form an interconnected system, a complex LR, from the user's point of view. We assume that the descriptions 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 for all types of LRs, but wordnets are in the focus since they are usually very large resources, providing rich descriptions, but are not so easily accessible for many users.

2 Related works

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

2.1 Formats and standards

One common format for LRs could solve most of the problems. Several standards have been proposed but none of them gained overwhelming coverage. Implementations of RDF for wordnets were proposed, but used only for single wordnets, e.g. Princeton WordNet (PWN) ([8]) and DanNet ([16]). Several implementations of Lexical Markup Framework (LMF), a generic ISO standard, have been proposed for wordnets, e.g. KYOTO LMF ([22]), GermaNet LMF ([10]), UBY LMF ([9]). However, KYOTO LMF is concentrated only on the representation of synsets, and the other two also do not allow for full representation of all existing wordnets, e.g. many features of plWordNet related to lexical units ([12]).

Lemon (13]) has been proposed as an ontology-based representation for lexicons and machine-readable dictionaries and linking them to the Semantic Web and the Linked Data cloud. The Lemon-based representation is still very much focused on PWN and cannot represent many elements present in different wordnets, but its various applications have already shown 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 hamper existing applications.

In human-oriented lexicography lexicons are mostly encoded as a tree, a hierarchical data structure of parent-child relations ([15]). Many authors ([1], [11], [15]) propose to use graph representation.

2.2 Platforms

UBY LMF[2] platform ([7]) was built as a solution for integrating LRs on both structural and semantic level. 12 LRs[3] have been linked into a complex system.

[2] https://dkpro.github.io/dkpro-uby/
[3] https://www.ukp.tu-darmstadt.de/data/lexical-resources/uby/

However, all these LRs have been converted to one common implementation of LFM. There is only one type of anchoring elements that are word senses.

CILI that is *Collaborative Interlingual Index* is described as "a flat list of concepts" which is currently based on PWN 3.0 set of synsets ([23]). It is intended to serve as an intermediary between wordnets of different languages within the Open Multilingual WordNet (OMW) ([4]). Currently, there are plans to extend CILI with concepts lexicalised in languages other than 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 *superceded.* Candidates for new ILI concepts must be linked to a concept in its 'mother' wordnet by one of well-known relations (hypernymy, meronymy, antonymy) and indirectly linked via this concept to the already existing CILI concept. CILI is available on open licence.

OMW[4] is an open platform aggregating wordnets of different languages linked via PWN 3.0 ([5]). Its component wordnets share a common representation format, i.e. currently CILI LMF format. For many wordnets the conversion to CILI LMF is unidirectional, i.e. it is not possible to reconstruct the original structure of a wordnet due to the flattening of the relation structure and the impossibility to reconstruct it.

PANACEA[5] ([3]) is a FP7 project focused on building a system of language resources (enhanced with a handful of tools) for Machine Translation. Wide range of resource for several languages have been developed and integrated, but LMF standard was chosen as the data format for dictionaries[6].

LEAP (Lexical Engine and Platform)[7] is a commercial product, focused on multilingual dictionary data, semantically linked combined with asymmetrical translation memory. It offers a REST API for developers. *Léacslann* ([14]) 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.

Lexonomy[8] (a descendant of the Léacslann) is a tool aimed for writing and publishing dictionaries. An entry includes mainly: a lemma (a headword form), PoS, word sense defined by a simple textual description and sense usage examples. Each entry description can be a mixture of text and marked elements (inline XML markup) corresponding to different elements of the entry structure. The dictionary has a structure of graph ([15]).

[4] http://compling.hss.ntu.edu.sg/omw/
[5] http://www.panacea-lr.eu/
[6] http://www.panacea-lr.eu/system/deliverables/PANACEA_D3.4.pdf
[7] https://www.oxforddictionaries.com/our-story/leap
[8] http://www.lexonomy.eu/_info/

3 Basic Assumptions

The idea of the Lexical Platform (LexP) originated from a handful of observations and intuitions.

LexP should group together different LRs as independent components, implemented as software modules. Only a minimal set of requirements should be imposed on developers. Moreover, individual identity of all LRs must be visible and preserved inside the platform. Crucially, LexP is not supposed to become a 'super-resource', because that may cause reluctance of resource creators to join in.

LexP will promote the use of a limited set of common formats, but it will not enforce any specific data format on its components. A component may be located in any freely selected network location. It does not need to be copied to LexP altogether with the LR data. This can be an interesting option from the point of view of IPR issues and data protection. A component will be accessible via a set of Programming Interfaces (PIs). They can be implemented, e.g. as traditional Web Services (WSs). One PI can be implemented as a one separate WS, or several PIs can be provided by a single WS - this is a matter of a detailed design decisions for LexP. Still, some minimal set of PIs will be specified and required to be implemented by every component (including PIs that allow the user to obtain the description of a component, access an element of the resource or get the visual representation of a resource element). However, components can provide any number of additional PIs.

LexP would not allow for any changes in the content of individual resources, as well as in the links between them. LexP is meant only to be a tool for accessing a complex system of linked resources, not a system supporting development of LRs.

The access to the data from the component lexical resources will be constrained in order to make the LexP construction feasible. LexP components will encapsulate the data, i.e. the only access to the data will be possible only via the PIs of the components. Every component can provide data in any format, but some formats, e.g. Lemon (or its expansions), may be suggested as common ones. Construction of converters from native formats to a limited number of common ones will be promoted. Every component will be required to support addressing elements of LexP via anchoring elements of limited and predefined types that will be specified by an ontology. Still every LexP-complaint component can offer expanded methods of addressing LR elements.

Inter-linking of LRs via LexP components is a key issue. It will be based exclusively on the content of LRs. Each component recognises references to elements of the limited set of types. Such elements serve as selected points by which the data from different components are *anchored* to the whole platform and inter-linked between them. Such selected data elements will be called *anchor elements*. Anchor elements should naturally originate from the construction of a LR. They should be its characteristic elements which users browse it by or which users most usually search for. Anchor elements should be also those data elements that provide native (or natural, typical) mapping to other LR (or knowledge resources). The selection of anchor element types can be left to resource creators, but if an anchor element is to be used by other resources of the same type the way of naming it must be known to the creators of those resources. The following types of anchor elements are expected to be

provided by different components, (a provisional list to be worked out in detail during the design process): *word form* (or word, including multi-word expressions), *lemma* (or literals, canonical forms, entry form, basic form), *lexical unit* (word sense), *synset*, *frame* (syntactic and/or semantic), *domain* (context), and *concept*. Every component will provide on request (via a PI) a list of anchor element types that it can recognise. An ontology of anchor element types will be created (if possible based on an existing one) and maintained as the only central knowledge resource of LexP, but this ontology should be limited.

The primary functionality of LexP will be focused on non-technological users and will be close to the idea of Federated Content Search[9] of CLARIN: the ability to search from one single point across many corpora, but with a limited query language. Users will be able at least

- to learn about existing LRs and the range of information provided by them,
- to search across combined LRs on the basis of anchor elements supported by different components and to browse LRs by lists of anchor elements retrieved from the components,
- to manually browse across linked LRs on the basis of names of anchor elements,
- and finally, to find out how to obtain and download original resources and to learn how to browse different LRs in their native browsers.

In order to make manual browsing of different LRs from LexP, it is assumed that all components are required to provide a PI generating a specified *presentation format* for a specified anchor element. The generated presentation should highlight anchor elements and user's clicks on the anchor elements should be reported together with the anchor element's name to LexP in order to enable interactive browsing. The exact presentation format is a matter of design decisions, but it should be as simple and as popular as possible (in order to simplify the construction of components), e.g. HTML, XML; SVG, etc.

Users should be also able to list anchor elements described by a component and the whole LexP. For this purpose, we need PIs that allow listing together with some forms of filtering, e.g. *PoS, UPOS, language, supertype* (for ontologies)*, hypernym* (for lexico-semantic networks)*, semantic domains* etc. Lists retrieved from components will be collected by LexP and presented to the users as merged lists.

It would be hard to follow versions of individual LR elements, so LR versions will be reported by the description PI of a component.

In the case of technological users, the support provided by LexP will be naturally limited without the guaranteed mapping to a common format, but still some functions can be envisaged. LexP can be used for collecting data sub-structures describing specified anchor elements in a native format or some other format if a converter is available. PIs for calculating similarity measures between anchor elements can be envisaged. Other possible functions could provide, e.g., some statistics, clustering of elements, mapping texts onto substructures extracted from LexP components.

[9] https://www.clarin.eu/content/content-search

4 Platform architecture

4.1 Lexical microservices

LexP will link diversified LRS in a flexible and autonomous way, i.e. the resources linked will not be merged in one big `super-resource', but each resource will be preserved as a separate module and will keep its identity. Such a strategy should help to convince a large group of resource creators for linking their resources to the platform.

The existing API for LRs are developed in different languages (Java, C++, Python). Moreover, many of them (e.g. plWordNet [12], Walenty [18]) stores very large data sets. Therefore, the time of loading such a component is much longer than processing a single task. The solution is to run a LR component as a service with data loaded into memory. Each service is running its own process. The usage of services communicating with others by lightweight mechanisms solves also a problem of 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" [6]. Microservices [21] and a service-oriented architecture [3] have recently started gaining wide popularity. The microservice architecture will enable continuous development/deployment [19] of LexP.

Each LR will be represented inside the platform as a separate microservice. Access to any LR requires some time, therefore it is worth to run several instances of microservices for a given LR component. A queuing system will be used to distribute requests among microservices. Each LR component will be assigned its own queue. LexP microservice will collect tasks from a given queue and send back messages when results are available. Such a solution will facilitate effective scalability capabilities since a queuing system acts as a load balancer.

Every LexP component implemented as a microservice will provide a set of required PIs. A minimal set of required (obligatory) PIs will encompass functions:

- *description* - delivers meta-data for the resource and the component (including license information) and information about the PIs provided by the component, facilitates component registration in LexP,
- *getElement* - returns all descriptions related to a specified resource anchor element in their native format, e.g. description of all synsets for a given lemma in a wordnet,
- *getHtml* - generates a simple visualisation of a specified resource anchor element in HTML that can be easily rendered in web browser without the need of interpretation of the data,
- *list* - returns a list of anchor elements, several ways of filtering are envisaged.

The list could be extended for the specific needs of a LR, e.g. functions such as the following (depending on the licence of the resource):

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

4.2 System architecture

The planned LexP architecture is presented in Fig. 2. We propose to use the AMQP[10] protocol for lightweight communication with lexical microservices and the open source RabbitMQ[11] 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, the additional server grants the access from Internet. It works as a proxy for the core system delivering synchronous HTTP-based REST API. Such an approach allows for easy integration with almost any kind of applications including JavaScript. For applications oriented on asynchronous processing the AMQP based access will be granted.

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

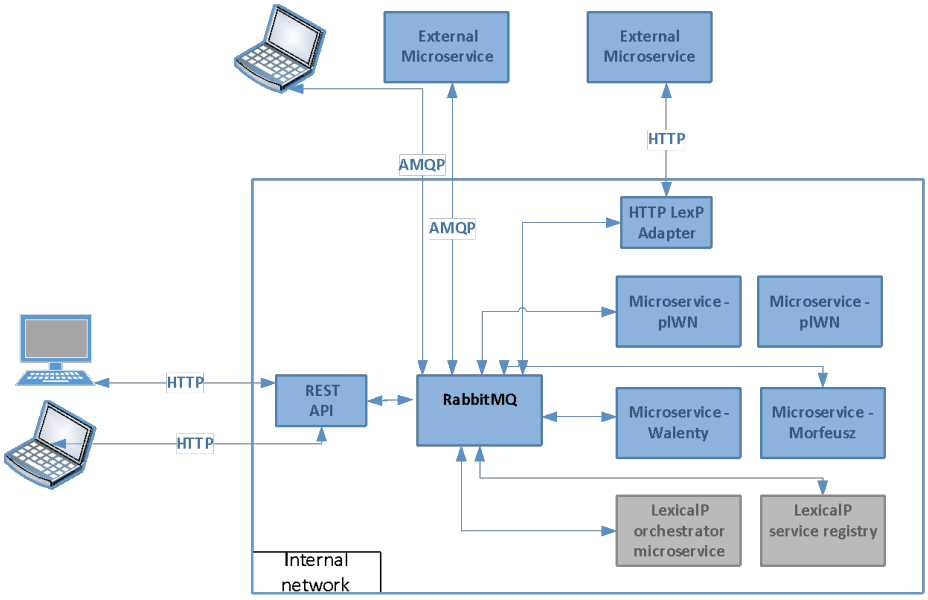

Fig. 1. Lexical Platform architecture

In addition, a LexP 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 microservice. The orchestrator is aimed to:

- filter all wrong requests,
- add mapping between an external resource name and an internal microservice name,
- send simple requests to a given type of a microservice (lexical resource),
- process complicated tasks (built on a sequence of calls to the lexical

[10] https://www.amqp.org
[11] https://www.rabbitmq.com

resources, such as):
    - 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,
    - providing access to the whole resource in a given resource specific format,
    - logging of external tasks and users' data (IP, user names) for 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 microservices can be deployed on the central server of LexP or on servers of their suppliers (or authors, owners, etc.). If the external microservice is not able to follow AMQP protocol, a specific, resource oriented adapter (see. Fig. 2) can be developed to connect any external resource to the platform. A resource adapter can include a cache capabilities to speed up the resource access.

The proposed architecture includes also a *service registry*. It plays a role of a simple database of microservices names. Microservice instances on the startup registers itself in the given queue and deregistered on shutdown. Moreover, the RabbitMQ broker may invoke a microservice health check to verify that an instance is able to handle requests (if not the instance is removed from a list of the queue consumers). The service registry monitors the number of clients of each queue and provides a list of working components.

The RabbitMQ is able to work in the distributed way. Several instances of RabbitMQ could cooperate in different manners[12] (with clustering, with federation, and using the shovel). Therefore, it will be easy to distribute LexP among different data centres.

[12] https://www.rabbitmq.com/distributed.html

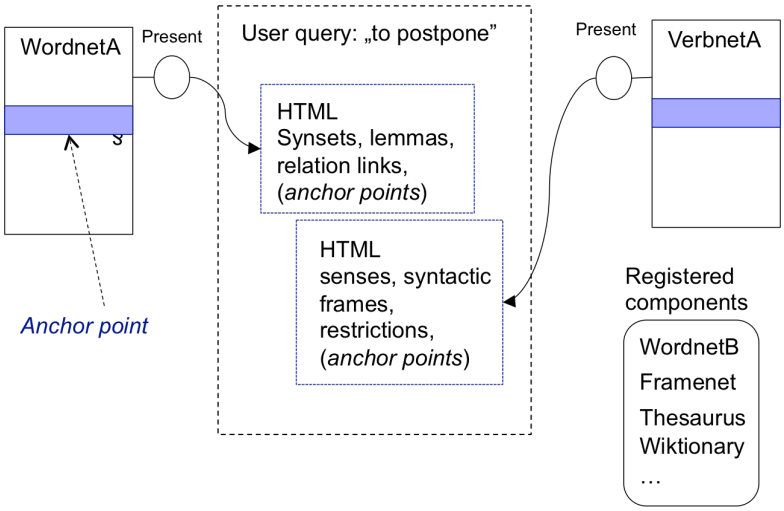

Fig. 2 An illustration of the working of Lexical Platform, within which modules making available specific resources provide a presentation widget (here for simplicity called HTML) for an element chosen by the user

4.3 Central web application

The architecture described in the previous chapter is oriented toward access to LRs for programs. To allow humans access to the platform a central web application will be developed. It will communicate with the platform core by a HTTP, JSON based REST API. The user will be able to access any functionality provided by the platform API. The results will be displayed on the screen by interactive widgets. Each of the LRs will have a specific JavaScript widget that will graphically present the requested resource. For example, in case of a wordnet, it can be an interactive graph that shows the synsets and all related synsets. In case a specific resource widget is not available (or not yet developed), the generic one will be called. It will use the basic HTML result from a lexical microservice (the result of the getHTML function of the lexical microservice).

Moreover, the user will be able to easily search the LexP resources. Thus, LexP expands in some sense the idea of Federated Content Search to the federated lexical resources search. Technical users, despite the easier download of the whole resources in accordance with their licences, will be able to download selected combinations of resources or their parts as a graph.

5 Illustration: Integration of Polish Lexical Resources

Integration of a set of comprehensive but heterogeneous and bilingual LRs for Polish can be a first case study for LexP idea. The set includes:

- *plWordNet* 3.0 emo (*Słowosieć*)[13] ([12]) - a very large wordnet for Polish,

[13] http://plwordnet.pwr.edu.pl

partially manually annotated with sentiment and basic emotions ([24]) emotive annotation - anchor elements: lemma (170k), lexical unit (245k), synset (184k),

- *enWordNet* 1.0 - a significantly expanded Princeton WordNet 3.1 ([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*[14] ([18]) - a Polish valence dictionary, describing both syntactic and semantic argument structures, frames are defined for lemmas and lexical units that correspond to large extent to lexical units from plWordNet - anchor elements: lemma (15k), lexical unit, frame,
- *Polimorf*[15] - a Polish morphological dictionary and SGJP[16] - a grammatical dictionary of Polish - anchor elements: word form (~4M), lemma,
- *NELexicon* 2.0[17] - a lexicon of Polish Proper Names described with semantic categories, - anchor elements: lemma (~2.4M), synset (representing semantic classes),
- *MWELexicon*[18] - a lexicon of Polish Multi-word Expressions described by their lexicon-syntactic structures, all MWEs are described as lemmas in plWordNet 3.0 emo - anchor elements: word form, lemma (54k),
- *Hask*[19] ([17]) - a set of collocational databases - anchor elements: word form, lemma (150k for English).

All the LRs mentioned above form a huge system, but they are not now connected for browsing, and a user needs to consult several different specialised browsers to learn how much information he can obtain from the system. However, there are web applications for all these LRs, they already offer some presentation formats for the web, so construction of LexP components for them should not be a very difficult exercise.

A minimal set of anchor element types provides links between all these resources. So having the exercise done, a user can come with a word form, e.g. from a text, next he can learn potential lemmas (highlighted in the presentation format) from the morphological components, navigate through lemma links to plWordNet component, check the possible valence frames in Walenty component, collocations from Hask, learn about potential translations of the lexical meanings through mapping presented by plWordNet component etc.

[14] http://zil.ipipan.waw.pl/Walenty
[15] http://zil.ipipan.waw.pl/PoliMorf
[16] http://sgjp.pl/leksemy/
[17] http://hdl.handle.net/11321/247
[18] http://hdl.handle.net/11321/274
[19] http://pelcra.pl/hask_en/

6 Further Works

We have started implementing the first version of LexP for the use cases for Polish LRs described shortly above. The idea of LexP has been thoroughly discussed inside CLARIN and several groups declared support for it. This can be the first step for collaborative development of LexP.

LexP should allow for better promotion and accessibility of existing LRs. Non-technical users will be able to discover and browse LRs from a single access point. Technical user will see the content of many LRs in a single place. However, LexP will not be a new `super-resource' which fully integrates the information in the various resources, at the cost of obscuring the individual sources. Instead, all LRs linked to LexP will preserve their identity. LRs can be kept in the original sites, which also allows for the possibility to provide access protected by authentication to those LRs that require restricted access.

LexP will be open for relatively easy linking of new resources to it. One of the biggest problems to be solved is proper treatment of information about versions of different resources, e.g. lexical units from Walenty correspond to a certain version of plWordNet, while LexP in its simplest form will present the latest version of plWordNet. There are several potential solutions, but a final one must preserve simplicity of the LexP idea. The platform can be also a good tool for promoting the need for developing one common format for LRs and converging descriptions of their models.

Acknowledgments

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

References

1. **Aguado-de-Cea**, G., Montiel-Ponsoda, E., Kernerman, I., Ordan, N. (2016). From dictionaries to cross-lingual lexical resources. In: Kernerman Dictionary News, 24, pp. 25-31.
2. **Bel**, N., Poch, M., Toral, A. (2012). **PANACEA (Platform for Automatic, Normalised Annotation and Cost-Effective Acquisition of Language Resources for Human Language Technologies)**. In Proceedings of the 16th Annual Conference of the European Association for Machine Translation: EAMT 2012: Trento Italy, May 28th - 30th 2012. pg. 90.
3. **Bell**, M.: SOA Modeling Patterns for Service-Oriented Discovery and Analysis. Wiley & Sons (2010).
4. **Bond**, F., Paik, K. (2012). A survey of wordnets and their licenses. In *Proceedings of the 6th Global WordNet Conference (GWC 2012)*. Matsue. 64–71.
5. **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. 1352–1362. http://aclweb.org/anthology/P/P13/P13-1133.pdf
6. **Dragoni**, N., Giallorenzo , S., Lluch-Lafuente, A., Mazzara M., Montesi F., Mustafin R., Safina L.: Microservices: yesterday, today, and tomorrow, CoRR, vol abs/1606.04036 (2016).

7. **Eckle-Kohler**, Gurevych, I., Hartmann, S., Matuschek, M., Meyer, Ch.M. (2013). UBY-LMF - exploring the boundaries of language-independent lexicon models, in Francopoulo, G. (ed.) LMF Lexical Markup Framework, ISTE / Wiley (2013).
8. **Fellbaum**, Ch. (ed) (1998). WordNet: An electronic lexical database, MIT Press, Cambridge.
9. **Gurevych**, I., Eckle-Kohler, J., Hartmann, S., Matuschek, M., Meyer, Ch.M., Wirth, Ch. (2012). UBY - A Large-Scale Unified Lexical-Semantic Resource Based on LMF. *EACL 2012*.
10. **Heinrich**, V., Hinrichs, E. (2010). Standardizing Wordnets in the ISO Standard LMF: Wordnet-LMF for GermaNet. *Proceedings of COLING 2010*.
11. **Klimek**, B., Brummer, M.. (2015). Enhancing lexicography with semantic language databases. In: Kernerman Dictionary News, 23, pp. 5-10.
12. **Maziarz**, M., Piasecki, M., Rudnicka, E., Szpakowicz, S., Kędzia, P.: (2016) plWordNet 3.0 - a Comprehensive Lexical-Semantic Resource. In: Calzolari, N., Matsumoto, Y., Prasad, R. (eds.), COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, pp. 2259-2268, Osaka (2016).
13. **McCrae**, J., Cimiano, J. P., Montiel-Ponsoda, (2012). Integrating WordNet and Wiktionary with lemon. In *Linked Data in Linguistics*. eds. Chiarcos, Ch., Nordhoff, S., Hellman, S. Springer.
14. **Mechura**, M. (2012). Léacslann: A platform for building dictionary writing systems, Proceedings of the 15th Euralex International Congress. Oslo: 855-861.
15. **Mechura**, M. (2016). Data Structures in Lexicography: from Trees to Graphs. RASLAN 2016 Recent Advances in Slavonic Natural Language Processing, 97.
16. **Pedersen**, B. S., Nimb, S., Asmussen, J., Sørensen, N. H., Trap-Jensen, L. and Lorentzen, H. (2009). DanNet -- the challenge of compiling a WordNet for Danish by reusing a monolingual dictionary *Language Resources and Evaluation.* Volume 43:3 pp. 269-299.
17. **Pęzik**, Piotr. Graph-Based Analysis of Collocational Profiles. In Phraseologie Im Wörterbuch Und Korpus (Phraseology in Dictionaries and Corpora), edited by Vida Jesenšek and Peter Grzybek, 227–43. ZORA 97. Maribor, Bielsko‑Biała, Budapest, Kansas, Praha: Filozofska fakuteta, 2014.
18. **Przepiórkowski**, A. Hajnicz, E., Patejuk, A. Woliński, M., Skwarski, F. Świdziński, M. Walenty: Towards a comprehensive valence dictionary of Polish. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014*, pages 2785–2792, Reykjavík, Iceland, 2014. ELRA.
19. **Richardson**, Ch. (2016). What are microservices? http://microservices.io/
20. **Rudnicka**, E., Witkowski, W., Kaliński M. (2015). "Towards the Extension of Princeton WordNet". *Cognitive Studies 15*, 335-351. Retrieved from: https://ispan.waw.pl/journals/index.php/cs-ec/article/download/cs.2015.023/1774.
21. **Wolff**, E. (2016). Microservices: Flexible Software Architectures, Addison-Wesley.
22. **Vossen**, P., Soria, C. , Monachini, M. (2013). Wordnet-LMF: A Standard Representation for Multilingual Wordnets. In *{LMF} - Lexical Markup Framework.* ed.. Francopoulo, G. ISTE Ltd + John Wiley \& sons, Inc.
23. **Vossen** P., Bond, F., McCrae, J., Fellbaum, Ch. (2016). CILI: the Collaborative Interlingual Index. In *Eighth meeting of the Global WordNet Conference (GWC 2016)*, Bucharest.
24. **Zaśko-Zielińska**, M., Piasecki, M., Szpakowicz, S. (2015). *A Large Wordnet-based Sentiment Lexicon for Polish*. In *Proceedings of RANLP 2015*. pp.721-730.