

Comparing and Benchmarking Semantic Measures Using SMComp*

Teresa Costa¹ and José Paulo Leal²

- 1 CRACS & INESC-Porto LA, Faculty of Sciences, University of Porto, Porto, Portugal
teresa.costa@dcc.fc.up.pt
- 2 CRACS & INESC-Porto LA, Faculty of Sciences, University of Porto, Porto, Portugal
zp@dcc.fc.up.pt

Abstract

The goal of the semantic measures is to compare pairs of concepts, words, sentences or named entities. Their categorization depends on what they measure. If a measure only considers taxonomy relationships is a similarity measure; if it considers *all* type of relationships it is a relatedness measure.

The evaluation process of these measures usually relies on semantic gold standards. These datasets, with several pairs of words with a rating assigned by persons, are used to assess how well a semantic measure performs.

There are a few frameworks that provide tools to compute and analyze several well-known measures. This paper presents a novel tool – SMComp – a testbed designed for path-based semantic measures. At its current state, it is a domain-specific tool using three different versions of WordNet.

SMComp has two views: one to compute semantic measures of a pair of words and another to assess a semantic measure using a dataset. On the first view, it offers several measures described in the literature as well as the possibility of creating a new measure, by introducing Java code snippets on the GUI. The other view offers a large set of semantic benchmarks to use in the assessment process. It also offers the possibility of uploading a custom dataset to be used in the assessment.

1998 ACM Subject Classification E.1 Graphs and networks, I.2.4 Knowledge Representation Formalisms and Methods, Semantic networks, H.3.5 Online Information Services, Web-based services, H.5.3 Group and Organization Interfaces, Web-based interaction

Keywords and phrases Semantic similarity, semantic relatedness, testbed, web application

Digital Object Identifier 10.4230/OASICS.SLATE.2016.4

1 Introduction

Semantic measures are an attempt to quantify and compare pairs of concepts, words, sentences or named entities that can be regarded as a kind of semantic distance in a semantic space [12].

* This work is partially financed by the ERDF – European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation – COMPETE 2020 Programme and by the FCT within project POCI-01-0145-FEDER-006961 and project “NORTE-01-0145-FEDER-000020” financed by the North Portugal Regional Operational Programme (NORTE 2020), under the PORTUGAL 2020 Partnership Agreement and through the European Regional Development Fund (ERDF).



© Teresa Costa and José Paulo Leal;
licensed under Creative Commons License CC-BY

5th Symposium on Languages, Applications and Technologies (SLATE'16).

Editors: Marjan Mernik, José Paulo Leal, and Hugo Gonçalves Oliveira; Article No. 4; pp. 4:1–4:13

Open Access Series in Informatics



OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

This analysis is difficult since the object of semantic measures is inherently psychological, thus their evaluation rely on datasets that average the human perception of the semantic relationships.

There are two main types of semantic measures: similarity and relatedness. Despite being two different concepts they are commonly confused. Similarity measures the amount of common features between a pair of entities. It depends on the size of the smallest class that contains them. Relatedness depends on any relationship type connecting two elements, including but not restricted to class membership and inclusion. For instance, cat and dog are two similar concepts since they are both mammals; the same can be said about ant and flea since they are both insects. An ant and a dog are also similar insofar they are both animals, but less similar than cat and dog. The reason why is that the class *animals* contains both the classes *mammal* and *insect*. Fleas are related to cats and dogs since they parasite them, thus fleas are more related to dogs than ants. This happens not because the features they share, and they share because they are animals, but because of other relationships, in this case parasitism. In such way, the similarity of dogs and fleas may be the same as the similarity of dogs and ants, but the relatedness between fleas and dogs is greater than dogs and ants.

In spite of the clear difference between similarity and relatedness, people often confuse them or value them differently, as the classical example of how people compare a magazine, a pencil and a notepad [9] demonstrates. This difference is particularly noted when the same pair of words is measure using both types: similarity and relatedness.

This paper presents SMComp, a testbed for semantic measures freely available on-line¹. This web application provides several well-known path based semantic measures and also supports user-defined measures. It has also an assessment mode, using several semantic gold standards or a custom dataset, uploaded by the user.

Section 2 surveys different semantic measures, datasets and libraries. It categorizes the types of semantic measures regarding their source, with focus on knowledge-based measures using the structural approach. This section also provides an overview on other tools used to compute and analyze semantic measures. It identifies also the semantic datasets commonly used in the assessment of semantic measures.

Section 3 introduces SMComp, providing an overview of this testbed. It details on its architecture and its behavior. The validation process was performed using the semantic graphs and measures available in the web application. The results are presented in Section 4. Section 5 summarizes the research present in this paper and highlights its main contribution.

2 Background

A semantic measure is a numerical estimation of how semantically connected two elements are. This evaluation relies on the analysis of information extracted from semantic sources. The type of a semantic measure depends on the type of the semantic source. The unstructured / semi-structured sources (plain texts or dictionaries, for instance) are used by Distributional Measures. Structured sources, such as semantic graphs (WordNet or DBPedia, for instance) are used by Knowledge-based Measures. These measures follow three different approaches: the structural [15, 23, 30] approach, the feature-based [4, 25, 28] approach and the Shannon's Information Theoretical [16, 22, 20, 21] approach. There are also hybrid measures [2, 3, 18, 26] that combine distributional and knowledge-based approaches and take advantage of both representations.

¹ <http://quilter.dcc.fc.up.pt/smcomp/>

This paper focus on structural knowledge-based measures, namely on path-based measures. A common knowledge source used on the assessment of these measures is WordNet² [7]. This knowledge base is usually represented as a semantic graph that models the English lexical knowledge. Its structure uses synsets as nodes and they are connected by semantic and lexical relationships.

Path-based measures follow a structural approach. They take advantage of several graph traversal strategies, such as shortest path, random walks and other interaction analysis. These measures are based on the analysis of the interconnections between nodes and use it to estimate the similarity (or relatedness) between them. They rely on the definition of *shortest path* and *least common subsumer*. The measures proposed by Rada et al. [23], Resnik [26], Leacock & Chodorow [15] and Wu & Palmer [30] are examples of path-based similarity measures. Hirst & St-Onge [14] and Strube & Ponzetto [29] proposed several relatedness measures.

There are several tools for the computation and analysis of semantic measures. They can be categorized according to the interface they support. The programming packages are usually general libraries, not focused in any particular semantic graph. They can be used only through their application interface (API) which requires programming just to be used. SML³ [11] is an example of this kind of tool. It is simultaneously a library and a toolkit designed to the computation of semantic measures, such as similarity, relatedness and distance, supporting a large range of knowledge-based measures.

User packages provide *both* API and a graphic user interface (GUI), usually deployed on the web. Typically, they are domain-specific since they are designed for a particular semantic graph. WordNet::Similarity [19] is a Perl module that implements several semantic similarity and relatedness measures. The results are based on the information found in a specific version of WordNet. WordNet::Similarity is freely available for download and has also a web interface⁴. WS4J⁵ is a Java API for several semantic similarity and relatedness measures. It is freely available for download and has also a web application⁶ where one can compute simultaneously all the semantic measures available between a pair of concepts also in a specific version of WordNet. None of these tools provide measure assessment from the GUI.

The assessment of semantic measures usually rely on the correlation of a measure with expected scores of word pairs. This provides an insight of how well they mimic the human capacity to compare things. Most of datasets used in the measure assessment average human ratings for a set of word pairs [12]. The ratings are collected during experiments involving several participants. According to the instructions provided to them, they assign a similarity or relatedness score to the word pairs. Usually, each dataset has its own set of instructions which may influence the notions of similarity and relatedness. Thus, datasets are also categorized as intended for similarity or relatedness. This work uses four different similarity datasets [27, 17, 1, 13] and five relatedness datasets [8, 1, 5, 10, 24].

3 SMComp

SMComp is a testbed for semantic measures, developed with Google Web Toolkit (GWT), that is freely available on-line⁷. It presents a novel approach that combines the GUI and

² <https://wordnet.princeton.edu/>

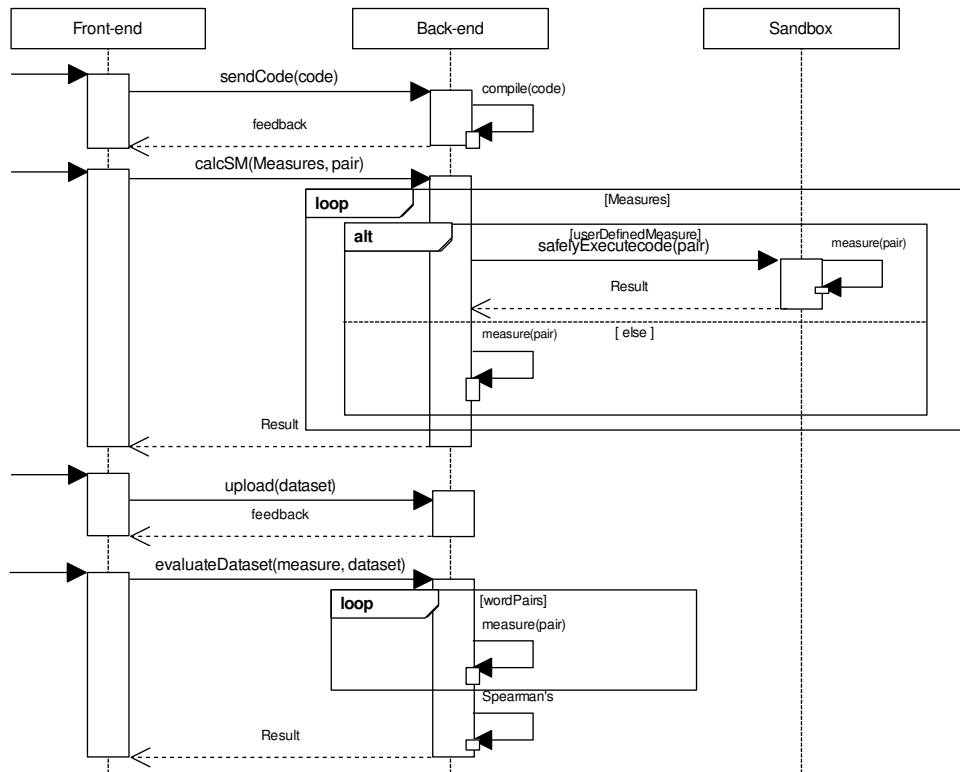
³ <http://www.semantic-measures-library.org/sml/index.php>

⁴ <http://marimba.d.umn.edu/cgi-bin/similarity/similarity.cgi>

⁵ <https://code.google.com/archive/p/ws4j/>

⁶ <http://ws4jdemo.appspot.com/>

⁷ <http://quilter.dcc.fc.up.pt/smcomp/>



■ **Figure 1** SMComp sequence diagram.

API features. Using its web interface the user can test a range of semantic measures and assess them with several datasets referred in the literature. This GUI also allows the user to provide a snippet of Java code to implement his own semantic measure and quickly assess its performance with relevant benchmarks.

The presented web application has four main distinctive features:

- it compares a semantic similarity measure with its corresponding relatedness version;
- it allows the user to create his own semantic measure and test it with the available semantic sources;
- it allows the user to upload a custom dataset;
- it assesses semantic measures using a set of semantic datasets.

The following Subsections detail on each aspect of SMCom: architecture, front-end, back-end and sandbox.

3.1 Architecture

SMComp is a testbed composed of 3 main components: a *front-end*, running on a web client, responsible for user interaction; a *back-end* running on a server responsible for computing semantic measures; and a *sandbox* responsible for the safe execution of user-provided Java code.

Figure 1 shows an UML sequence diagram summarizing the interaction between components. It has 4 key moments: submit a user-defined method, compute semantic measures, upload a custom dataset and assess a measure. On the first moment, the Java code is sent to the back-end server to be compiled. On the second moment, a selection of measures is sent

for computation. The back-end server iterates over the set of measures. If it includes the user-defined measure, it is sent to the sandbox to be safely executed. On the third moment, the user uploads his dataset to be used in the evaluation process. This process occurs on the final moment, where a single measure is assessed using a selected dataset.

3.2 Back-end

Semantic Graph Processing

The implemented semantic measures follow a structural approach, depending on graph traversal strategies to find paths connecting two words. This strategy can be a very time-consuming process, in particular if a remote source is used. To rely on remote sources would raise some issues: there is no guarantees that they are always available and network performance issues may hinder the execution of a semantic measures, since it requires a massive amount of graph queries.

In order to overcome these constraints, this application relies on a pre-processed graph. Known knowledge-bases, such as WordNet, usually provide dumps of their data. These dumps, in Resource Description Framework (RDF), were previously pre-processed in order to store the graph locally.

The graph pre-processing requires to parse RDF data. This data can be retrieved in several formats, such as Turtle, RDF/XML or N-Triples. To simplify and unify the process, all dumps are converted to N-Triples, the simplest RDF serialization. This conversion relies on the Apache Jena Framework⁸.

The SMComp measures are path-based. Thus, the internal representation of semantic graphs in SMComp is designed to simplify the manipulation of paths. A path can be seen as a sort of chain, composed by a sequence of links. Hence, a pre-processed graph in SMComp can be seen as a collection of links ready to form chains. These chain links are called **Transition** and are made of a pair of **Node**. This internal representation is created by loading semantic graph serializations.

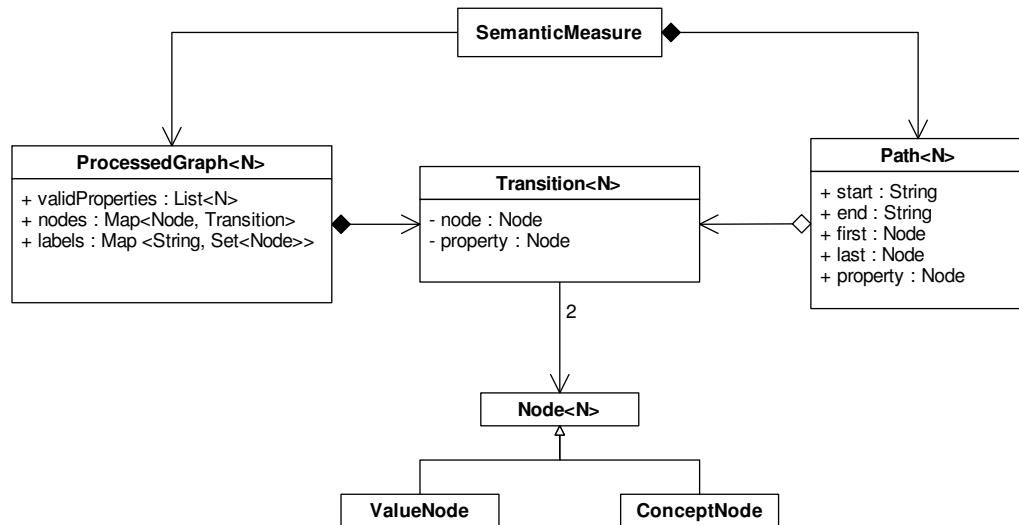
A semantic graph in RDF is a collection of triples of the form *subject - predicate - object*. This data is represented in SMComp using the types depicted in the UML class diagram in Figure 2. Each element of a triple is converted to a **Node<N>**, where the type parameter N is the type of the node representation, such as **String** or **Jena Resource**. A *subject* must be a **Concept Node** and the same is true for the *predicate*. The *object* can be either a **Value Node** or a **Concept Node**. **Concept Node** are serialized in RDF by URIs and **Value Node** are serialized as literals. For each *subject*, a **Transition** is created and associated to it, using the values of the *predicate* and *object*.

A **Path** is a more complex data type that has a node at its first position followed by a sequence of transitions. Paths are create by semantic measures using the **ProcessedGraph**.

This mapping process creates a locally stored graph, called **ProcessedGraph**. Figure 2 shows its operations and the relationships with the other types. The use of the **ProcessedGraph** eases the process of adding and retrieving path components, i.e. **Transition** instances.

SMcomp uses the **ProcessedGraph** format to store the available knowledge bases. This conversion process is executed only once, when a new semantic graph is added to SMComp.

⁸ <https://jena.apache.org/>



■ **Figure 2** UML class diagram of the ProcessedGraph.

The RDF dumps used are available for WordNet 2.1⁹, WordNet 3.0¹⁰ and WordNet 3.1¹¹. WordNet 2.1 uses 4.8GB of disk space, WordNet 3.0 uses 6.6GB of disk space and WordNet 3.1 uses 6.3GB.

Semantic Measures

SMcomp implements several path-based semantic measures, namely those described in Section 2. With the exception of the Hirst & St-Onge, the proposed measures were originally designed to estimate semantic similarity. However, they were adapted by Strube & Ponzetto to also measure semantic relatedness.

In addition to these measures, SMComp also implements a Resnik and a Hirst & St-Onge adaptation, enabling measure relatedness with the former and similarity with the later. The approach followed in these adaptations is similar to that used by Strube & Ponzetto [29]. To compute semantic relatedness using the Resnik measure, all the properties must be considered instead of only the taxonomic ones. To compute semantic similarity with Hirst & St-Onge, one must only consider the only taxonomic properties instead of the *allowable*¹² ones.

The implementation process considered the following assumptions:

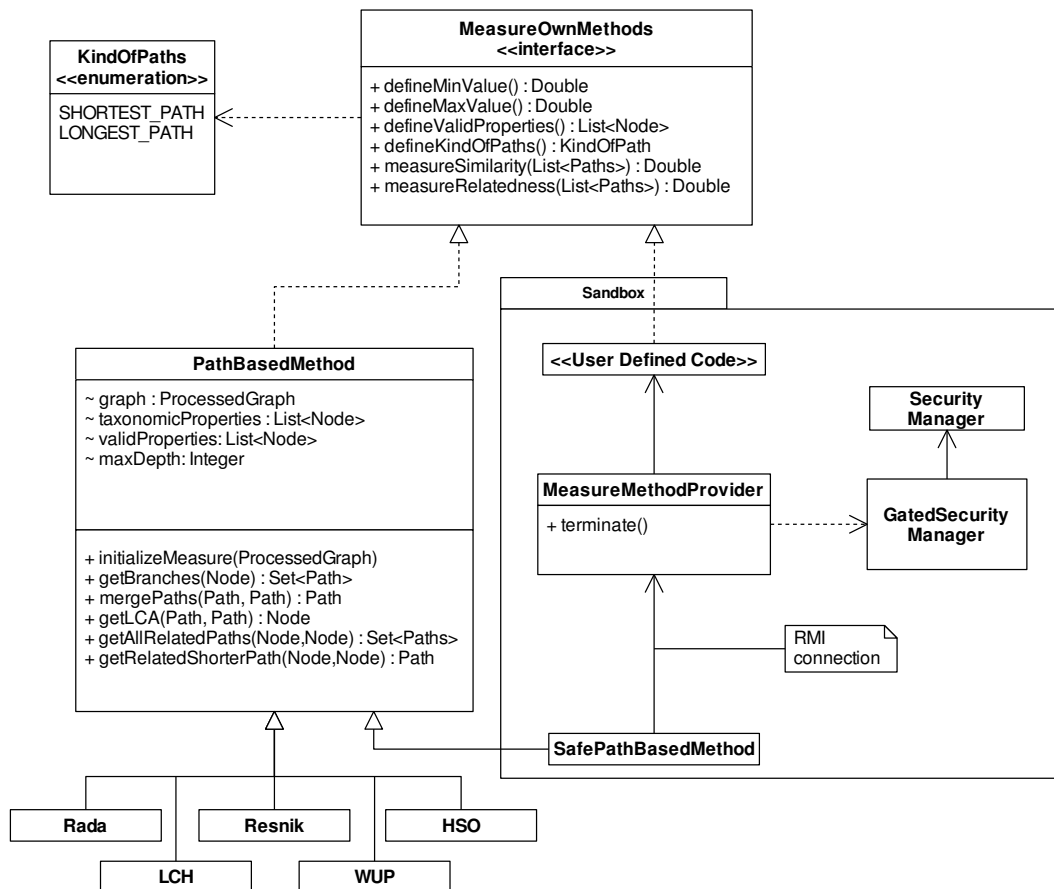
- the value of a semantic measure between a word and itself it is the maximum value;
- if at least one of the words is not in the semantic graph the value of its measure is the minimum value;
- if the semantic graph has several roots, a new node is inserted to create a semantic tree with a single root node;

⁹ <http://sourceforge.net/projects/texai>

¹⁰ <http://semanticweb.cs.vu.nl/lod/wn30/>

¹¹ <http://wordnet-rdf.princeton.edu/>

¹² The *allowable* relations include: *see also*, *antonymy*, *attribute*, *cause*, *entailment*, *holonymy*, *hypernymy*, *hyponymy*, *meronymy*, *pertinence* and *similarity*.



■ **Figure 3** UML class diagram of the path-based semantic measures.

- the disambiguation strategy selects the pair of concepts (derived from the input words) that produces the best measure.

Figure 3 is an UML class diagram for the available methods. An important advantage of the SMComp design is the abstraction of common features used by all measures, exploiting the fact they are all path-based. This simplified the implementation of the literature methods and provided a skeleton to easily implement new measures with only a few lines of code.

All methods follow the same approach. The first step is to define the range of values, which is done by `defineMinValue()` and `defineMaxValue()`. If the measure considers a specific set of properties when computing relatedness, those must be declared using `defineValidProperties()`. A method can consider either the shortest path or a path of any size. To define which kind should be considered the `defineKindOfPaths()` is used. At last, the methods that actually implement the measures are `measureSimilarity` and `measureRelatedness`. The methods receive a list of `Path` and return a numerical value.

3.3 Sandbox

The Java code submitted by the user is compiled by back-end. Compilation errors detected at this stage are reported back to the client and shown on the user interface. For safety

reasons, the execution of this code is performed in a different Java Virtual Machine (JVM). Thus, the back-end can ensure termination of the submitted code.

The sandbox is the component responsible for the execution of submitted code. This component uses its own **Security Manager** to deny the access of user submitted code to any sensitive resources, such as the file system or the network, avoiding the execution of malicious code. It also enforces time limit on the execution of individual methods.

The communication between the back-end and the sandbox relies on Remote Method Invocation (RMI). The sandbox exposes all the methods defined by **MeasureOwnMethods** interface and also the `terminate()` method, which is invoked by the back-end whenever the `user-submitted code` can no longer be executed. The sandbox is also responsible for reporting back to the back-end, and are forward to the front-end, any timeout, execution errors or security violations.

3.4 Front-end

Figure 4 presents a screenshot with an edit view of the SMComp front-end. The web interface is a single window and requires no authentication. An user can select one of two modules: word pairs or datasets. This selection is available by using the tabs on the top of the window.

Both tabs have a similar layout. The major difference is in the area right bellow the tabs: in the word pairs view it has entries for the words to compare; in the datasets view it has a selector with the available benchmarks. The selector of knowledge source is also similar in both tabs. They also have similar grids where the computed results are displayed and a large button labelled “Compute” at the bottom.

All computation results are displayed in a two column grid. Each row corresponds to a semantic measure and the columns to the semantic type (left column is assigned to relatedness and the right column is assigned to similarity). The row and column headers have check boxes where the user selects which measures and variants he wants to compute. Bellow each row label, there is a button with a magnifying glass. When pressed, it shows a dialog box with the Java implementation of the measure. This should be useful as an example for the user implementation.

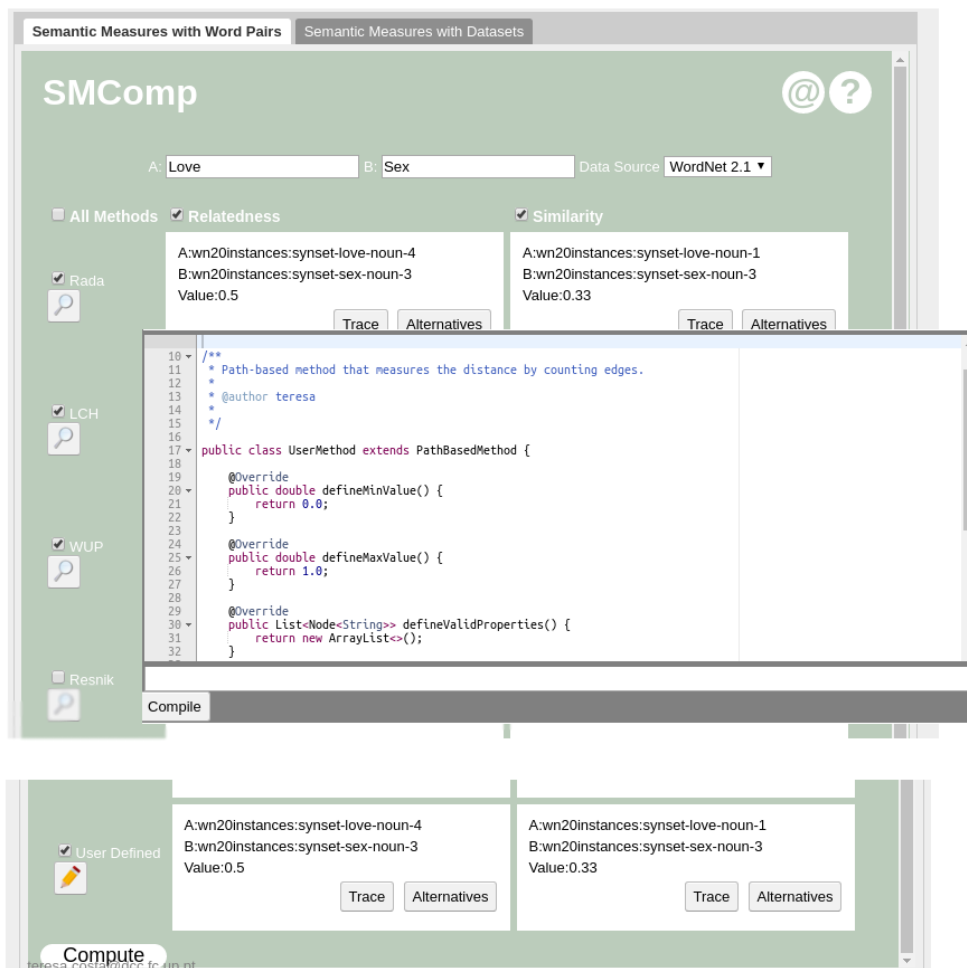
Bellow the *User Defined* label, the button icon shows a pencil. When pressed, the user enters on edit mode of its measure, as shown on Figure 4. This dialog provides a code editor, where he can enter a Java class with a new method, extending **PathBaseMethod**.

The assessment view using datasets is similar. The computation results are also displayed in a two column grid where each row corresponds to a semantic measure and each column to a semantic variant. The measure and its variant are selected using radio buttons. The available datasets are displayed in a list on the top. It is possible to upload a custom dataset by selecting *USER* from this list. In this case, a dialog box opens where values can be inserted using the format `word, word, value`. Clicking in the button labelled “Upload” dataset is sent to server and is ready to use in the assessment.

Use-case example

The code in Listing 1 is a simple example of the implementation of a user defined measure. The first step is to give a name to the class, which in this case is **UserDefinedMethodExample**. This class implements **MeasureOwnMethods** that provides a skeleton of the measure to complete.

The minimum and maximum numerical value of the measure must be defined. In this example, the minimum value is set to 0 and the maximum is set to the largest double. These



■ **Figure 4** SMComp user interface.

were defined with `defineMinValue()` and `defineMaxValue()` respectively. The next step is to decide which properties should be considered when measuring relatedness since in similarity only *is-a* relationships are considered. If *all* should be considered, `defineValidProperties()` must return a empty list, which is the case.

It is also needed to define the size of paths. One can select between the shortest path between two elements or the longest path. This example uses the shortest path which is the enumerate value defined by `defineKindOfPaths()`.

After these initialization the user can define the measures itself. In this case, the measure takes in consideration the number of edges in the shortest path. The score is given by $1/(\text{number_of_edges_in_the_path} + 1)$.

4 Validation

This work is part of a comprehensive research on semantic measures that aimed at understanding the misconceptions between similarity and relatedness. SMComp was developed to support its validation process. The detailed results obtained are available on [6].

■ **Listing 1** User defined method example.

```
import pt.up.fc.dcc.SemArachne.datagraphTypes.*;
import pt.up.fc.dcc.SemArachne.methods.MeasureOwnMethods;
import pt.up.fc.dcc.SemArachne.methods.KindOfPaths;
import java.util.*;

public class UserDefinedMethodExample implements MeasureOwnMethods {
    @Override
    public double defineMinValue() {
        return 0.0;
    }

    @Override
    public double defineMaxValue() {
        return Double.MAX_VALUE;
    }

    @Override
    public List<Node<String>> defineValidProperties() {
        return new ArrayList<>();
    }

    @Override
    public KindOfPaths defineKindOfPaths() {
        return KindOfPaths.SHORTEST_PATH;
    }

    @Override
    public double measureSimilarity(List<Path<Node<String>>> paths) {
        int edges = paths.get(0).getProperties().size();
        return score = 1.0 / ((double) edges + 1);
    }

    @Override
    public double measureRelatedness(List<Path<Node<String>>> paths) {
        int edges = paths.get(0).getProperties().size();
        return score = 1.0 / ((double) edges + 1);
    }
}
```

The validation performed executed a cross evaluation using 10 different semantic measures (5 similarity and 5 relatedness) and 9 semantic datasets (4 similarity and 5 relatedness). The three versions of WordNet were used in the process.

Using WordNet 2.1, WUP and HSO similarity measures stood out since they had a better performance with similarity datasets. WS Sim dataset was correctly identified by all measures and MTurk-287 and MEN was always misidentified. With WordNet 3.0, WUP and HSO similarity measures stood out again. WS Sim was once again always identified as a similarity dataset. WordNet 3.1 only WUP stood out for similarity and HSO for relatedness. WS Rel was correctly identified by all semantic measures.

5 Conclusion

This work presents a testbed to compare, analyze and assess path-based semantic measures. SMComp is a novel approach for tools of this kind that couples GUI and API in a web interface. It provides three versions of a widely used semantic knowledge-base (WordNet); implementations of the most referenced path-based measures described in the literature; support for user-provided measures, coded with small snippets on the web interface; and evaluation of semantic measure quality using standard and custom datasets.

A validation of SMComp was performed during an experiment to compare similarity and relatedness measures and datasets. This experiment covered all the measures and datasets supported by SMComp. Using SMComp we were able to show that most of the semantic measures do not have the best performance with datasets of their type. It also allowed to pinpoint some measures and datasets more accurate, namely Wu & Palmer similarity measure and WS-Sim and WS-Rel datasets.

References

- 1 Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paçca, and Aitor Soroa. A study on similarity and relatedness using distributional and wordnet-based approaches. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL'09, pages 19–27, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. URL: <http://dl.acm.org/citation.cfm?id=1620754.1620758>.
- 2 Satanjeev Banerjee and Ted Pedersen. An adapted lesk algorithm for word sense disambiguation using wordnet. In *Computational linguistics and intelligent text processing*, pages 136–145. Springer, 2002.
- 3 Satanjeev Banerjee and Ted Pedersen. Extended gloss overlaps as a measure of semantic relatedness. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, IJCAI'03, pages 805–810, San Francisco, CA, USA, 2003. Morgan Kaufmann Publishers Inc. URL: <http://dl.acm.org/citation.cfm?id=1630659.1630775>.
- 4 Olivier Bodenreider, Marc Aubry, and Anita Burgun. Non-lexical approaches to identifying associative relations in the gene ontology. In *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, pages 91–102. NIH Public Access, 2005.
- 5 Elia Bruni, Nam Khanh Tran, and Marco Baroni. Multimodal distributional semantics. *J. Artif. Int. Res.*, 49(1):1–47, January 2014. URL: <http://dl.acm.org/citation.cfm?id=2655713.2655714>.
- 6 Teresa Costa and José Paulo Leal. Semantic measures: How similar? How related? in print, 2016.
- 7 Christiane Fellbaum. *WordNet*. Wiley Online Library, 1999.
- 8 Lev Finkelstein, Evgeniy Gabilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppín. Placing search in context: The concept revisited. In *Proceedings of the 10th international conference on World Wide Web*, pages 406–414. ACM, 2001.
- 9 Yuriy Gorodnichenko and Gerard Roland. Understanding the individualism-collectivism cleavage and its effects: Lessons from cultural psychology. In Masahiko Aoki and Timur Kuran, editors, *Institutions and Comparative Economic Development*, volume 150, page 213. Palgrave Macmillan, 2012.
- 10 Guy Halawi, Gideon Dror, Evgeniy Gabilovich, and Yehuda Koren. Large-scale learning of word relatedness with constraints. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD'12, pages 1406–1414, New York, NY, USA, 2012. ACM. doi:10.1145/2339530.2339751.

- 11 Sébastien Harispe, Sylvie Ranwez, Stefan Janaqi, and Jacky Montmain. The semantic measures library and toolkit: fast computation of semantic similarity and relatedness using biomedical ontologies. *Bioinformatics*, 30(5):740–742, 2014.
- 12 Sébastien Harispe, Sylvie Ranwez, Stefan Janaqi, and Jacky Montmain. Semantic similarity from natural language and ontology analysis. *Synthesis Lectures on Human Language Technologies*, 8(1):1–254, 2015.
- 13 Felix Hill, Roi Reichart, and Anna Korhonen. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *arXiv preprint arXiv:1408.3456*, 2014.
- 14 Graeme Hirst and David St-Onge. Lexical chains as representations of context for the detection and correction of malapropisms. *WordNet: An electronic lexical database*, 305:305–332, 1998.
- 15 Claudia Leacock and Martin Chodorow. Combining local context and wordnet similarity for word sense identification. *WordNet: An electronic lexical database*, 49(2):265–283, 1998.
- 16 Dekang Lin. An information-theoretic definition of similarity. In *Proceedings of the Fifteenth International Conference on Machine Learning, ICML’98*, pages 296–304, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc. URL: <http://dl.acm.org/citation.cfm?id=645527.657297>.
- 17 George A. Miller and Walter G. Charles. Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6(1):1–28, 1991. doi:10.1080/01690969108406936.
- 18 Siddharth Patwardhan, Satanjeev Banerjee, and Ted Pedersen. Using measures of semantic relatedness for word sense disambiguation. In *Proceedings of the 4th International Conference on Computational Linguistics and Intelligent Text Processing, CICLing’03*, pages 241–257. Springer-Verlag, Berlin, Heidelberg, 2003. URL: <http://dl.acm.org/citation.cfm?id=1791562.1791592>.
- 19 Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. Wordnet:: Similarity: measuring the relatedness of concepts. In *Demonstration papers at HLT-NAACL 2004*, pages 38–41. Association for Computational Linguistics, 2004.
- 20 Giuseppe Pirró. A semantic similarity metric combining features and intrinsic information content. *Data & Knowledge Engineering*, 68(11):1289–1308, 2009.
- 21 Giuseppe Pirró and Jérôme Euzenat. A feature and information theoretic framework for semantic similarity and relatedness. In *The Semantic Web-ISWC 2010*, pages 615–630. Springer, 2010.
- 22 Giuseppe Pirró and Nuno Seco. Design, implementation and evaluation of a new semantic similarity metric combining features and intrinsic information content. In *On the Move to Meaningful Internet Systems: OTM 2008*, pages 1271–1288. Springer, 2008.
- 23 Roy Rada, Hafedh Mili, Ellen Bicknell, and Maria Blettner. Development and application of a metric on semantic nets. *Systems, Man and Cybernetics, IEEE Transactions on*, 19(1):17–30, 1989.
- 24 Kira Radinsky, Eugene Agichtein, Evgeniy Gabrilovich, and Shaul Markovitch. A word at a time: Computing word relatedness using temporal semantic analysis. In *Proceedings of the 20th International Conference on World Wide Web, WWW’11*, pages 337–346, New York, NY, USA, 2011. ACM. doi:10.1145/1963405.1963455.
- 25 Sylvie Ranwez, Vincent Ranwez, Jean Villerd, and Michel Crampes. Ontological distance measures for information visualisation on conceptual maps. In *On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops*, pages 1050–1061. Springer, 2006.
- 26 Philip Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence – Volume 1, IJCAI’95*, pages 448–453, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc. URL: <http://dl.acm.org/citation.cfm?id=1625855.1625914>.

- 27 Herbert Rubenstein and John B. Goodenough. Contextual correlates of synonymy. *Commun. ACM*, 8(10):627–633, 1965.
- 28 Nenad Stojanovic, Alexander Maedche, Steffen Staab, Rudi Studer, and York Sure. Seal: A framework for developing semantic portals. In *Proceedings of the 1st International Conference on Knowledge Capture*, K-CAP’01, pages 155–162, New York, NY, USA, 2001. ACM. doi:10.1145/500737.500762.
- 29 Michael Strube and Simone Paolo Ponzetto. Wikirelate! Computing semantic relatedness using Wikipedia. In *Proceedings of the 21st National Conference on Artificial Intelligence – Volume 2*, AAAI’06, pages 1419–1424. AAAI Press, 2006. URL: <http://dl.acm.org/citation.cfm?id=1597348.1597414>.
- 30 Zhibiao Wu and Martha Palmer. Verbs semantics and lexical selection. In *Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics*, pages 133–138. Association for Computational Linguistics, 1994.