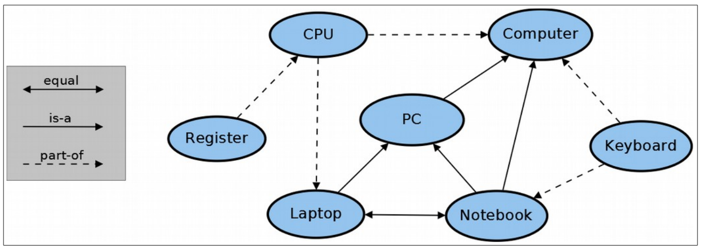
SemRep

User's Guide

1 Overview

1.1 Introduction

SemRep is a semantic repository to import and combine different lexicographic resources. These resources consist of relations of the kind ***Concept A [Type] Concept B***, like *Hand part-of Arm*. As illustrated in the figure below, SemRep stores these lexicographic relations in a graph-like structure and allows the determination of the relation type holding between two concepts. It is also able to determine the type across several relations (indirect paths). For instance, SemRep can determine an is-a relation between *Laptop* and *Computer* if the two relations *Laptop is-a PC* and PC *is-a Laptop* are contained. In this case, a path of length 2 is found between *Laptop* and *Computer.* SemRep allows the resolution up to paths of length 4, though the default configuration is 2. Longer paths increase the execution time and reduce the precision of the relation type determination, but are able to determine the relation types between more difficult (or semantically more distinct) concepts.



In this brief manual, the use of SemRep in the context of mapping enrichment and semantic matching is explained. Because of space issues, most of the technical and implementation details are disregarded or only shortly explained. They are described in our scientific publications, especially in [1].

1.2 Using SemRep

There are two ways to use SemRep:

(1) Using the **console-based user interface**. This allows to manually check concepts and paths, determine the relation type, display statistical information or carry out advanced tasks (e.g., mapping processing from an input file).

(2) Using the **API** (jar file). This allows to use SemRep within another program resp. software project.

Using the console-based user interface is described in Section 3. In Section 4, the Java API is shortly introduced.

1.3 Resources

Currently, the following resources are provided:

- Wikipedia Data Set + Wikipedia Field References
- Wikipedia Redirects
- Princeton WordNet 3.0
- UMLS MetaThesaurus
- OpenThesaurus (German)
- ConceptNet

Note that *ConceptNet* is not used by default, but can be optionally enabled. Usually, SemRep loads all resources that are enabled in the properties file (see below). However, it is also possible to load a single resource instead of all data sets, or a selection of single resources. The Wikipedia data sets have been automatically extracted and are described in [2, 3].

The Wikipedia Data Sets consists of two subsets:

- Wikipedia
- Wikipedia Filtered

Wikipedia Filtered is a subset of the Wikipedia data set where most entities such as persons, geographic places, book titles, companies, events, movies and the like were removed and mostly concept terms remain. As SemRep is rather designed for schema and ontology mapping, such entities are rather unimportant and can impair the overall quality of SemRep. For this reason, *Wikipedia Filtered* is the default data set to be loaded and consists of about 570,000 concepts and 1.4 million relations in contrast to the much more comprehensive Wikipedia data set (about 4.3 million concepts and 12 million relations). The precision of the automatically extracted relations may be about 80 %.

Each resource is of a specific language. SemRep currently provides English and German as selectable languages. All concepts are treated case-insensitive, e.g., there is no difference between the concepts “student” and “Student”. SemRep is generally extendable by new resources. This requires changes of the source code, though.

1.4 Configuration

SemRep is highly configurable. Some configuration parameters are essential to correctly execute SemRep, while other parameters are used for the technical configuration of SemRep. All parameters are configured in a properties file in the bin/resources directory. Please see our publications for detailed information about the different parameters.

The properties file consists of three elementary parts:

Resource loading options These parameters allow to enable a resource (“true”) or to disable it (“false”). If the entire repository is loaded, those enabled resources will be regarded; disabled resources will be ignored. However, a disabled resource can always be manually loaded if necessary.

Repository parameters These parameters allow to configure SemRep. Changes will impair the way how SemRep calculates paths and how it handles results. The default values will generally lead to the best results.

Confidence values The confidence values specify how specific relation types and resources are treated for path calculation.

SemRep provides an auto-stemming function that can be set in the properties file. If this function is enabled, it will automatically stem the concepts from the resource files. Given two concepts like *computer* and *computers*, SemRep would only create one node *computer* instead of two different ones. Stemming requires some extra-time when the repository is loaded.

SemRep uses an implementation of the Pling Stemmer to stem English concepts. For German stemming, the Apache Lucense stemmer from the Apache Software Foundation is used.

1.5 Import Files

Each lexicographic resource is represented by a single text file, and each line represents exactly one lexicographic (semantic) relation. Each line carries three pieces of information, which are separated by two consecutive colons: First comes the source concept, then the target concept and finally the relation type. Relation types are encoded by a digit. It holds: 0 = equality, 1 = is-a, 2 = inverse is-a, 3 = has-a and 4 = part-of. All files are located in the directory "Repository", which has to be specified in the properties file.

1.6 Technical Requirements

SemRep was developed and tested on a Windows Server with two Intel XEON E5540 CPUs. To load the entire repository, at least 4 – 6 GB RAM are required. Loading just selected resources requires less memory. Depending on the system, loading the full repository takes between 1 and 2 minutes of time. If auto-stemming is used, the setup time increases to 6 – 7 minutes. In case of a memory leak, the setup time can become very high and SemRep may not perform conveniently anymore.

2 Technical Overview

2.1 Path Search

SemRep searches paths of a maximum path length *L* that is specified in the properties file (key: *maximal_path_length*). SemRep allows maximum path lengths of 2, 3 or 4 (setting a value of 1 will still result in SemRep searching for paths up to length 2). The path length is defined as the number of links in the path. The following path has a length of 2:

mouse [is-a] rodent [is-a] animal

SemRep can be used in two modi: *First path* and *all paths*. In the properties file, the option can be set using the boolean entry *all_paths_option*. The **first path** option will return the first valid path SemRep found between source and target concept (and will then stop the path search immediately). This will always be the shortest possible path, because SemRep first calculates paths of length 1, then of length 2 and so forth. However, the first path is not necessarily the path having the highest score.

To obtain the path having the highest score, the **all paths** option has to be used. SemRep will then calculate all possible paths within the specified length L and return the path having the highest score. Experiments showed that this option scarcely leads to better results, yet increases the execution time excessively. Thus, first path is the default option (*all_paths_option* = false).

Each valid path produced by SemRep must have a minimal score to be accepted. This score can be set in the properties file (key: *acceptance_threshold*) and is 0.4 by default.

Details of the scoring function will not be explained in this manual, but have been described in our publications.

2.2 Relation Types

SemRep can return 6 relation types:

Relation Type	**Equivalent in Semantics**
equal	Synonymy
is-a	Hypnerymy
inverse is-a	Hyponymy
has-a	Holonymy
part-of	Meronymy
related	Co-hyponymy (same semantic level of concepts required)

2.3 Preprocessing

SemRep automatically performs the following forms of preprocessing if necessary:

- Stemming, when stemming is enabled
- Genitive Back Transformation
- Gradual Modifier Removal on either Concept 1 or Concept2
- Same Modifier Removal
- Same Head Removal
- Fuzzy Matching

Genitive Back Transformation

This option transforms an of-genitive expression in its equivalent compound form. The feature only works for the English language and is only applied if the expression itself is not contained in the repository. Stemming is automatically reapplied after the transformation took place.

Examples:

- University of Leipzig → Leipzig University
- Aorta of the neck → Neck aorta
- Cells of the brain → brain cells → brain cell

Note that transformations from of-genitive to s-genitive or compound usually do not impair the semantic and that this form of preprocessing can increase the recall, but will most likely not reduce the precision.

Gradual Modifier Removal

In case that an open compound is not contained in the repository, the first (left-hand) modifier is removed and it is analyzed whether the reduced word is now contained in the repository. This form of step-by-step modifier removal continues until the word is either found or the compound head is reached. The removal steps are naturally regarded when the final path type is computed.

Example: Concept is *Ultra-light office laptop*

- Step 1: Ultra-light office laptop – NOT CONTAINED!
- Step 2: Office laptop – NOT CONTAINED!
- Step 3: laptop – CONTAINED

SemRep will work with the concept *laptop* then, yet assumes that ultra-light office laptop is a specification of a laptop.

Note: GMR has no influence on the maximum path length.

Same-Modifier-Removal

If the two input concepts are open compounds and possess the same first (left-hand) modifier, these modifiers are both removed if at least one of the concepts is not contained in SemRep.

Example: *kitchen chair – kitchen seat*

SemRep will work with *chair* and *seat* if either of the two terms is not part of SemRep.

Same-Head-Removal

If two input concepts are open compounds and share the same head, these heads are both removed if at least either of the concepts is not contained in SemRep. In contrast to Same-Modifier-Removal, this step cannot always produce a semantic relation.

Example: Apple cake – fruit cake

SemRep removes 'cake' and works with 'apple' and 'fruit' instead. It will conclude that an apple cake is a fruit cake, because of the is-a relation between cake and fruit.

Example: Station hall – City hall

SemRep removes 'hall' and works with 'station' and 'city' instead. It will not conclude any type, though, as it cannot find any sensible (is-a or equal) relation between station and city. SemRep returns "undecided".

Fuzzy Search (Matching)

If a concept cannot be found in SemRep, even with the previously described means of preprocessing, SemRep can use the Fuzzy Search strategy which is based upon lexicographic matching very similar to the matching used in classic schema and ontology matchers. Given a concept C that is not contained in SemRep, the fuzzy matcher compares C with any concept X in the repository which is of the same language as C. The concept X for which holds *similarity(C, X)* is maximal is used instead of C then.

Fuzzy search can be enabled or disabled in the properties file (key: *fuzzy_search*). It can improve the recall, yet reduce the precision.

Example: Input concepts are *housex* and *doorx*.

Both concepts are not part of the repository and cannot be preprocessed. Fuzzy search is a search for the next-closest concepts in the repository based on lexicographic similarities (Trigram, Jaccard and Jaro-Winkler distance). In this case, *house* and *door* are the most similar concepts and will be used to process the query.

Fuzzy search may require some extra seconds per query. Therefore, it should be disabled if a fast execution time is preferred against a better recall.

2.4 Post-processing

Post-processing comprises two possible steps:

- Post Gradual Modifier Removal (Post-GMR)
- Node Degree Heuristics

Post Gradual Modifier Removal is used if at least one input concept is an open compound and SemRep could not produce any result. SemRep now forced GMR on this concept, even though it is contained in SemRep. At times, SemRep can conduct the correct relation type this way, while it would fail without it.

Node Degree Heuristics is the last (and most vague) means of relation type determination if all previous steps failed and no relation type could be determined. Based on the node degree of the two matching concept nodes, the node degree quotient is calculated as follows:

max([degree(node1) / degree(node2)], [degree(node2) / degree(node1)])

If the quotient exceeds a specified threshold T, SemRep suggests is-a resp. inv. is-a. The value can be set in the properties file (key: *degree_comparison_factor*).

Example: SemRep could not determine the type between *convertible* and *car.* It holds *degree(convertible)* = 55 and *degree(car)* = 598. The quotient of 10.9 is above the threshold of 1.5 and thus indicates that *convertible* is something more specific than *car.* Consequently, is-a is returned.

2.5 Universal Concept Filtering

Universal concepts (also called top-level concepts in previous work) are very general, rather meaningless concepts usually found very high up in the lexicographic taxonomy. Examples include terms like *object, entity, thing, being* or *place.* There are also many further concepts which seem less "universal", but are still rather meaningless (e.g., *structure, place, unit, form* etc.).

Such universal concepts can impair queries, because they often link rather unrelated concepts as a result of their universality. Consider the two concepts *car* and *computer*. Apparently, both concepts are objects, but they are semantically not related. Having links from *car* to *object* and *computer* to *object*, can put them in an indirect relationship (co-hyponyms) and thus lead to a conclusion that *car* and *computer* are related.

Therefore, universal concepts should be blocked from the repository. SemRep marks these concepts and makes sure that they do not appear in a path such as:

Computer [is-a] Object [inv. is-a] Car

However, these objects are not entirely deleted from the repository, as queries containing a universal concept as input term will be still executed. Thus, the following path can be returned (as *object* is the target term of the query):

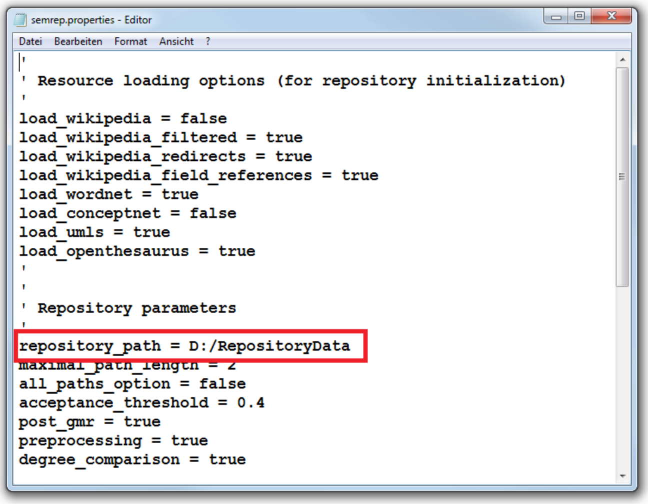
Computer [is-a] Object

If the entire repository is loaded, universal concept filtering is automatically applied. If specific resources are loaded, the user can decide whether he wants to apply universal concept filtering or not.

The file containing all universal concepts can be found in the *prog* directory. Each line of the file *universal_concepts.txt* contains a term or list of terms, separated by a '#'. The lines starting with "Wiki:" indicate terms that are only blocked for relations coming from the Wikipedia resources. Lines starting with an apostrophe are ignored. As matter of fact, this file can be extended or adopted for advanced usage.

3 Console-based Interface

The console-based interface allows you to use SemRep manually. It can be easily started by running the *Run.bat* file in the SemRep directory. Entering the keyword **help** will display a list of all accepted commands together with a short description. Some commands allow an additional parameter -v. This parameter leads to more detailed information ("verbose").

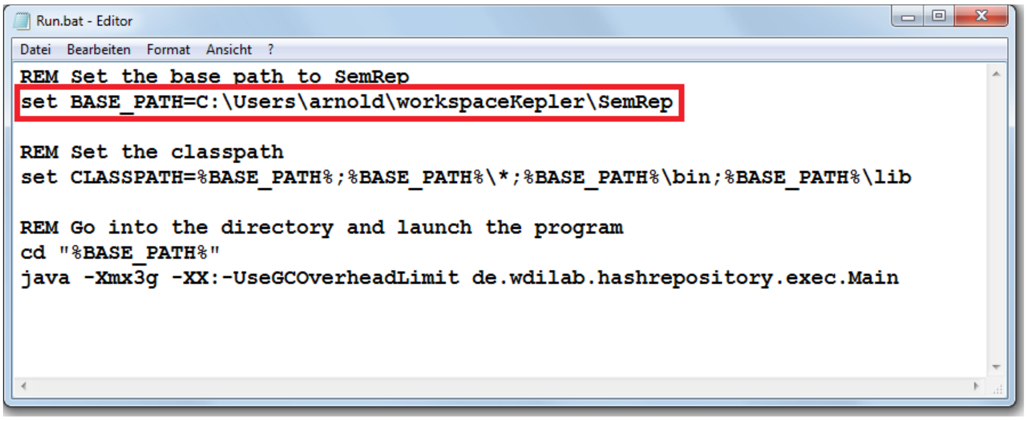


```
'
' Resource loading options (for repository initialization)
'
load_wikipedia = false
load_wikipedia_filtered = true
load_wikipedia_redirects = true
load_wikipedia_field_references = true
load_wordnet = true
load_conceptnet = false
load_umls = true
load_openthesaurus = true
'
'
' Repository parameters
'
repository_path = D:/RepositoryData
maximal_path_length = 2
all_paths_option = false
acceptance_threshold = 0.4
post_gmr = true
preprocessing = true
degree_comparison = true
```

3.1 Installation

Please carry out the following steps to install SemRep:

1) Download the SemRep archive. Unzip the file and save it in any directory of your choice.
2) Edit the Run.bat file and set the correct path to the SemRep directory in the second line (see the screenshot below).
3) Download the resource folder (semrep_resources.zip). Unzip the file and save it in any directory of your choice.
4) Open the properties file "semrep.properties" (directory SemRep/prog/). Set the correct path to the repository data directory (semrep_resources) as shown in the screenshot above.
5) Start the Run.bat (possibly as administrator) to eventually launch SemRep.

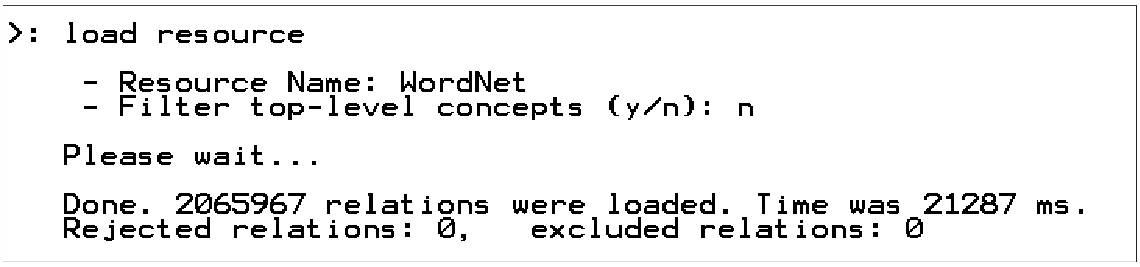


```
REM Set the base path to SemRep
set BASE_PATH=C:\Users\arnold\workspaceKepler\SemRep

REM Set the classpath
set CLASSPATH=%BASE_PATH%;%BASE_PATH%\*;%BASE_PATH%\bin;%BASE_PATH%\lib

REM Go into the directory and launch the program
cd "%BASE_PATH%"
java -Xmx3g -XX:-UseGCOverheadLimit de.wdilab.hashrepository.exec.Main
```

3.2 Loading the Repository

Before SemRep can be used, the repository has to be loaded. The usual way to load resources is to use the following command: **load repository**. Consequently, SemRep will import all resources that are enabled in the properties file. By default, this includes the Wikipedia data sets, WordNet, OpenThesaurus and UMLS. After this step, SemRep can be fully used.

Instead of loading all resources, it is also possible to load a selected resource, e.g., WordNet. In this case, the command **load resource** has to be used. Subsequently, SemRep asks what kind of resource should be used. Enter the resource to be loaded, e.g., *WordNet* or *Wikipedia Filtered*. A list of all resources can be obtained using the command **show resources**. Additionally, SemRep asks whether top-level concepts (universal concepts) should be blocked or not. By default, universal concepts are always blocked, so enter 'y' if this default option should be used. If they should not be blocked, enter 'n'.



```
>: load resource

    - Resource Name: WordNet
    - Filter top-level concepts (y/n): n

   Please wait...

   Done. 2065967 relations were loaded. Time was 21287 ms.
   Rejected relations: 0,   excluded relations: 0
```

Note: In previous versions, blocking universal concepts implicated the full absence of these terms from the repository, i.e., the terms and all their relations were not imported. In the current version of SemRep, the terms are not blocked from the repository, however, valid result paths will not contain blocked concepts in their path as inner nodes. For example, *structure* is one of the universal words. SemRep will thus consider the path *house* – *structure* – *door* as invalid, because *structure* appears as an inner node. However, a path between *house* and *structure* is generally possible. SemRep thus allows to handle correspondences where either source or target node is a universal concept.

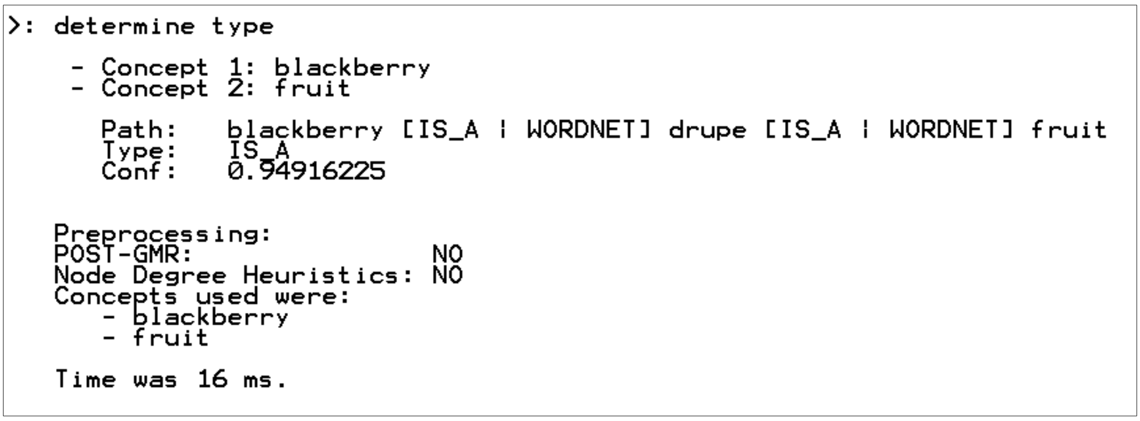
After a resource has been loaded, it is also possible to import further resources. The already imported relations will not be overwritten, i.e., new resources will be simply added to the already loaded relations. This way, it is also possible to load more than one selected resource, but still without loading the entire repository.

Loading all resources should require about 1 – 2 minutes. If auto-stemming is used, the loading time increases to about 6 – 7 minutes.

3.3 Determining Relation Types

The core feature of SemRep is the determination of relation types. Given two input concepts A and B, SemRep tries to find paths from A to B and to determine the semantic type holding between the two concepts.

To determine the relation type, enter the command **determine type**. Subsequently, the two input concepts have to be specified. SemRep will then show the most relevant (highest scored) path, the relation type that was calculated and the confidence value between 0 and 1. The path shows the link from source to target concept. Relations are represented in the form [Type | Resource], i.e., SemRep shows what kind of relation holds between the concepts in the path and from which resource it has been used. In the example below, there is a path from *blackberry* to *fruit* across the *drupe* concept. There are two is-a relations in between, which originate from the resource WordNet.



```
>: determine type

    - Concept 1: blackberry
    - Concept 2: fruit

      Path:    blackberry [IS_A | WORDNET] drupe [IS_A | WORDNET] fruit
      Type:    IS_A
      Conf:    0.94916225


   Preprocessing:
   POST-GMR:                  NO
   Node Degree Heuristics:  NO
   Concepts used were:
      - blackberry
      - fruit

   Time was 16 ms.
```

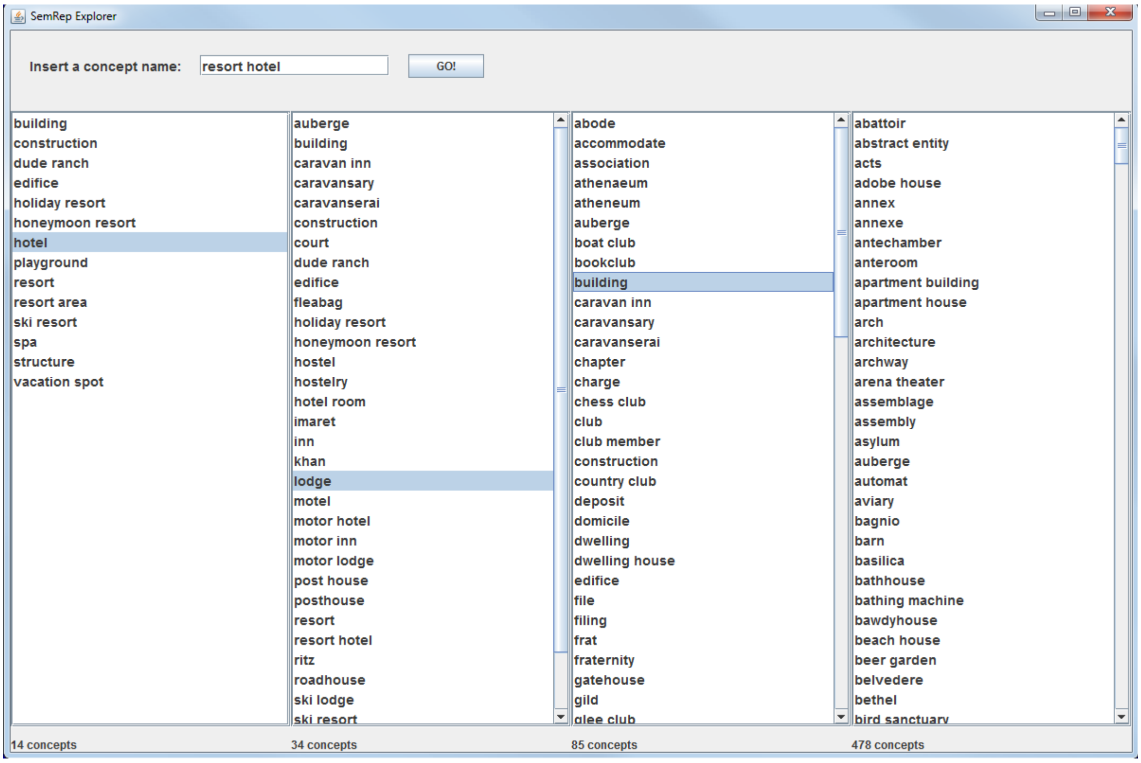
Additionally, information about preprocessing, Post Gradual Modifier Removal, Node Degree Heuristics and the used concepts are given. The latter denote the concepts after preprocessing has been applied (e.g., after GMR or Fuzzy Matching). Please see our publications for detailed information.

Using the -v parameters shows a complete list of paths that have been found from source to target node. The paths are ordered by their confidence. This option is useful to obtain an overview about the different paths existing between source and target concept. Note that the maximum path length L specifies how many paths SemRep will ultimately calculate.

Using the default configuration of SemRep, average execution times should be between 1 and 50 ms. The execution times depend highly on the used input concepts and how many resources have been loaded. More general concepts (concepts having many links to other concepts) lead to longer execution times compared to less general concepts. If longer path lengths are used, execution times will increase dramatically. For maximum path lengths of 4, execution times will usually take several seconds.

3.4 Using the Explorer

In order to obtain a better overview of the repository, the explorer can be used to investigate how concepts are interrelated. The explorer does not show the relation types between concepts, but only the way how they are linked. To start the explorer, enter the command **run explorer**. Then, enter a term in the text field and click "GO!". In the first column, all neighbors of the concept are listed. Selecting one of these concepts will show all neighbors of the concept in the second column and so forth.



Insert a concept name: resort hotel

building
construction
dude ranch
edifice
holiday resort
honeymoon resort
hotel
playground
resort
resort area
ski resort
spa
structure
vacation spot

14 concepts

auberge
building
caravan inn
caravansary
caravanserai
construction
court
dude ranch
edifice
fleabag
holiday resort
honeymoon resort
hostel
hostelry
hotel room
imaret
inn
khan
lodge
motel
motor hotel
motor inn
motor lodge
post house
posthouse
resort
resort hotel
ritz
roadhouse
ski lodge
ski resort

34 concepts

abode
accommodate
association
athenaeum
atheneum
auberge
boat club
bookclub
building
caravan inn
caravansary
caravanserai
chapter
charge
chess club
club
club member
construction
country club
deposit
domicile
dwelling
dwelling house
edifice
file
filing
frat
fraternity
gatehouse
gild
glee club

85 concepts

abattoir
abstract entity
acts
adobe house
annex
annexe
antechamber
anteroom
apartment building
apartment house
arch
architecture
archway
arena theater
assemblage
assembly
asylum
auberge
automat
aviary
bagnio
barn
basilica
bathhouse
bathing machine
bawdyhouse
beach house
beer garden
belvedere
bethel
bird sanctuary

478 concepts

3.5 Loading a Test Resource

SemRep contains a test resource in the program directory. This resource can be loaded using the **load resource** command and then enter the name "Test". As matter of fact, this test resource can be updated, extended or replaced by any other resource. It allows the systematic testing of SemRep.

3.6 Further Commands

Please use the **help** command to obtain a complete overview about all commands resp. features provided by SemRep. For example, SemRep also provides statistics to a specific concept and to the imported resources in general. It also allows to build word lists based upon the imported resources or to execute mappings (for each correspondence, a relation type will be calculated and added to the correspondence).

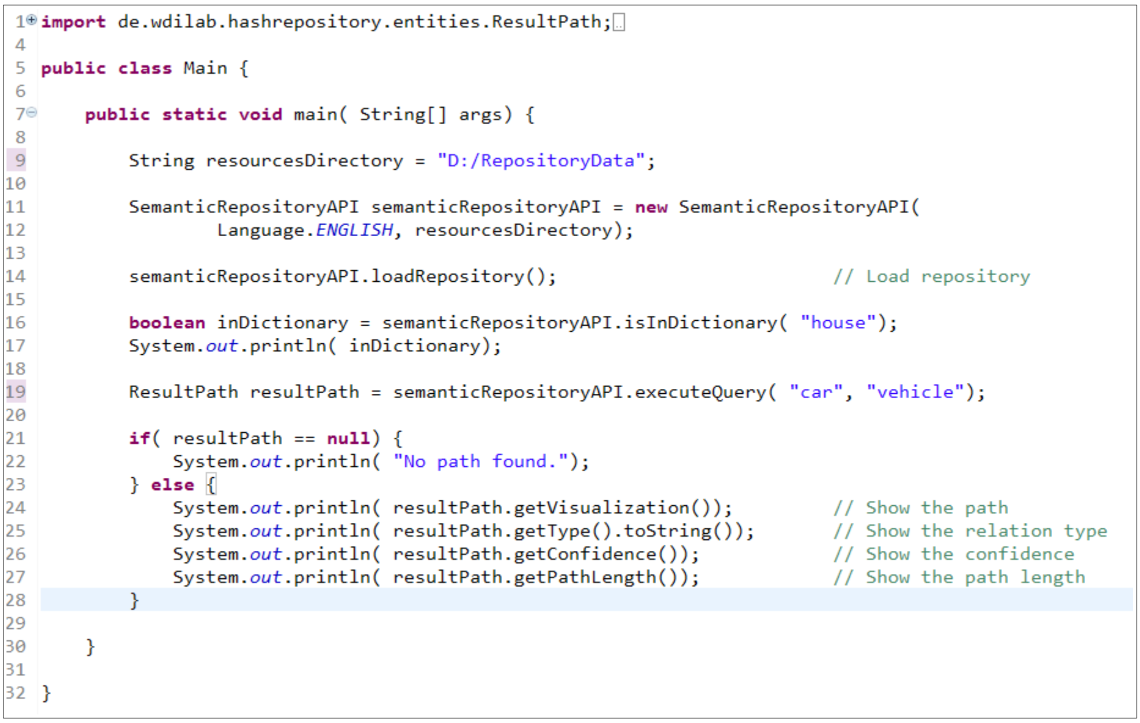
4 Java API

4.1 Basic Usage

SemRep also provides an API to be used in other software projects. Basically, the Java library SemRep.jar contains the entire functionality of SemRep, except for the resource files which have to be stored in a separate directory.

The interface of SemRep is **SemanticRepositoryAPI.java**. This class provides access to some core functionality of SemRep, namely, to check whether a given concept is contained in SemRep and second, to determine the relation type between two given input concepts.

In the following, a minimal example is given how to use SemRep in another project. As matter of fact, the library has to be included in the Java build path. Then, an instance of SemanticRepositoryAPI.java has to be created. The constructor obtains two parameters: A language object (English or German) and the absolute file path of the resources directory (Line 11 – 12). Subsequently, the repository has to be loaded, which may take about a minute or two (Line 14). Note that the loading time increases considerably if auto-stemming is used.



```java
 1 import de.wdilab.hashrepository.entities.ResultPath;
 4
 5 public class Main {
 6
 7     public static void main( String[] args) {
 8
 9         String resourcesDirectory = "D:/RepositoryData";
10
11         SemanticRepositoryAPI semanticRepositoryAPI = new SemanticRepositoryAPI(
12                 Language.ENGLISH, resourcesDirectory);
13
14         semanticRepositoryAPI.loadRepository();                    // Load repository
15
16         boolean inDictionary = semanticRepositoryAPI.isInDictionary( "house");
17         System.out.println( inDictionary);
18
19         ResultPath resultPath = semanticRepositoryAPI.executeQuery( "car", "vehicle");
20
21         if( resultPath == null) {
22             System.out.println( "No path found.");
23         } else {
24             System.out.println( resultPath.getVisualization());        // Show the path
25             System.out.println( resultPath.getType().toString());      // Show the relation type
26             System.out.println( resultPath.getConfidence());           // Show the confidence
27             System.out.println( resultPath.getPathLength());           // Show the path length
28         }
29
30     }
31
32 }
```

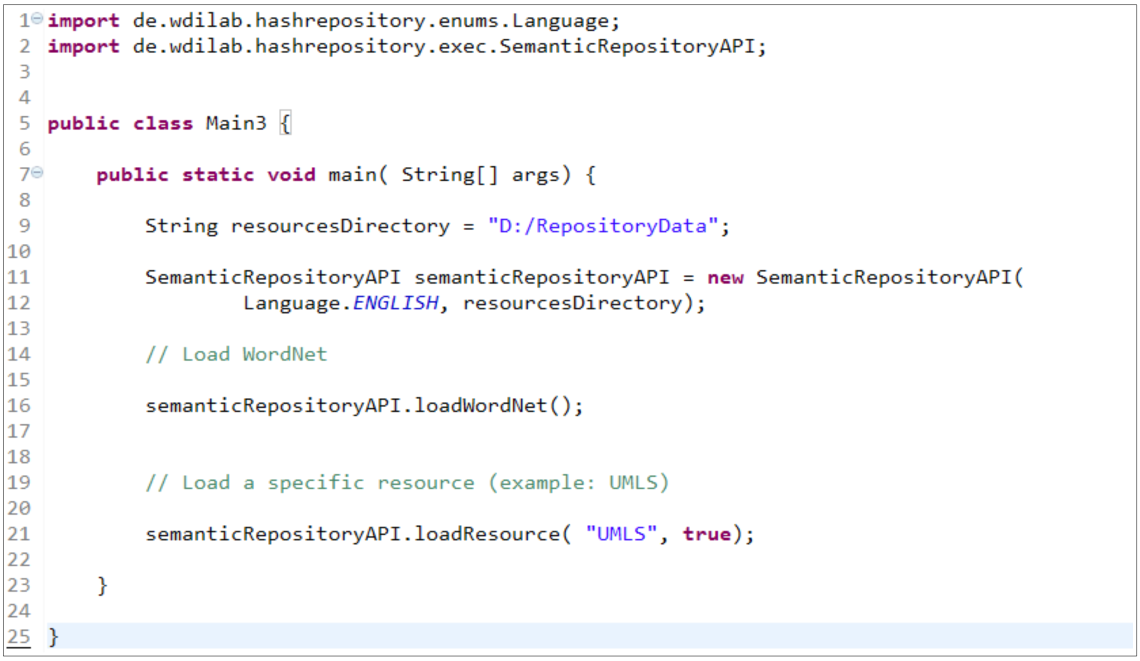
The API can then be used to check whether a given concept like “house” is contained in the repository (Line 16 – 17). Additionally, it can be used to find the most relevant path between two concepts (Line 19). If no path can be found, this method will return null (Line 21 – 23). Otherwise, a visualization of the path can be obtained (Line 24), the relation type holding between the two concepts (Line 25), the confidence score of the path (Line 26) and the path length (Line 27).

CAUTION!

When using the API you need to grant the JVM enough memory both for your application and for SemRep.

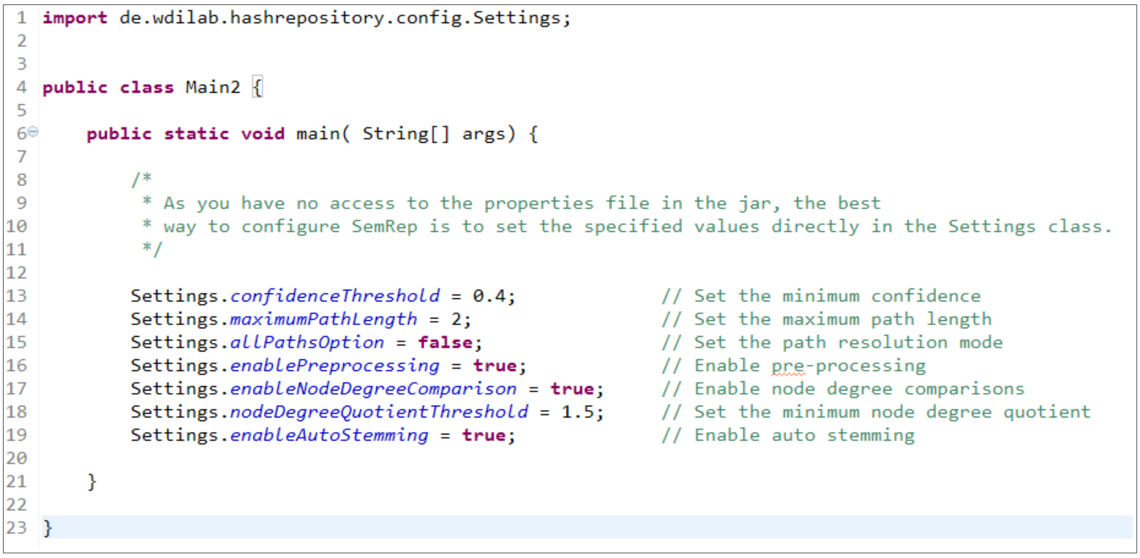
4.2 Advanced Usage

The SemRep API allows you to load specific resources as the following sample shows. You can load WordNet as a single resource (line 16) or you can load any other resource from the Repository Data Set. For example, in line 21 the UMLS resource is loaded. Please use the console-based interface to obtain a list of all available resources and their name id (enter *show resources*).



```
 1 import de.wdilab.hashrepository.enums.Language;
 2 import de.wdilab.hashrepository.exec.SemanticRepositoryAPI;
 3
 4
 5 public class Main3 {
 6
 7     public static void main( String[] args) {
 8
 9         String resourcesDirectory = "D:/RepositoryData";
10
11         SemanticRepositoryAPI semanticRepositoryAPI = new SemanticRepositoryAPI(
12                 Language.ENGLISH, resourcesDirectory);
13
14         // Load WordNet
15
16         semanticRepositoryAPI.loadWordNet();
17
18
19         // Load a specific resource (example: UMLS)
20
21         semanticRepositoryAPI.loadResource( "UMLS", true);
22
23     }
24
25 }
```

In order to change the settings, you may directly access Settings.java, where some basic settings are defined as simple static fields. The following sample shows the access to several important configuration parameters.



```
import de.wdilab.hashrepository.config.Settings;


public class Main2 {

    public static void main( String[] args) {

        /*
         * As you have no access to the properties file in the jar, the best
         * way to configure SemRep is to set the specified values directly in the Settings class.
         */

        Settings.confidenceThreshold = 0.4;             // Set the minimum confidence
        Settings.maximumPathLength = 2;                 // Set the maximum path length
        Settings.allPathsOption = false;                // Set the path resolution mode
        Settings.enablePreprocessing = true;            // Enable pre-processing
        Settings.enableNodeDegreeComparison = true;     // Enable node degree comparisons
        Settings.nodeDegreeQuotientThreshold = 1.5;     // Set the minimum node degree quotient
        Settings.enableAutoStemming = true;             // Enable auto stemming

    }

}
```

4.3 Using SemRep in the Context of Schema or Ontology Matching

SemRep may also be used for (small) match tasks, though it comes with several obstacles:

- The query execution time exceeds classic match strategies (like trigram) by far. Depending on the concepts used and resources loaded, best-case execution times are still in the milliseconds and may become as long as several 100 ms.
- To some degree, SemRep is based upon the assumption that input words are (somehow) semantically related. Asking SemRep about the semantic type between two random concepts (like *bird* and *car*) may occasionally lead to very peculiar results.
- Some heuristics are no longer useful if SemRep is used for matching. For instance, the node degree quotient becomes futile when relatedness cannot be granted in the first place.

Recommendations for SemRep configuration in match scenarios:

- Only load resources that seem necessary for your match task. For instance, in the biodomain Wikipedia and UMLS may suffice.
- Never use a maximum path length larger than 2!
- Run SemRep in first-path-mode.
- Do not use node degree comparison!
- Consider to turn off preprocessing entirely (less recall, more precision).

- Do not pass itemizations like “Cars and Motorbikes” to SemRep. Resolving such complex expression can only be conducted by STROMA.
- Possibly, use SemRep only for non-obvious (or “critical”) correspondences. For instance, it is unnecessary to pass two identical strings to SemRep.
- Clean strings before you send them to SemRep (e.g., remove underscores and the like).
- Possibly, check the relation type that is returned. You may consider some types such as PART-OF or RELATED as a non-match or unlikely match. For example, a *CPU* is part of a *computer*, but might be actually matched to *Processor* (EQUAL). Thus, EQUAL should be preferred against IS-A, which should be preferred against PART-OF and RELATED.

5 References and Contact

[1] Patrick Arnold and Erhard Rahm. SemRep: A Repository for Semantic Mapping. 16. Fachtagung “Datenbanksysteme für Business, Technologie und Web”, 2015.

[2] Patrick Arnold and Erhard Rahm. Extracting Semantic Concept Relations from Wikipedia. Proceedings of the 4th International Conference on Web Intelligence, Mining and Semantics (WIMS), 2014.

[3] Patrick Arnold and Erhard Rahm. Automatic Extraction of Semantic Relations from Wikipedia. International Journal on Artificial Intelligence Tools, 24(2), 2015.

All references are available at http://dbs.uni-leipzig.de/person/patrick_arnold

Contact:

Patrick Arnold
Database Department
Leipzig University
October 2015

arnold@informatik.uni-leipzig.de