Corpus-Based Multilingual Event-type Ontology: Annotation Tools and Principles

Eva Fučíková, Jan Hajič, and Zdeňka Urešová
Institute of Formal and Applied Linguistics, Faculty of Mathematics and Physics
Charles University, Prague, Czech Republic
{fucikova,hajic,uresova}@ufal.mff.cuni.cz

Abstract

In the course of building a multilingual Event-type Ontology resource called SynSemClass, it was necessary to provide the maintainers and the annotators with a set of tools to facilitate their job, achieve data format consistency, and in general obtain high-quality data. We have adapted a previously existing tool (Urešová et al., 2018b), developed to assist the work in capturing bilingual synonymy. This tool needed to be both substantially expanded with some new features and fundamentally changed in the context of developing the resource for more languages, which necessarily is to be done in parallel. We are thus presenting here the tool, the new data structure design which had to change at the same time, and the associated workflow.

1 Introduction

This paper describes the tools and the associated annotation process used for building up a corpus-based multilingual event-type ontology, called SynSemClass (Urešová et al., 2022). Since the overall premise is based on working from data ("bottom-up", see esp. Urešová et al. (2018a)), the work starts from a parallel corpus (at least between English and the given language being processed/annotated). Similarly, the ontology classes are also built from the language side: there is no predefined ontology. The words included in the classes are translational counterparts and as such can be considered synonyms (and we will refer to them in such a way with all the caveats connected with such simplification). The language of original texts is English, the verb synonyms captured in the ontology are, at the moment, only English, Czech, German, and Spanish.

In order to allow for independent annotation of different languages, it was necessary to develop guidelines for an annotation procedure applicable to many languages, design a workflow of the annotation for new languages, and create a configurable annotation tool for adding such new languages. This involved, among other things, designing a general configuration, including e.g., URLs for external linking of language resources, a stand-off annotation scheme (each language needs to be annotated separately by, presumably, teams scattered all over the world), and an editor capable of working with just the relevant language(s). Here, we describe the principles of restructuring and reformatting the dataset to accommodate multilinguality as well as the description of the capabilities of the extended new tool.

2 Related Work

General editors over databases used for editing lexical resources are not suitable due to the amount of customization and overhead needed for the complex structure of SynSemClass ontology, as argued already by Urešová et al. (2018b).

Specific tools for building lexicons have been built and/or used since at least the 1980s, as described e.g., in (Teubert, 2007). *Lexicon Creator* is suitable for working with pre-extracted wordlists. *Lexicon Builder* is a web service (Parai et al., 2010) for compiling custom lexicons from BioPortal ontologies. *CoBaLT Editor* (Kenter et al., 2012) has been used for historical texts and lexica. *Dicet* (Gader et al., 2012) is aimed at lexical graphs (this one is closest to the needs of SynSemClass annotation).

A broad overview and a brief description of some available editors and environments that can be used for the building of ontologies is provided for example by Alatrish (2012). Others are, e.g., Apollo[1], OntoStudio[2], Protégé[3], Swoop[4] and Top-

[1] http://apollo.open.ac.uk/index.html
[2] https://www.semafora-systems.com/
[3] https://protege.stanford.edu/
[4] https://www.softpedia.com/get/Internet/Other-Internet-Related/MIND-lab-SWOOP.shtml

Proceedings of the 21st International Workshop on Treebanks and Linguistic Theories (TLT, GURT/SyntaxFest 2023), pages 1 - 10

Braid Composer Free Edition[5].

Also relevant for our work are the general Linguistic Linked Open Data (LLOD) editors, but the SynSemClass data are still to be (re)defined as LLOD.[6]

SynSemClass is linked to a number of existing resources having their own specific editors so we tested also the suitability of their editors for our purposes but we found them not readily adaptable to the SynSemClass annotation scheme, since it requires more tasks to be covered than e.g., FrameNet editor (Fillmore, 2002) or Propbank frameset editor (Choi et al., 2010) can provide.[7]

3 Starting Point

As described in (Urešová et al., 2018a) and especially in (Urešová et al., 2018b), the previous version of the SynSemClass ontology was aimed at building the core, bilingual event-type ontology. It has been done in a specific situation - when advanced, manually annotated resources existed for both Czech and English, which had (indeed) been used to get an efficient workflow and accurate, richly annotated resource. The complexity of the definition of the then-called CzEngClass resource - with its syntactic-semantic mappings of valency slots to the newly developed semantic roles (associated with every class), linking to 9 external resources, and examples from a parallel corpus - has led to the development of the SynEd annotation tool with its functions tailored to the resources at hand. While the the semantic roles resemble FrameNet (Baker et al., 1998a) "Frame Elements", and sometimes borrow their names from there, it should be pointed out that there is one fundamental difference: the semantic roles used in SynSemClass aim at being defined across the ontology and not per class (as they would be if we follow the "per frame" approach used in FrameNet). In addition, the existence of the parallel treebank (the Prague Czech-English Dependency Treebank, (Hajič et al., 2012)) with its rich annotation scheme, exactly matching the task at hand in that it contained the necessary sense distinctions as recorded in the valency frames of the Czech and English valency lexicons, was taken advantage of in the design. The associated workflow was then very efficient, including complete double annotation and adjudication to arrive at high-quality resource.

The resources used come from the following datasets:

- Prague Czech-English Dependency Treebank (PCEDT) (Hajič et al., 2012),
- PDT-Vallex (Urešová et al., 2021),
- CzEngVallex lexicon (Urešová et al., 2015),
- EngVallex lexicon (Cinková et al., 2014),
- VALLEX lexicon (Czech) (Lopatková et al., 2020),
- FrameNet (Baker et al., 1998b; Fontenelle, 2003)[8],
- VerbNet (Schuler, 2006)[9],
- PropBank (Palmer et al., 2005)[10],
- OntoNotes Groups (Pradhan and Xue, 2009)[11], and
- WordNet 3.1(Fellbaum, 1998)[12].

The result of the previous efforts to create CzEngClass and subsequently extend it as SynSemClass is publicly available as a dataset[13] and for browsing as a web interface and service.[14] This latest version contains 883 classes; 63 of them already contain German verbs (while still being added using the original workflow and editor). Of the original 67,401 class member candidates, approx. 8,000 class members remained in this SynSemClass version, i.e., approx. 3,595 English, 464 German, and 4,110 Czech class members.[15] Adding German (and then Spanish) brought many new issues that needed to be addressed, and eventually it led to the development of the new data structure, editor and workflow that we are presenting in this paper.

We summarize briefly the design of the lexicon (Fig. 1) and the main points of its composition and structure.[16] Each class in SynSemClass is assigned a common set of semantic roles, called a "roleset",

[5] https://franz.com/agraph/tbc/

[6] Under the HumanE AI Net Micro-Project called Multilingual LLOD for the Semantic Web, still under construction.

[7] VerbNet uses the XML structure supplied in the associated DTD file.

[8] https://framenet.icsi.berkeley.edu

[9] https://verbs.colorado.edu/verbnet

[10] https://propbank.github.io/v3.4.0/frames/index.html

[11] https://doi.org/10.35111/xmhb-2b84

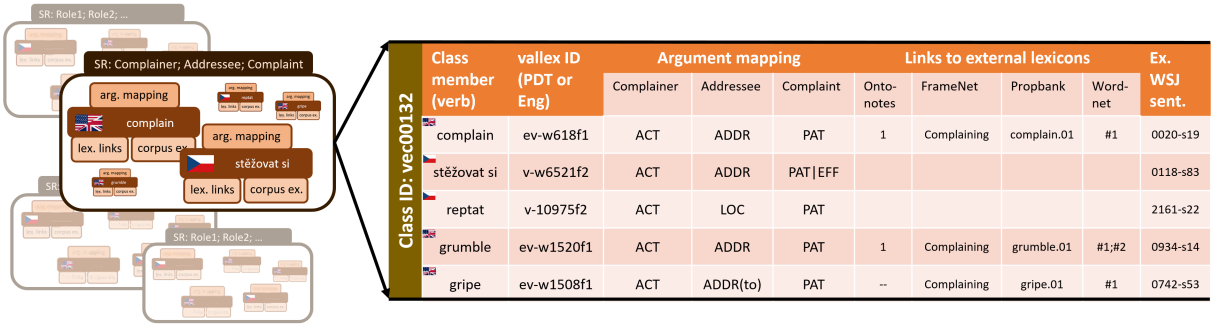
[12] https://wordnet.princeton.edu

[13] https://hdl.handle.net/11234/1-4746

[14] https://lindat.cz/services/SynSemClass.

[15] Currently, there are approx. 70 Spanish classes annotated with about 5,200 class members.

[16] Described in detail in (Urešová et al., 2020).



Class member (verb)	vallex ID (PDT or Eng)	Argument mapping			Links to external lexicons				Ex. WSJ sent.
		Complainer	Addressee	Complaint	Onto-notes	FrameNet	Propbank	Word-net	
complain	ev-w618f1	ACT	ADDR	PAT	1	Complaining	complain.01	#1	0020-s19
stěžovat si	v-w6521f2	ACT	ADDR	PAT|EFF					0118-s83
reptat	v-10975f2	ACT	LOC	PAT					2161-s22
grumble	ev-w1520f1	ACT	ADDR	PAT	1	Complaining	grumble.01	#1;#2	0934-s14
gripe	ev-w1508f1	ACT	ADDR(to)	PAT	--	Complaining	gripe.01	#1	0742-s53

Figure 1: Example entry in SynSemClass ("complain-stěžovat si")

indicating the prototypical meaning of the given class. A roleset contains the core "situational participants" labelled as "semantic roles" common for all the multilingual class members (the individual multilingual verb senses) in one class. Each class in SynSemClass is viewed as a substitute for an ontology unit, similar to the treatment of WordNet synsets. Class members are (for the time being) verbs (in different languages). It is essential that these verbs are sense-distinguished; more precisely, each "class member" is meant to be a verb sense. These senses must be predefined (or defined on-the-fly and assigned a particular sense ID). For Czech end English, they have been taken from the existing valency lexicons PDT-Vallex and EngVallex, where the individual valency frames are already sense-disambiguated (and IDs assigned to them).

Class members are linked to both internal and external resources. The Czech class members are linked to the following Czech valency lexicons: PDT-Vallex, CzEngVallex (both internal, linked by means of their ID's), and VALLEX (external).[17] The English class members are linked to the internal lexicons EngVallex and CzEngVallex and to external sources, i.e., FrameNet, VerbNet, PropBank, OntoNotes Groups and WordNet.

4 Towards Multilingual Lexicon Design

The work on adding German and now Spanish (Fernández-Alcaina et al., 2023) made it clear that a refactorization of both the data structure of SynSemClass and corresponding changes in the SynEd and the associated workflow are necessary in order to be able to concurrently work on more languages. We are testing the new, more "universal" approach on Spanish. In the future, we would like to engage external teams (anyone who wants to contribute their language). In such a case, the design, datasets, suggested workflow and tools provided must be easy to use and understandable. However, this has not changed the original idea nor the overall design of SynSemClass - it is still an event-type ontology with classes as the main units representing event-type concepts, associated with a fixed set of semantic roles, and the class members are word senses representing the expression of that event-type concept in a particular language. This schema will not change even if other parts of speech (nouns, adjectives, etc.) are added.

But still, some assumptions providing a basis for the creation of the original structure and tools have to be scaled down. For example, a parallel (deeply and richly annotated) corpus exists only for a handful of language pairs and only some of them have associated valency (or predicate-argument, or word-sense-disambiguated, or similar) lexicons linked to the corpus. Word (or even just verb) senses are sometimes available (e.g., in multilingual WordNets), but generally not in referred to as such from an annotated corpus. Each language has a different set of available semantic lexicons to which the class members in that particular language can be linked.

4.1 The Overall Design

The main reason for the new design of the SynSemClass resource as a whole comes from the following basic requirements:

- work on different languages will be carried out in parallel by different teams, without the need for continuous access to the main repository,
- versions of the lexicon will integrate various versions of the language-dependent parts,
- common data (such as the set of semantic roles) cannot be amended independently.

[17] https://ufal.mff.cuni.cz/vallex/4.0

In addition, there are other constraints, like the size of the language-dependent part and the whole resource, time needed to copy the whole resource or its parts over the internet when editing and committing changes, etc.

A natural question arises regarding a comparison to the massively parallel and massively multilingual effort of building the Universal Dependencies treebanks (Nivre et al., 2016), which looks similar, and which uses a simple GitHub repository[18] that contains everything from the documentation and guidelines to the validation scripts. The SynSemClass annotation is similar but differs in one important point: while in the UD case, the only thing that is shared across the languages is the CoNLL-U format and the sets of base (or core) labels used for annotating POS, morphological features, and dependency relations that allow only for some language-specific flexibility, in the SynSemClass case, the common set of classes (event-type concepts) of the ontology, to which all the language-dependent data point to, will certainly undergo much more frequent changes than the shared UD "tagsets" did.[19] This factor has to be reflected in the data structure design, the workflow, and the editor as well.

Based on these requirements, the data structures and the editor have been designed as follows:

- the structure of the resource is implemented in a stand-off mode, i.e., the common part will be shared by the language-dependent parts (i.e., by the dataset containing words expressing the classes (event-type concepts) in the particular language),
- the editor remains a desktop application working on locally available data (possibly versioned in github or svn or similar system),
- the central repository will be a GitHub repository, with a "read-only" part containing the common data (i.e., the set of current classes with definitions, set of semantic roles and their distribution across classes as the main contents),
- the minimal requirement for existing resources to work on a new language will be the existence of a parallel corpus[20] between a language already covered by SynSemClass (preferably English that will, presumably, always have the highest coverage) and the language being added.

The design is such that there are no redundancies - all the common data are in the centrally maintained dataset (a single file) while all the language-dependent data are separate, making the parallel work possible and independent of the changes made in other languages.

Of course, the very existence of the dynamically changing common dataset is a complication that cannot be circumvented by technical means. However, it is unavoidable in all such multilingual projects - as known from, e.g., medicine (MESH databases, the ICD classification of diseases, etc.). It implies continued commitment on the maintainers side and also some commitment on the side of the authors of the individual language datasets, even if many amendments caused by changes in the central common datasets are either "non-breaking", such as adding a class and related semantic roles, or can be done automatically, e.g., renaming a role.

The workflow is then as follows:

1. based on the required parallel corpus, candidates are determined for each class in the current SynSemClass version,
2. an initial stand-off style language-dependent file is created with the correct format, annotators allowed to edit, etc., properly linking to the central common file classes and semantic roles,
3. annotators, following the guidelines for editing individual classes, work on pruning the language-dependent file from wrongly suggested class member candidates, assign roles mapped to syntactic arguments, add links to external resources, and select examples, using the SynEd editor and described in Sect. 4.3),
4. the annotators suggest changes by creating GitHub issues, or emailing the central maintainers to change or add classes and/or roles, edit role definitions, etc.; the maintainers will have to decide which changes to implement to the common dataset that will not break the

[18] https://universaldependencies.org/

[19] They have only changed between version 1 and 2, and were extended in a central way for the "Enhanced dependencies" available now for several treebanks.

[20] This could be, in the future, further relieved to assume just an existence of a monolingual corpus, depending on the progress in the way initial assignment to classes can be done, e.g., by multilingual embeddings, the results of current experiments with multilingual BERT(s), transfer learning, etc.

other language-dependent datasets, or batch-edit them to validate against the common (amended) dataset,

5. after adjusting for these changes, the language-dependent file will be committed and validated, iteratively and in cooperation with the annotators, until an error-free version can be declared publishable.

In this paper, we further elaborate and demonstrate the editor (point 3 of the above workflow) in Sect. 4.3. However, before presenting the main features of the editor, we describe also the new structure of the datasets in more detail in Sect. 4.2 below.

4.2 Structure of the Datasets

The datasets are the files the new editor works with, in a configurable way. We distinguish:

1. the common dataset and
2. the language-dependent dataset.

4.2.1 Common Dataset

The common dataset is a single file with the following structure:

```
<synsemclass_main owner="EF">
```

... (header with main users and roles [only])

```
<body>
<veclass id="vec00001">
 <commonroles>
  <role idref="vecroleAgent" />
  <role idref="vecroleComponents" />
  <role idref="vecroleCreated_Entity" />
  <role idref="vecroleAssets_currency" />
 </commonroles>
 <classnote/>
 <local_history><local_event
                time_stamp="..." .../>
                ...</local_history>
</veclass>
<veclass ...>
...
</veclass>
```

... (more classes of synonyms, using the veclass element)

```
</body> </synsemclass_main>
```

As seen from the above extract from the common (main) file, it only contains the definitions of semantic roles (which are common to the whole SynSemClass ontology) and a list of classes, with only the list of roles assigned to that particular class. For each class, this list of roles is fixed and common for all languages that are part of the SynSemClass ontology but which are contained in separate files, one file per language (see Sect. 4.2.2).

4.2.2 The Language-dependent Dataset

The language-dependent dataset has the following structure (German examples shown, simplified):

```
<synsemclass_DE>
 <header>
```

... (The first part of header with edition, version and description info)

```
  <list_of_users>
   <user id="2" annotator="yes" name=.../>
   <user .../>
  </list_of_users>
  <reflexicons>
   <lexicon id="\ssclass{}" name=.../>
```

... (default predicate-argument IDs for verbs with no entry in existing valency lexicons for German)

```
   </lexicon>
   <lexicon id="gup" name="gup">
    <lexref>http://alanakbik...
     </lexref>
    <lexbrowsing>http://alanakbik...
     </lexbrowsing>
    <lexsearching>http://alanakbik...
     </lexsearching>
    <argumentsused>
     <argdesc id="vecargA0">
      <comesfrom lexicon="gup"/>
      <label>Arg0</label>
      <shortlabel>A0</shortlabel>
     </argdesc>
     <argdesc id="vecargA1">
      ...
     </argdesc>
     ...
    </argumentsused>
   </lexicon>
   <lexicon>
    ...
   </lexicon>
  </reflexicons>
 </header>
 <body>
  <veclass id="vec00201"
   lemma="einwenden
   (\ssclass{}-ID-vec00201-de-cm00026)">
   <classmembers>
    <classmember id="vec00201-de-cm00016"
      idref="GUP-ID-argumentieren-01"
      lang="de" status="yes"
      lexidref="gup"
      lemma="argumentieren">
     <maparg>
      <argpair>
```

Argument-Role mapping here: A0 → Arguer

```
       <argfrom idref="vecargA0">
        <form/>
        <spec/>
       </argfrom>
       <argto idref="vecroleArguer"/>
      </argpair>
```

... (other argument to semantic roles mappings)

```
     </maparg>
     <restrict/>
     <cmnote/>
```

```
<extlex idref="gup" no_mapping="0">
 <links>
    Links to external lexicon
    here: the German UPB ("gup")
  <link predicate="argumentieren"
    rolesetid="01"
    filename="argumentieren"
    divid="argue.01"/>
 </links>
</extlex>
<extlex idref="fnd" no_mapping="0">
 <links>
    Links to external lexicon
    here: the German FrameNet ("fnd")
  <link frameid="937"
    framename="Begründen"/>
 </links>
</extlex>
    ... (more links to external German
    lexicons, such as E-VALBU or Woxikon[21])
<examples>
  <example corpref="paracrawl_ge"
  frpair="argue.argumentieren"
  nodeid="G-vec00060-001-s040"/>
 </examples>
</classmember>
<classmember id="vec00201-de-cm00017"
  ...
  lemma="argumentieren">
 ...
</classmember>
 ... (more German classmembers)
 </classmembers>
</veclass>
... (more classes)
</body> </synsemclass_DE>
```

In the above simplified example of the language-dependent file structure, the header contains some versioning information, list of allowed users (= annotators and maintainers), and list of pre-existing lexicons for the particular language (German in this case) to which the individual class members are being linked. Some of these pre-existing lexicons contain predicate-argument structure information to which the semantic roles of the class are mapped (in the above example, it is GUP[22] and E-VALBU (Kubczak, 2014; Schumacher et al., 2018)[23] for German).

The body element of the file contains the class members as assigned to the classes defined in the main file, by means of reference (e.g., `vec00201`), stating also the class name and ID of the references lemma in German. The individual class member entries contain the usual parts - the lemma and reference ID to one of the defining predicate-argument structure lexicons, then argument mapping(s) to semantic roles of the referenced class, links to external lexicons (the `extlex/link` element(s)), notes, and links to example sentences (here, from the ParaCrawl English-German corpus).

4.3 The SynEd Editor

The SynEd editor (Fig. 2) - in its "stand-off" version capable of working with one or (only) a few languages - has the following features:

- It can be customized to work with external lexicons (lexical resources) for the given language(s).
- It works with any number of language-specific files; these are typically two: English or some other already included language (in a "read-only" mode), and the language being added and worked on.
- It allows for marking the pre-extracted class members as OK (yes, to be kept) or as "no" meaning "to be deleted" (in fact, it allows for even more fine-grained distinctions, on a five-value scale: both "yes" and "no" have a weaker version ("rather yes/no") and there is also the possibility of marking the word as "undecided"; all decisions undergo a review by the maintainer).
- It allows for creating and editing the mapping of the semantic roles defined for the given class to syntactic arguments of the word (verb) in question (if some lexical resource describing these arguments exists).
- It allows for adding links (Fig. 3) to existing external lexical resources, such as WordNet or any other resource available on the web.
- It allows searching by lemma (cs, en, de) (Fig. 4), by semantic role to find classes that contain it (Fig. 5) or by class ID (Fig. 6).
- It allows for selecting textual examples from a user-defined language-specific corpus (if available, a parallel one), to exemplify the particular word sense or use of the word being assigned to the class as a class member.

[21] https://synonyme.woxikon.de

[22] GUP stands for [German] Universal Propositions Bank (Akbik et al., 2016), see https://github.com/UniversalPropositions/UP-1.0/tree/master/UP_German

[23] E-VALBU stands for Elektronisches Valenzlexikon des Deutschen, see https>//https://grammis.ids-mannheim.de/verbvalenz

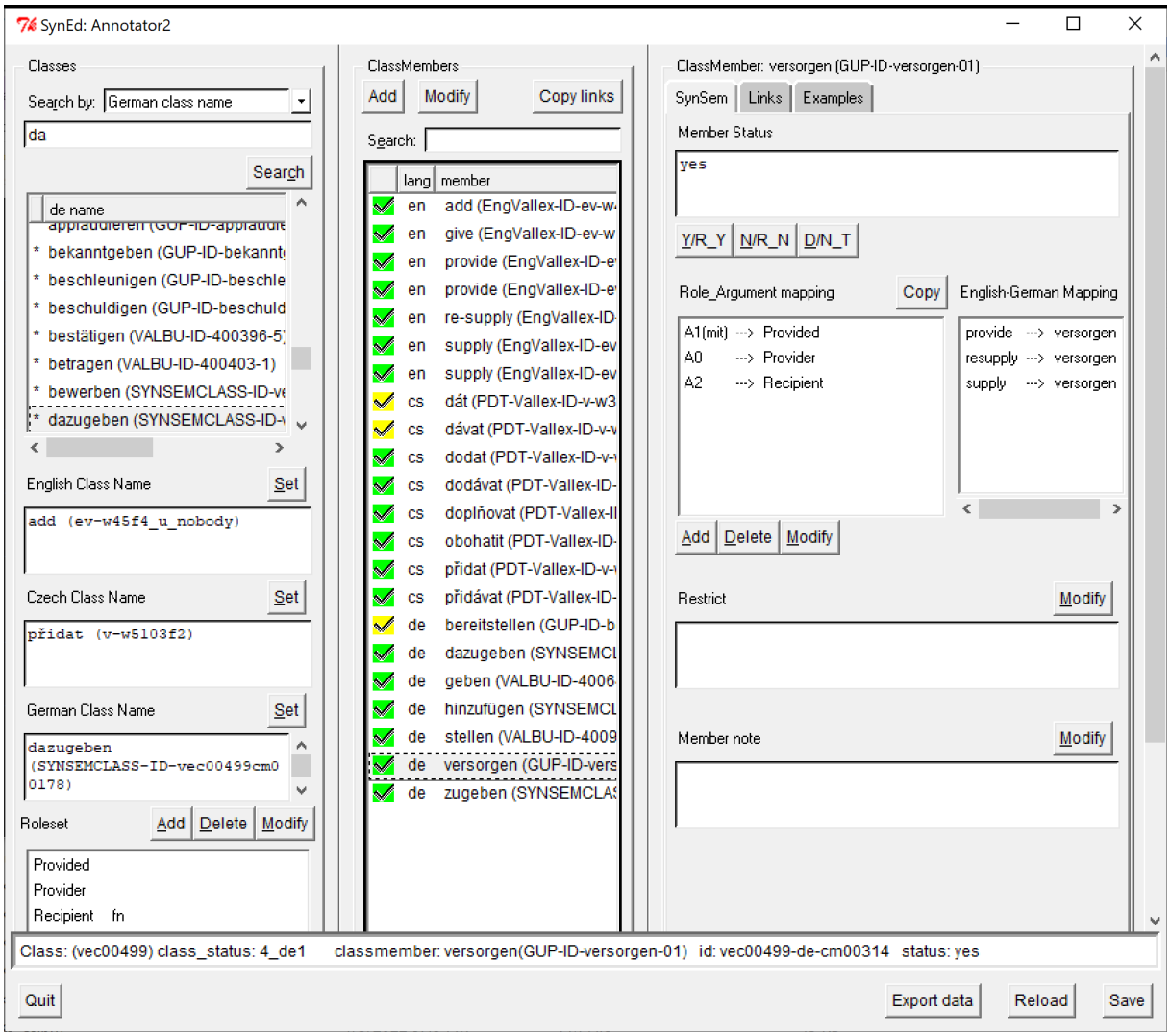


Figure 2: The "add" class with cs/en/de entries; "versorgen" (German) is highlighted to show mapping to roles

The editor allows to **edit the class members** for classes referred to in the language-specific file (Sect. 4.2). The typical workflow, as specified in the common guidelines (Urešová et al., 2019), asks for first pruning the pre-fetched class member candidates (middle column in Fig. 2), using the corresponding examples from the input parallel or monolingual corpus.

After filtering out unsuitable class members, the annotator proceeds via the editor **to map the semantic roles to the arguments** of the predicates represented by the class members (in the SynSem tab with the Role_Argument mapping window). The arguments are taken from the external valency or similar resources (defined for each language in the language-dependent file) if they exists; if not, special IDs are generated.

Next, in the Link tab, the editor allows to **edit links to other external semantic resources** (Fig. 3).

Finally, the editor allows to **select examples**; typically, 5 to 10 examples from the corpora used for pre-selection are marked and stored with the class member (in the Examples tab).

Language-specific annotators are not allowed to edit the main file or its parts, but they can suggest changes by means of GitHub issues, by emailing the central maintainers, or (if trained) by creating pull requests for the main file. It has to be stressed again that it is then the responsibility of the central maintainers to implement these changes carefully, since some changes may require that all languages be updated. This update might not be readily feasible, might need the cooperation (read: manual edits, or at least a check) by the maintainers of all language-dependent files, and must be scheduled and possibly discussed carefully in some form of, e.g., maintainers forum.

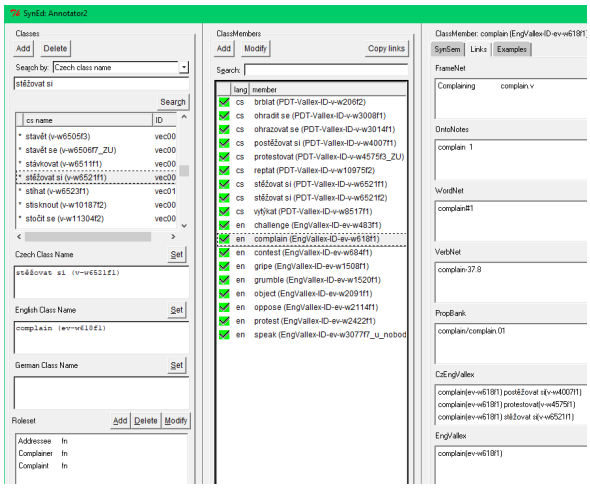


Figure 3: The “stěžovat si/complain” class with external “Links” for CM “complain” (on the right-hand side)

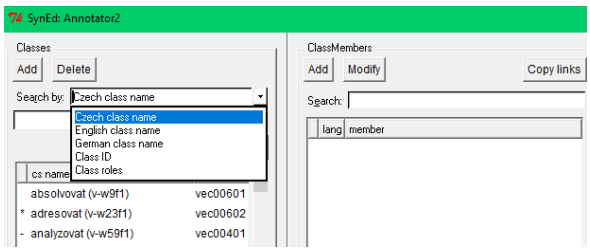


Figure 4: The search possibilities in the editor

5 The Current State of SynSemClass and SynEd

The latest release of SynSemClass ontology (SynSemClass 4.0)[24] is already in a stand-off format; it underwent both a detailed and intensive annotation check and contains new features, such as semantic role definitions, semantic role hierarchy, aspect verb pairs replenishing, search, etc.

The alignment within the input parallel corpus has been done by MGIZA++ (Gao and Vogel, 2008) and then from there, the initial language-specific file is created by a specific script that will be also released as part of the language-specific setup guidelines.

SynEd is available currently in an experimental version,[25] which allows for editing selected language-dependent files, and for the administrator and main maintainer also to edit the language-independent part in the main file. Classes can be searched for in the editor by a name in any language (of those selected for editing). The mappings to

[24] http://hdl.handle.net/11234/1-4746

[25] http://github.com/fucikova/SynSemClass_multi

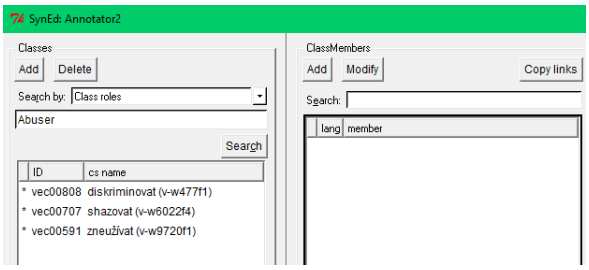


Figure 5: The results of searching the role “Abuser”

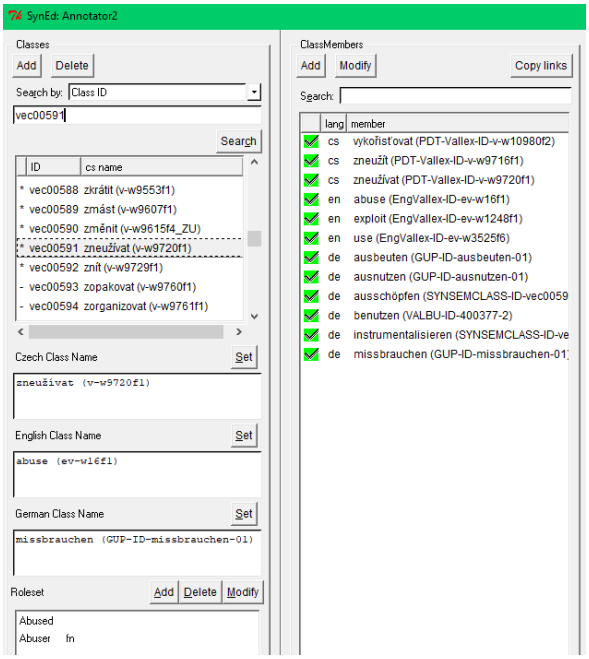


Figure 6: The results of searching by class ID - the class vec00591 “zneužívat”

syntactic properties of the individual class members, which are also language-dependent, can be created to multiple sources (a feature added while working on German, which does not have one large coverage valency dictionary, and therefore several of them had to be used). The external links and examples are shown and amended or added again only for the class member in “its” language, as configured in the language-dependent file. The editor, once fully tested on at least one more language will be also released publicly.[26]

6 Conclusions and Future Work

We have demonstrated a gradual approach to adapting a data specification and an annotation tool and the associated workflow to a multi-language, or in other words, partly language independent

[26] The editor will be available under Mozilla Public License 2.0 (MPL-2.0), presumably with SynSemClass version 5.

model, allowing concurrent annotation by independent teams working on the individual languages. Each new version of all the components is based on a practical experience with the previous version. While inspired very much by the UD approach to adding, maintaining, validating and publishing new datasets and languages. However, work on a multilingual ontology has one substantial difference: the amount of data that are language independent, but which are heavily being linked to from the language-dependent parts is much larger and will be changing often.

The experimental version of SynEd described herein is now being used for adding Spanish (Fernández-Alcaina et al., 2023).[27] Both the Spanish data and the editor will be publicly released as version 5.0 of SynSemClass.

Part of future work - once the components are in place - will be to open the development to the community, interested in similar resources, and also to develop tools that would allow possible (semi-)automatic "conversions" of those resources to the SynSemClass set.

The web application that shows the then-current version of SynSemClass is in place.[28] It is automatically generated from the dataset. However, at the moment it lacks more advanced search features that would serve possible more complex research tasks on this resource - another future work item.

Acknowledgements

The work described herein has been supported by the Grant Agency of the Czech Republic under the EXPRO program as project "LUSyD" (project No. GX20-16819X) and uses resources hosted by the LINDAT/CLARIAH-CZ Research Infrastructure (project No. LM2018101, supported by the Ministry of Education of the Czech Republic).

References

Alan Akbik, Xinyu Guan, and Yunyao Li. 2016. Multilingual aliasing for auto-generating proposition Banks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3466–3474, Osaka, Japan. The COLING 2016 Organizing Committee.

Emhimed Alatrish. 2012. Comparison of ontology editors. *ERAF Journal on Computing*, 4:23–38.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998a. The Berkeley FrameNet Project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 1*, ACL '98, pages 86–90, Stroudsburg, PA, USA. Association for Computational Linguistics.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998b. The Berkeley FrameNet Project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 1*, ACL '98, pages 86–90, Stroudsburg, PA, USA. Association for Computational Linguistics.

Jinho D. Choi, Claire Bonial, and Martha Palmer. 2010. Propbank Frameset Annotation Guidelines Using a Dedicated Editor, Cornerstone. In *International Conference on Language Resources and Evaluation*.

Silvie Cinková, Eva Fučíková, Jana Šindlerová, and Jan Hajič. 2014. EngVallex - English Valency Lexicon. http://hdl.handle.net/11858/00-097C-0000-0023-4337-2, LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics, Charles University.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. Language, Speech, and Communication. MIT Press, Cambridge, MA.

Cristina Fernández-Alcaina, Eva Fučíková, and Zdeňka Urešová. 2023. Spanish verbal synonyms in the synsemclass ontology. In *Proceedings of the 21st TLT conference*, pages 1–12. Georgetown University in Washington D.C.

Charles J. Fillmore. 2002. Linking sense to syntax in FrameNet. In *Proceedings of 19th International Conference on Computational Linguistics*, Taipei. COLING, COLING.

Thierry Fontenelle. 2003. FrameNet and Frame Semantics. *International Journal of Lexicography*, 16(3):231–231.

Nabil Gader, Veronika Lux-Pogodalla, and Alain Polguère. 2012. Hand-crafting a lexical network with a knowledge-based graph editor. In *Proceedings of the 3rd Workshop on Cognitive Aspects of the Lexicon*, pages 109–126, Mumbai, India. The COLING 2012 Organizing Committee.

Qin Gao and Stephan Vogel. 2008. Parallel implementations of word alignment tool. In *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, pages 49–57, Columbus, Ohio. Association for Computational Linguistics.

Jan Hajič, Eva Hajičová, Jarmila Panevová, Petr Sgall, Silvie Cinková, Eva Fučíková, Marie Mikulová, Petr Pajas, Jan Popelka, Jiří Semecký, Jana Šindlerová, Jan Štěpánek, Josef Toman, Zdeňka Urešová, and Zdeněk Žabokrtský. 2012. Prague Czech-English Dependency Treebank 2.0. https://hdl.handle.

[27] This paper also discusses issues related to use of the editor, e.g., how long it takes to annotate an entry, etc.

[28] https://lindat.cz/services/SynSemClass

net/11858/00-097C-0000-0015-8DAF-4, LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Tom Kenter, Tomaž Erjavec, Maja Žorga Dulmin, and Darja Fišer. 2012. Lexicon construction and corpus annotation of historical language with the CoBaLT editor. In *Proceedings of the 6th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, pages 1–6, Avignon, France. Association for Computational Linguistics.

Jacqueline Kubczak. 2014. Valenzwörterbuch e-VALBU. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Markéta Lopatková, Václava Kettnerová, Anna Vernerová, Eduard Bejček, and Zdeněk Žabokrtský. 2020. VALLEX 4.0. https://hdl.handle.net/11234/1-3524, LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*, pages 1659–1666, Paris, France. European Language Resources Association.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An Annotated Corpus of Semantic Roles. *Computational Linguistics*, 31(1):71–106.

Gautam K. Parai, Clement Jonquet, Rong Xu, Mark A. Musen, and Nigam H. Shah. 2010. The Lexicon Builder Web service: Building Custom Lexicons from two hundred Biomedical Ontologies. In *American Medical Informatics Association Annual Symposium, AMIA'10*.

Sameer S. Pradhan and Nianwen Xue. 2009. OntoNotes: The 90% solution. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Tutorial Abstracts*, pages 11–12, Boulder, Colorado. Association for Computational Linguistics.

Karin Kipper Schuler. 2006. *VerbNet: A Broad-Coverage, Comprehensive Verb Lexicon*. Ph.D. thesis, University of Pennsylvania.

Helmut Schumacher, Jacqueline Kubczak, Renate Schmidt, and Vera de Ruiter. 2018. *VALBU - Valenzwörterbuch deutscher Verben*. Narr, Tübingen.

Wolfgang Teubert. 2007. *Text Corpora and Multilingual Lexicography*. John Benjamins Publishing Company.

Zdeňka Urešová, Alevtina Bémová, Eva Fučíková, Jan Hajič, Veronika Kolářová, Marie Mikulová, Petr Pajas, Jarmila Panevová, and Jan Štěpánek. 2021. PDT-vallex: Czech Valency lexicon linked to treebanks 4.0 (PDT-Vallex 4.0). https://hdl.handle.net/11234/1-3499, LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Zdeňka Urešová, Eva Fučíková, Jan Hajič, and Jana Šindlerová. 2015. CzEngVallex - Czech English Valency Lexicon. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics, Charles University, http://hdl.handle.net/11234/1-1512.

Zdeňka Urešová, Eva Fučíková, and Eva Hajičová. 2019. Czengclass: Contextually-based synonymy and valency of verbs in a bilingual setting. Technical Report 62, ÚFAL MFF UK, Prague, Czechia.

Zdeňka Urešová, Eva Fučíková, Eva Hajičová, and Jan Hajič. 2018a. Creating a verb synonym lexicon based on a parallel corpus. In *Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC 2018)*, pages 1432–1437, Paris, France. European Language Resources Association.

Zdeňka Urešová, Eva Fučíková, Eva Hajičová, and Jan Hajič. 2018b. Tools for Building an Interlinked Synonym Lexicon Network. In *Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC'18)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Zdeňka Urešová, Karolina Zaczynska, Peter Bourgonje, Eva Fučíková, Georg Rehm, and Jan Hajič. 2022. Making a semantic event-type ontology multilingual. In *Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022)*, pages 1332–1334, Marseille, France. European Language Resources Association.

Zdeňka Urešová, Eva Fučíková, Eva Hajičová, and Jan Hajič. 2020. SynSemClass linked lexicon: Mapping synonymy between languages. In *Proceedings of the 2020 Globalex Workshop on Linked Lexicography*, pages 10–19, Marseille, France. European Language Resources Association.