Information Systems 96 (2021) 101636

Contents lists available at ScienceDirect

Information Systems

journal homepage: www.elsevier.com/locate/is

A large reproducible benchmark of ontology-based methods and word embeddings for word similarity

Juan J. Lastra-Díaz [a,*], Josu Goikoetxea [b], Mohamed Ali Hadj Taieb [c], Ana Garcia-Serrano [a], Mohamed Ben Aouicha [c], Eneko Agirre [b], David Sánchez [d,**]

[a] NLP & IR Research Group, ETSI de Informática (UNED), Universidad Nacional de Educación a Distancia, Juan del Rosal 16, 28040 Madrid, Spain
[b] IXA NLP group, Faculty of Informatics, UPV/EHU Manuel Lardizabal 1 (20018), Donostia, Basque Country, Spain
[c] Faculty of Sciences of Sfax, Tunisia
[d] Department of Computer Engineering and Mathematics, Universitat Rovira i Virgili, Spain

ARTICLE INFO

Article history:
Received 1 November 2019
Received in revised form 3 September 2020
Accepted 5 September 2020
Available online 30 September 2020
Recommended by Fernando Chirigati

Keywords:
Ontology-based semantic similarity measures
Word embeddings
Information Content models
Reproducible benchmark
HESML
Reprozip

ABSTRACT

This work is a companion reproducibility paper of the experiments and results reported in Lastra-Diaz et al. (2019a), which is based on the evaluation of a companion reproducibility dataset with the HESML V1R4 library and the long-term reproducibility tool called Reprozip. Human similarity and relatedness judgements between concepts underlie most of cognitive capabilities, such as categorization, memory, decision-making and reasoning. For this reason, the research on methods for the estimation of the degree of similarity and relatedness between words and concepts has received a lot of attention in the fields of artificial intelligence and cognitive sciences. However, despite the huge research effort done, there is a lack of a self-contained, reproducible and extensible collection of benchmarks which being amenable to become a de facto standard for large scale experimentation in this line of research. In order to bridge this reproducibility gap, this work introduces a set of reproducible experiments on word similarity and relatedness by providing a detailed reproducibility protocol together with a set of software tools and a self-contained reproducibility dataset, which allow that all experiments and results in our aforementioned work to be reproduced exactly. Our aforementioned primary work introduces the largest, most detailed and reproducible experimental survey on word similarity and relatedness reported in the literature, which is based on the implementation of all evaluated methods into the same software platform. Our reproducible experiments evaluate most of methods in the families of ontology-based semantic similarity measures and word embedding models. We also detail how to extend our experiments to evaluate other unconsidered experimental setups. Finally, we provide a corrigendum for a mismatch in the MC28 similarity scores used in our original experiments.

1. Introduction

Human similarity and relatedness judgements between concepts underlie most of cognitive capabilities, such as categorization, memory, decision-making and reasoning. Thus, the proposal of methods for the estimation of the degree of similarity and relatedness between words and concepts has been a very active line of research in the fields of Artificial Intelligence (AI), Natural Language Processing (NLP) and Information Retrieval (IR). For this reason, the authors of this work have been largely involved in this line of research during the last decade by proposing new ontology-based semantic similarity measures [1–8], Information Content (IC) models [8–12], word embeddings [13–16], distributional semantics measures [17–19], semantic measure libraries [20,21], reproducibility resources [22,23], word similarity benchmarks [18], reproducible experiments on word similarity based on WordNet [24,25], and reproducible benchmarks between semantic measures libraries [24,26].

Semantic similarity is defined like the degree of resemblance between two concepts, whilst semantic relatedness is defined like the degree of relatedness by considering any kind of relationship linking them. For instance, the concepts *animal* and *zoo* have a low degree of similarity but a high degree of relatedness. Semantic similarity measures only consider 'is-a' relationships between concepts, whilst semantic relatedness measures consider a wide range of relationships, such as hypernymy, hyponymy,

DOI of original article: https://doi.org/10.1016/j.dib.2019.104432.
* Corresponding author.
** Reviewer.
E-mail addresses: jlastra@invi.uned.es (J.J. Lastra-Díaz), josu.goikoetxea@ehu.eus (J. Goikoetxea), mohamedali.hadjtaieb@gmail.com (M.A. Hadj Taieb), agarcia@lsi.uned.es (A. Garcia-Serrano), mohamed.benaouicha@fss.usf.tn (M. Ben Aouicha), e.agirre@ehu.eus (E. Agirre), david.sanchez@urv.cat (D. Sánchez).

https://doi.org/10.1016/j.is.2020.101636
0306-4379/

meronymy, antonymy, synonymy, as well as other relationships which are manifested by some form of co-occurrence of words. The main approaches on word similarity and relatedness proposed in the literature can be categorized in two large families as follows: (1) Ontology-based semantic similarity Measures (OM), and (2) distributional measures, whose most recent and successful methods are based on Word Embedding (WE) models. Ontology-based semantic similarity Measures are mainly focused on word similarity, whilst word embeddings mainly focus on word relatedness. However, recent state-of-the-art word embeddings achieve state-of-the-art results on word similarity by integrating ontologies in their models, as shown in our primary work [27].

Our primary work [27] introduces a comprehensive experimental study on the main aforementioned families of methods on word similarity and relatedness, which is based on the implementation and evaluation of all methods in a same software platform based on HESML V1R4 [28] and WordNet 3.0 [29]. Likewise, all experiments reported in our primary work [27] were recorded with the Reprozip long-term reproducibility tool [30]. Before the publication of this work, the only large reproducible experimental surveys on word similarity reported in the literature were those introduced by Lastra-Díaz and García-Serrano [1,9,10] in another reproducibility paper [20] belonging to this same reproducibility initiative [31], in which we also find other works such as those introduced by Wolke et al. [32] and Fariña et al. [33]. However, there is neither joint reproducible benchmarks on word embeddings and ontology-based semantic similarity measures nor other ones evaluating the latest family of methods on so large count of datasets as those evaluated by our primary work [27].

The lack of reproducibility of the methods and research results in the field of NLP has become a serious problem, which severely hampers any research effort and the smooth integration of newcomers in the field. This reproducibility gap was already highlighted in a pioneering work by Pedersen [34], being subsequently confirmed by Fokkens et al. [35] by evaluating several works in the same line of research tackled herein. More recently, Branco et al. [36] introduce a call for reproducibility submissions in a known NLP journal to bridge the aforementioned reproducibility gap. We subscribed to this reproducibility alarm by adopting as basic norm the detailed replication of all methods evaluated in our papers, as well as the warning on many contradictory or unreproducible results in a series of papers in this line of research, such as the works introduced by Lastra-Díaz and García-Serrano [1,9,10], Lastra-Díaz et al. [20] and our primary paper [27]. Likewise, in a recent and valuable reproducibility study in the field of NLP, Wieling et al. [37, p.641] found that only a third part of the published works in 2016 (36.2%) provided their source code; however, they found by evaluating a random sample of ten works that only a tenth part of the former group could be reproduced exactly. Thus, this later finding yields an alarming ratio of only a 3.62% of reproducible works in this aforementioned study. For all reasons above, we subscribe both the reproducible manifesto [38] for a reproducible science and reproducibility initiative lead by Information Systems [31], as well as the slow science manifesto[1] for a reflective research. Finally, we make our own the words of Pedersen [34, p.470]: "we might one day only accept for publication articles that are accompanied by working software that allows for immediate and reliable reproduction of results".

The aim of this work is to introduce a detailed experimental setup based on a collection of publicly available software tools [28] and reproducibility resources [23,39], which are provided as supplementary material, with the aim of exactly reproducing all experiments and results reported in our primary work [27].

[1] http://slow-science.org/

1.1. Contributions and plan of this paper

Our main contribution is the introduction of a self-contained and easily reproducible set of experiments on word similarity and relatedness, which allow to reproduce all experiments, results, and conclusions introduced by our primary work [27] exactly. We provide a very detailed reproducibility protocol together with a set of software tools [28] and a companion reproducibility dataset [23] which is publicly available at [39]. We also detail how our reproducible experiments could be extended for setting up and evaluating unconsidered experimental setups including other datasets, word embeddings, or ontology-based semantic similarity measures.

A second contribution is the introduction, for the first time, of a self-contained, reproducible and extensible collection of benchmarks on word similarity and relatedness which jointly evaluate the most recent methods on the families of ontology-based semantic similarity measures and word embedding models on a same software platform, and consequently, being amenable to become a de facto standard for large scale experimentation in this line of research. Despite the huge research effort done during the last decades, such as witnessed by the plethora of methods reviewed and evaluated in our primary work [27], there is still a lack of a fully automatic, reproducible and extensible collection of benchmarks which make the evaluation and development of word similarity and relatedness methods easier. In general, there is a lack of reproducibility resources in this line of research which was partially bridged by the introduction of several semantic measures libraries, such as SML [40], SISR [21], and the most recent called HESML [20], which is the largest and most efficient among them, in addition to provide self-contained and easily reproducible experiments for the first time. Likewise, the reproducible experiments and reproducibility datasets introduced by Lastra-Díaz et al. [20] and Lastra-Díaz and García-Serrano [22,25,26] respectively have allowed for the first time to reproduce a set of large experimental surveys on ontology-based semantic similarity measures based on WordNet [1,9,10] exactly. However, recent and fast advances in the family of word embedding models together with the active research on ontology-based methods have raised the need to carry-out joint evaluations of both families of methods in a large set of benchmarks to elucidate the state of the problem, as done in our primary work [27].

The rest of the paper is structured as follows. Section 2 introduces the HESML library [20] and the Reprozip tool [31], which set the software platform originally used to run all experiments introduced herein and our long-term reproducibility platform respectively. Section 3 introduces the new reproducible experiments on word similarity, whilst Section 4 details how them can be extended, or created new ones from scratch. Section 5 introduces a corrigendum for several data tables reported in our primary work [27] to fix a mismatch detected in the MC28 [41] similarity scores used in our original experiments. Finally, we introduce our conclusions and future work.

2. Background on HESML and Reprozip

HESML [20] is a self-contained Java software library of semantic measures based on WordNet whose latest version, called HESML V1R4 [28], also supports the evaluation of pre-trained word embedding models, such as those introduced by Mikolov et al. [43], Pennington et al. [44], Schwartz et al. [45], Wieting et al. [46], Goikoetxea et al. [14], Bojanowski et al. [47], Agirre and Soroa [48], Camacho-Collados et al. [49] and Mrkšić et al. [50]. HESML is a self-contained experimentation platform on word similarity and relatedness which is especially well suited to run large experimental surveys by supporting the execution

Table 1
Technical and legal information of the latest version of the HESML software library [20] used in our experiments.

HESML software library	Description
Latest version.	V1R5
Code version used in this work.	V1R4
Legal Code License.	Creative Commons By-NC-SA 4.0
Permanent code repository of HESML V1R5 [42]	https://doi.org/10.21950/1RRAWJ
Permanent code repository used for this work.	http://dx.doi.org/10.17632/t87s78dg78.4
GitHub project repository	https://github.com/jjlastra/HESML.git
Software code languages and tools.	Java 8, Java SE DevKit 8, NetBeans 8.0 or higher
Compilation requirements and operating systems.	Java SE Dev Kit 8, NetBeans 8.0 or higher and any Java-compliant operating system.
Documentation and source code examples	Sample source code in the HESMLclient program.
HESML web site	http://hesml.lsi.uned.es
Community forum	hesml+subscribe@googlegroups.com, hesml+unsubscribe@googlegroups.com

Table 2
Technical and access information of Reprozip long-term reproducibility tool [31].

Reprozip tool	Description
Current version	1.0.16
Web site	https://www.reprozip.org
Supported platforms	Linux, Windows and MacOS

Table 3
Our two methods to reproduce all experiments and results introduced by our primary work [27]. HESMLclient method is that originally used to run our experiments in our primary work, whilst ReproUnzip provides a long-term reproducibility method regardless the original testing platform used to run our experiments.

Software used	Supported reproducibility methods
HESMLclient [28]	You should download HESML V1R4 [28] and a supplementary ZIP file containing the collection of pre-trained word embedding files (*WordEmbeddings.zip* [39]), and then run *HESMLclient* with the reproducible file as input, as detailed in Section 3.4.
ReproUnzip [30]	You should download our supplementary Reprozip file [39] and setting up and running Reprounzip as detailed in Section 3.5.

of automatic reproducible experiment files based on a XML-based file format (*.exp). Despite the latest version of HESML only supports WordNet, it could be easily extended to manage other ontologies by implementing the proper parsers as detailed by Lastra-Díaz et al. [20]. HESML library has been completely developed in NetBeans 8 and Java 8, being distributed with three WordNet versions, whilst *HESMLclient* is a complementary Java console program whose aim is to run word similarity experiments by calling HESML functionality. For a detailed introduction to HESML, we refer the reader to its introductory paper [20]. Table 1 shows a summary of technical and legal information of the latest HESML version used in our experiments.

On the other hand, ReproZip is a virtualization tool introduced by Chirigati et al. [30], whose aim is to warrant the exact replication of experimental results onto different systems from that originally used in their creation. Reprozip captures all the program dependencies and is able to reproduce the packaged experiments on any host platform, regardless of the hardware and software configuration used in their creation. Thus, ReproZip warrants the reproduction of the experiments introduced herein in the long term. Other valuable feature of Reprozip is that it allows to modify the input files of any Reprozip package with the aim of evaluating a set of experiments using originally unconsidered methods, configuration parameters or datasets. Reprozip supports main virtualization platforms as Docker and VirtualBox; however, our preferred option is Docker. For a comparison of these two types of virtualization platforms, we refer the reader to the survey introduced by Merkel [51], in which the author introduces Docker and compares it with classic Virtual Machines (VM), such as VirtualBox. Finally, Reprozip also simplifies the generation, packaging, and execution of Docker-based experiments. For all reasons above, we encourage the research community to use Reprozip as a long-term reproducibility backup. Table 2 shows a summary of technical and access information of the Reprozip reproducibility tool.

3. The reproducible experiments on word similarity

The aim of this section is to introduce a set of detailed experimental setups in order to replicate the methods and experiments introduced by our primary work [27] exactly. Section 3.1 details the experimental setup for the implementation of our experiments in our primary work [27], then Section 3.2 details the minimal system requirements for the testing platforms with the aim of running our reproducible experiments. Likewise, Section 3.2 reports the running times obtained by the authors and reviewers in the evaluation of our reproducible experiments in different testing platforms. Section 3.3 details the procedure for obtaining and compiling HESML source code, as well as running its pre-compiled jar files. We note that it is not needed to compile the HESML source code to run the experiments, because the HESML distribution already includes pre-compiled versions of the HESMLclient program with the latest HESML version. Next, Section 3.4 introduces the method to run our experiments which is based on the running of HESMLclient program, whilst Section 3.5 introduces our long-term reproducibility method based on Reprozip. Finally, Section 3.6 introduces the automated data analysis carried-out to process the raw similarity values generated by our experiments and computing all evaluation metrics reported in our aforementioned primary work [27], as well as a report in HTML file format showing all data tables generated from our raw data.

3.1. Experimental setup in our primary paper

All experiments carried-out in our primary paper [27] were implemented in *HESML V1R4* [28] by running *HESMLclient* program with a reproducible experiment file in XML-based (*.exp) file format, which encodes the evaluation of all semantic measures in all datasets as listed in Table 4. The experimental setup and software platform used to implement all our experiments is detailed in [27, figure 3]. HESML V1R4 implements all ontology-based semantic similarity measures based on WordNet 3.0, as well as all pre-trained word embedding models evaluated in our benchmarks. In addition, the execution of our experiments was recorded into a long-term reproducibility Reprozip file, called "*WN_ontology_measures_vs_embeddings.rpz*", which is part of our companion reproducibility dataset [23], being publicly available

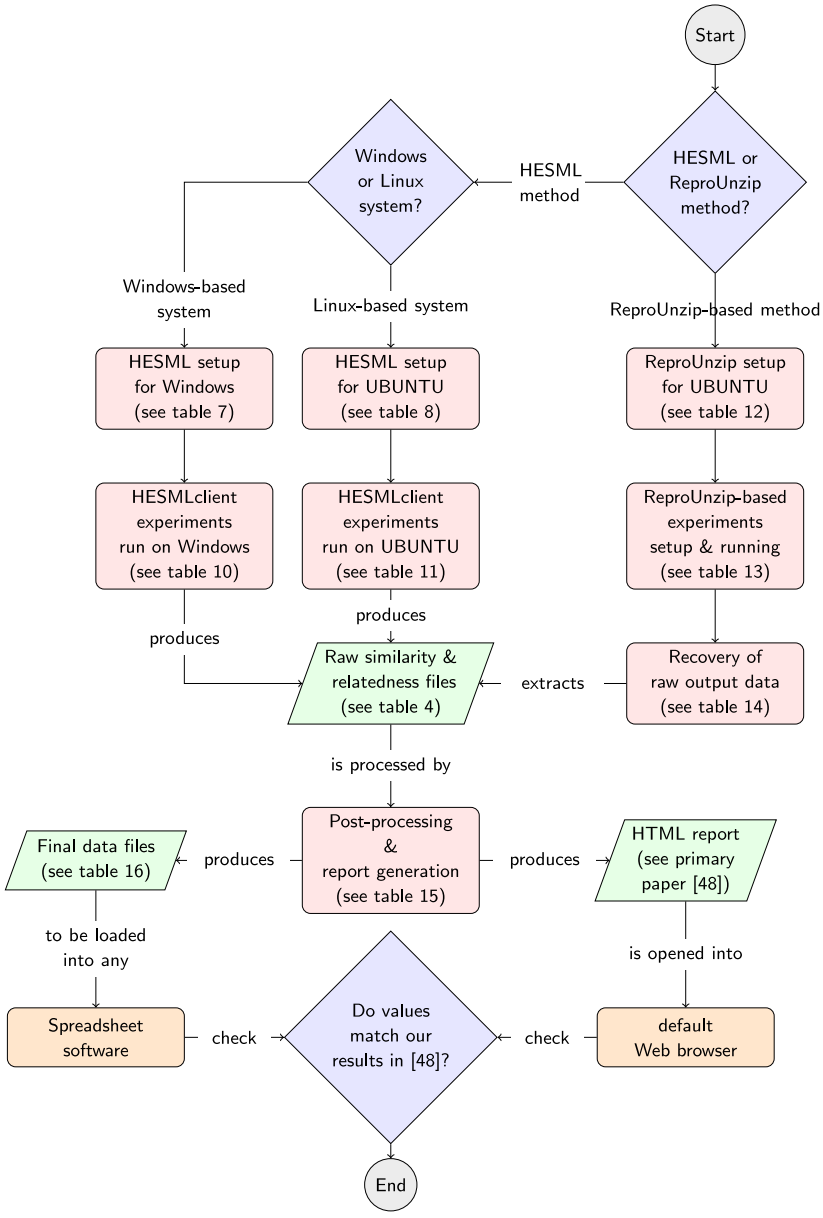

Fig. 1. Reproducibility workflows using either HESMLclient or ReproUnzip programs to run the reproducible experiments introduced herein. The three workflows detailed above produce the same raw and processed data files, as well as a collection of HTML pages which reproduce all data tables reported by our primary paper [27].

at [39]. Our aforementioned Reprozip file can be reproduced in any Reprozip compliant platform.[2] as detailed in Section 3.5. Thus, all our methods, experiments, and results can be reproduced by using two different software platforms and methods as detailed in Table 3. First reproducibility method is based on the execution of *HESMLclient* program, whilst second one is based on the execution of the aforementioned Reprozip file.

The two reproducibility methods cited in Table 3 were introduced in the HESML paper [20], which provides a detailed protocol to reproduce all experiments, results, data tables and figures reported in three papers previously introduced by Lastra-Díaz and García-Serrano [1,9,10], as well as the benchmarks between semantic measures libraries reported in [20]. All experiments detailed herein were originally implemented on an UBUNTU 16.04 virtual computer with 8 Gb of RAM and 100 Gb of disk space called UBUNTU-base1 as detailed in Table 5. However, it could be reproduced in any Java 8, or Reprounzip compliant platform, by using any of the two aforementioned methods above, which includes most Linux-based, MacOS-based and Windows-based platforms. For this reason, our experiments have been successfully reproduced using both HESMLclient and ReproUnzip methods (see Fig. 1) in all testing platforms detailed in Table 5, with the running times reported in Table 6.

Fig. 1 shows the three reproducibility workflows introduced herein, which are defined by the selection of one of the two reproducibility methods shown in Table 3 with a specific testing platform. HESML distribution includes the pre-compiled version of HESML V1R4 and HESMLclient.jar files, thus any reader interested in reproducing our experiments can directly follow the setup instructions in Tables 7 and 8, and subsequently running the experiments as detailed in Tables 10 and 11. On the other hand, Table 4 shows the full collection of reproducible experiments encoded by the *"benchmark_survey.exp"* file (see Fig. 2), as well as the corresponding raw output files that are generated

[2] https://www.reprozip.org/

Table 4
Collection of raw output files generated by the execution of the "benchmark_survey.exp" reproducible experiment file by any of the two aforementioned reproducibility methods. Each raw output file contains the raw similarity or relatedness values returned for each word pair by each semantic measure. These raw output files are subsequently processed by a R-language script to produce the final data tables shown in our primary paper [27], as detailed in Section 3.6. For further details on the datasets above, we refer the reader to our primary paper [27, table 3].

Word similarity and relatedness benchmarks reproduced herein [27]		
Dataset	Type	Raw output files generated by our experiments
MC28 [41]	Similarity	raw_similarity_values_MC28_dataset.csv
RG65 [52]	Similarity	raw_similarity_values_RG65_dataset.csv
PS_{full} [53]	Similarity	raw_similarity_values_PSfull_dataset.csv
Agirre201 [18]	Similarity	raw_similarity_values_Agirre201_lowercase_dataset.csv
SimLex665 [54]	Similarity	raw_similarity_values_SimLex665_dataset.csv
MTurk771 [55]	Relatedness	raw_similarity_values_MTurk771_dataset.csv
MTurk287/235 [56]	Relatedness	raw_similarity_values_MTurk287-235_dataset.csv
WS353Rel [57]	Relatedness	raw_similarity_values_WS353Rel_dataset.csv
Rel122 [58]	Relatedness	raw_similarity_values_Rel122_dataset.csv
WS353Full [57]	Relatedness	raw_similarity_values_WS353Full_dataset.csv
SimLex111 [54]	Similarity	raw_similarity_values_SimLex111_dataset.csv
SimLex222 [54]	Similarity	raw_similarity_values_SimLex222_dataset.csv
SimLex999 [54]	Similarity	raw_similarity_values_SimLex999_dataset.csv
SimVerb3500 [59]	Similarity	raw_similarity_values_SimVerb3500_dataset.csv
MEN [60]	Relatedness	raw_similarity_values_MEN_dataset.csv
YP130 [61]	Relatedness	raw_similarity_values_YP130_dataset.csv
RW2034 [62]	Relatedness	raw_similarity_values_RareWords2034_dataset.csv
RW1401 [62]	Relatedness	raw_similarity_values_RareWords1401_dataset.csv
SCWS [63]	Relatedness	raw_similarity_values_SCWS1994_dataset.csv

during its execution, whose subsequent processing allows to reproduce the results reported in our primary paper [27] exactly, as detailed in Section 3.6.

HESML V1R4 distribution [28] contains all source files and pre-compiled versions of the *HESML-V1R4.jar* library and the *HESMLclient.jar* Java console program. Thus, it is enough to download its official distribution from Mendeley [28], or GitHub[3], in order to run our experiments. However, for the sake of completeness, Section 3.3 introduces the detailed steps to obtain and compile HESML V1R4. Finally, we introduce a companion reproducibility dataset [23] which is publicly available at [39]. This aforementioned reproducibility dataset gathers into a common repository all data files required to reproduce our experiments with the aim of providing a consolidated and permanent version of these files, and thus avoiding the tedious work of gathering all these stuff, as well as any risk of alteration or unavailability of them in the future.

3.2. System requirements and performance evaluation

Table 5 shows the testing platforms in which we have successfully reproduced the experiments detailed herein, whilst Table 6 shows their running times in the completion of all experiments for each aforementioned reproducibility method and testing platform. The configuration of these platforms sets the minimal system requirements to reproduce our experiments. Unlike the execution of our experiments using *HESMLclient* program on the UBUNTU-based computers detailed in Table 5, the execution using Reprounzip demands much more disk space because it needs to setup a docker container to run the experiments. For this reason, UBUNTU-Reprounzip platforms shown in Table 5 are based on a minimal overall disk space of 200 Gb to allow the set up of UBUNTU, Docker, and the resources required by our Reprozip package.

3.3. Obtaining and compiling HESML

Table 1 shows the technical information required to obtain and compile the *HESML* source code and run the experiments detailed in Table 4. HESML V1R4 distribution includes compiled versions of *HESML* library and the *HESMLclient* program, thus this later program could be directly used without the need of compiling the source code in NetBeans. There are two different ways of

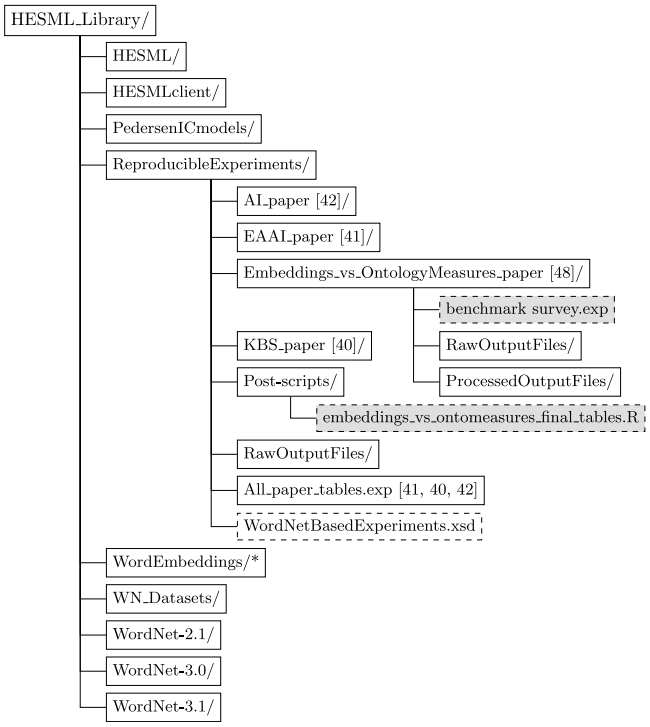

Fig. 2. Directory structure of the HESML library once it has been extracted onto disk. The reproducible experiment file and the post-processing R-language script used to reproduce and generate our final data tables, respectively, are shown in dashed-line boxes in gray, whilst XML-based experiment file format is detailed by XML-schema file shown in unfilled dashed-line box. (*) WordEmbeddings folder contains the pre-trained files for all word embedding models used in our experiments; however, this folder is neither included by the HESML V1R4 [28] distribution nor HESML V1R4 release at GitHub repository because its large size. Thus, you must download the "*WordEmbeddings.zip*" file [39] and extract it onto the main *HESML_Library* directory to retrieve this folder and its content.

[3] https://github.com/jjlastra/HESML.git

Table 5
Testing platforms successfully used to reproduce our experiments. Virtual computers are cloud-based servers based on the OpenStack virtualization platform [64]. Ubuntu-base1 and Ubuntu-base2, as well as Ubuntu-base3 and Ubuntu-base4, differ only in the disk space demanded by Reprounzip. On the other hand, Ubuntu-base1 and Ubuntu-base2 differ from Ubuntu-base3 and Ubuntu-base4 in that these two later platforms use a more modern CPU than the former ones, which were used in the implementation of our original experiments in [27]. For this reason, the experiments reproduced on Ubuntu-base3 and Ubuntu-base4 configurations report lower running times than the former ones as shown in Table 6.

Testing platform	Type	Operating Sys.	Configuration	Tested by
Ubuntu-base1	Virtual	Ubuntu 16.04	1 Core Intel E5-2640-v2 CPU @2 GHz, 8 Gb RAM,100 Gb SSD disk	Authors
Ubuntu-base2	Virtual	Ubuntu 16.04	2 Intel Core Xeon E5 2699-v4 CPU @2.2 GHz, 8 Gb RAM, 100 Gb SSD disk	Authors
Ubuntu-base3	Virtual	Ubuntu 16.04	1 Core Intel E5-2640-v2 CPU @2 GHz, 8 Gb RAM, 200 Gb SSD disk	Authors
Ubuntu-base4	Virtual	Ubuntu 16.04	2 Intel Core Xeon E5 2699-v4 CPU @2.2 GHz, 8 Gb RAM, 200 Gb SSD disk	Authors
Windows-base1	Laptop	Windows 10 × 64	1 Intel Core i7-5500U CPU @2.4 GHz, 16 Gb RAM,100 Gb SSD disk	Authors
Windows-rev	Desktop	Windows 10 × 64	1 Intel Core i5-6400 @2.7 Ghz (3.3 Ghz Turbo), 16 Gb RAM, 240 Gb SSD disk	Reviewer
Ubuntu-rev	Server	Ubuntu 19.10	1 Intel Core i7-2600 @3.4 Ghz (3.8 Ghz Turbo), 16 Gb RAM, 500 Gb mechanical disk	Reviewer

Table 6
Running times obtained on different testing platforms for the execution of all benchmarks by using HESMLclient program with the 'benchmark_survey.exp' experiment file, or by running ReproUnzip program with the "WN_ontology_measures_vs_embeddings.rpz" file. (*) Comment from the reviewer: "this computer was a server devoted to handle other services, so there was no exclusive access to the CPU to run the experiments. This fact and the older hardware configuration justify the significantly larger runtime".

Run	Testing platform	Method	Running time	Tested by
1	Ubuntu-base1	HESMLclient	17581 min ≈ 12.2 days	Authors
2	Ubuntu-base3	ReproUnzip	18109 min ≈ 12.6 days	Authors
3	Ubuntu-base2	HESMLclient	9622 min ≈ 6.68 days	Authors
4	Ubuntu-base4	ReproUnzip	11732 min ≈ 8.15 days	Authors
5	Windows-base1	HESMLclient	10 days	Authors
6	Ubuntu-base2	HESMLclient	10201 min ≈ 7.08 days	Authors
7	Windows-rev	HESMLclient	10800 min ≈ 7.5 days	Reviewer
8	Ubuntu-rev	ReproUnzip	21768* min ≈ 15.12 days	Reviewer

Table 7
Detailed instructions for downloading HESML V1R4 onto a Windows-based system from its GitHub repository.

Step	Windows-based setup instructions for HESMLclient experiments
(1)	Install Java 8 runtime or higher in your workstation.
(2)	Open a PowerShell console (Windows 7 and higher) in any directory.

(3)	Create a working directory and move to it as follows:
	$ mkdir REPRODIR
	$ cd REPRODIR

(4)	Download and extract the latest HESML version from its GitHub repository (see URL below) using either any Web browser or PowerShell as detailed below: $ powershell -command "& { iwr https://github.com/jjlastra/HESML/archive/master.zip }" $ Expand-Archive ./master.zip .

(5)	Download the *WordEmbeddings.zip* file from our Dataverse repository [39, see URL below]
	and extract it onto HESML root directory using either any Web browser or PowerShell as detailed below:
	$ cd HESML-master/HESML_LIBRARY
	$ mkdir WordEmbeddings
	$ cd WordEmbeddings
	$ powershell -command "& { iwr https://doi.org/10.21950/wordembeddings.zip }"
	$ Expand-Archive ./wordembeddings.zip .

obtaining the *HESML* source code as follows: (1) by downloading the HESML V1R4 version from the permanent Mendeley Data link [28]; or (2) by downloading it from its GitHub repository detailed in Table 1. You could also use the latest HESML version (V1R5), which is available at its permanent repository [42] and HESML GitHub repository. Once the HESML source code has been downloaded and extracted onto your hard drive, the project will have the folder structure shown in Fig. 2 and detailed below:

HESML is the main software library folder containing the NetBeans project and the HESML source code. Below this folder you find the *dist* folder which contains the *HESML-V1R4.jar* distribution file generated during the compilation, whilst *HESMLclient* folder contains the source code of the HESMLclient console application. The main aim of the *HESMLclient.jar* application is to provide a collection of sample functions in order to show the HESML functionality, as well as running any (*.exp) reproducible experiment file.

PedersenICmodels folder contains the full WordNet-InfoContent-3.0 collection of WordNet-based frequency files created by Ted Pedersen [65]. The file names denote the corpus used to build each file. The readme file details the method used to build the frequency files, which is also detailed in [66].

ReproducibleExperiments folder contains one subfolder for each paper introduced by Lastra-Díaz and García-Serrano [1,9,10] and our primary paper [27] reproduced herein. Likewise, the aforementioned folder also contains a XML-schema file called "*WordNetBasedExperiments.xsd*", which describes the syntax of all XML-based experiment files (*.exp), and the *All_paper_tables.exp* file with the definition of all the reproducible experiments corresponding to the three aforementioned papers of Lastra-Díaz and García-Serrano. All (*.exp) files have been created with the XML Spy editor. In addition, this folder contains the *RawOutput-Files* subfolder with all the raw output files of the three aforementioned papers [1,9,10].

Post-scripts folder contains a set of post-processing R scripts which process the raw output files generated by all reproducible experiments to generate all final data tables and figures reported in our papers exactly.

Table 8
Detailed instructions for downloading HESML V1R4 from its GitHub repository onto a Linux-based system.

Step	Linux-based setup instructions for HESMLclient experiments

(1)	Install Java 8 and Java SE Dev Kit 8 or higher as follows:
	$ sudo apt-get update
	$ sudo apt-get -y install default-jdk

(2)	Install UNZIP program as follows:
	$ sudo apt-get update
	$ sudo apt-get -y install unzip

(3)	Create a working directory and move to it as follows:
	$ mkdir REPRODIR
	$ cd REPRODIR

(4)	Download the latest HESML version from GitHub (see URL below) as follows: $ wget https://github.com/jjlastra/HESML/archive/master.zip
(5)	Extract *master.zip* file onto your working directory as follows: $ unzip master.zip

(6)	Download the *WordEmbeddings.zip* file from our Dataverse repository [39, see URL below] and extract it onto HESML root directory as detailed below.
	$ cd HESML-master/HESML_LIBRARY
	$ wget https://doi.org/10.21950/wordembeddings.zip
	$ unzip wordembeddings.zip -d WordEmbeddings

Table 9
Detailed instructions for compiling HESML onto any Windows or Linux-based system. We recall that the compilation of HESML is not needed to run all experiments introduced herein.

Step	Detailed instructions to compile HESML
(1)	Follow the step-by-step procedures to download the HESML source code as detailed in Tables 7 and 8 for Windows or Linux-based systems respectively.
(2)	Install Java SE Dev Kit 8 and NetBeans 8.0.2 or higher in your workstation.
(3)	Launch NetBeans IDE and open the *HESML* and *HESMLclient* projects contained in the HESML root folder as shown in Table 2. NetBeans automatically detects the presence of a *nbproject* subfolder with the project files.
(4)	Select *HESML* and *HESMLclient* projects in the project treeview respectively. Then, invoke the "Clean and Build project (Shift + F11)" command in order to compile both projects.

WN_datasets folder contains a collection of (*.csv) data files with fields separated by semicolon which correspond to the word similarity benchmarks shown in Table 4, whilst *WordNet-2.1*, *WordNet-3.0* and *WordNet-3.1* contain the database files of three different versions of WordNet.

Embeddings_vs_OntologyMeasures_paper folder contains the reproducible experiment file "benchmark_survey.exp" encoding all benchmarks introduced herein and detailed in Table 4. In addition, this folder contains a subfolder called "RawOutputFiles" with all raw output similarity files generated by our experiments. The R-language script file called "embeddings_vs_ontomeasures_final_tables.R" generates all files in "ProcessedOutputFiles" subfolder.

Tables 7 and 8 show a detailed step-by-step procedure to set up our reproducible experiments based on HESML on any Windows or Linux-based system respectively. HESML distribution includes pre-compiled versions of HESMLclient program and HESML library; thus, you could skip the compilation step for running our experiments. However, for the sake of completeness, we briefly detail the compilation steps in Table 9.

3.4. Running the experiments with HESMLclient

Once you have downloaded and extracted the *HESML V1R4* library onto your hard drive as detailed in Section 3.3, you are ready to run the reproducible experiments by following the steps detailed in Tables 10 and 11 for testing platforms based on Windows and Linux respectively. However, before running the experiments, you must download the *WordEmbeddings.zip* file [39] and extract it onto the main *HESML_Library* directory as detailed in step 5 of Table 7 for Windows, and step 6 of Table 8 for the Linux-based case. This later ZIP file contains all pre-trained word embedding files; however, it is not included in the current HESML distribution because of its large size and the space limitations of GitHub and Mendeley repositories. We note that the original *HESMLclient* source code is defined to fetch the required input files from the folder structure of *HESML* as shown in Fig. 2.

3.5. Running the ReproZip experiments

The ReproZip[4] program was used for recording and packaging the running of the *HESMLclient* program with all the reproducible experiments defined by the "*benchmark_survey.exp*" file into the "*WN_ontology_measures_vs_embeddings.rpz*" file, which is publicly available at our UNED Dataverse repository [39]. This later Reprozip file was generated by running Reprozip on the Ubuntu-base1 workstation detailed in Table 5; however, in order to run ReproUnzip based on Docker as detailed below is needed to set up an Ubuntu-Reprounzip platform (see Table 5). Because the execution of the experiments takes long time, and Reprounzip with Docker cannot be executed in background mode without any output console, we will setup and use the "screen" program on Linux.

In order to set up and run the reproducible experiments introduced herein, you need to use ReproUnzip. ReproUnzip can be used with two different virtualization platforms: (1) Vagrant + VirtualBox, or (2) Docker. However, because of its simple setup and computational efficiency, our preferred ReproUnzip configuration is that based on Docker. For instance, in order to setup ReproUnzip based on Docker for Ubuntu, you should follow the detailed steps shown in Table 12, despite several steps possibly being unnecessary depending on your starting configuration. Once ReproUnzip and Docker have been successfully installed, Table 13 shows the detailed instructions to set up and run the reproducible experiments. Those readers who prefer to use ReproUnzip with VirtualBox instead of Docker can consult the ReproZip installation page.[5]

[4] https://www.reprozip.org/
[5] https://reprozip.readthedocs.io/en/1.0.x/install.html

Table 10
Detailed instructions for running our experiments with the *HESMLClient* program on any testing platform based on Windows.

Step	HESMLclient running instructions on any Windows-based system
(1)	Open a command console in the *HESMLclient* directory as shown in Fig. 2. $ cd REPRODIR/HESML_Library/HESMLclient
(2)	Run the following command with the reproducible experiment file: $ java -jar -Xms4096m dist/HESMLclient.jar *../ReproducibleExperiments/Embeddings_vs_OntologyMeasures_paper/benchmark_survey.exp*
(3)	Command in step 2 above will generate all raw output files listed in Table 4 onto *../ReproducibleExperiments/Embeddings_vs_OntologyMeasures_paper* folder (see Fig. 2).

Table 11
Detailed instructions for running our experiments with the *HESMLClient* program on any testing platform based on Linux.

Step	HESMLclient running instructions on any Linux-based system
(1)	Open a Linux command console in the *HESMLclient* directory (see Fig. 2). user@server$ cd REPRODIR/HESML_Library/HESMLclient
(2)	We create a "screen" session and run HESMLclient in background. Note that HESMLclient execution could take up to two weeks (see Table 6). user@server$ screen -S REPROEXPS user@screen$ java -jar -Xms4096m dist/HESMLclient.jar *../ReproducibleExperiments/Embeddings_vs_OntologyMeasures_paper/benchmark_survey.exp*
(3)	We detach from "screen" before to close the server main console user@screen$ CTRL+a, d
(4)	We reattach to the screen console to check the completion of HESMLclient user@server$ screen -r REPROEXPS
(5)	We destroy the "screen" console once finished HESMLclient execution user@server$ screen -X -S REPROEXPS quit
(6)	Second command in step (2) above will generate all raw output files listed in Table 4 onto *../ReproducibleExperiments/Embeddings_vs_OntologyMeasures_paper* folder (see Fig. 2).

Table 12
Detailed instructions on installing ReproUnzip with Docker for Ubuntu. Despite that steps above could look tedious, we prefer that readers are aware of all packages being installed instead of running a single setup script hiding this information.

Step	Detailed setup instructions for ReproUnzip on any Linux-based system
	Steps (1–4) below install Reprounzip and all its dependencies.
(1)	$ sudo apt-get update
(2)	$ sudo apt-get -y install libffi-dev libssl-dev openssl openssh-server In some systems, python3-pip should be used instead of python-pip in step (3). Thus, you should use "pip3" instead of "pip" in step (4) below.
(3)	$ sudo apt-get -y install libsqlite3-dev python-dev python-pip screen
(4)	$ sudo pip install reprounzip[all]
	Steps (5–11) below install the latest version of Docker CE whilst step 11 checks its installation. For further details, we refer the reader to the official Docker setup page: https://docs.docker.com/install/linux/docker-ce/ubuntu/
(5)	$ sudo apt-get -y install apt-transport-https ca-certificates
(6)	$ sudo apt-get -y install curl gnupg-agent software-properties-common
(7)	$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
(8)	$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
(9)	$ sudo apt-get update
(10)	$ sudo apt-get -y install docker-ce docker-ce-cli containerd.io
(11)	$ sudo docker run hello-world

The running of the reproducible experiments based on Docker for Ubuntu took approximately one week in a modern virtual computer as detailed in Table 6. Once the running is completed, you should follow the instructions shown in Table 14 to retrieve the raw output files from the Docker container, as listed in Table 4. Finally, Table 6 reports a sample of software platforms in which the Reprozip-based experiments introduced herein have been successfully reproduced.

3.6. Processing of the raw output files

The running of the "*benchmark_survey.exp*" experiment file generates the collection of comma-separated files (*.csv) listed in Table 4, whose values are separated by a semicolon. All raw output files are saved in the same folder as their corresponding input reproducible experiment file.

Raw output similarity files generated by our experiments must be processed in order to compute the Pearson, Spearman, and Harmonic score metrics matching the tables shown in our primary paper [27]. We provide a R-language script called "*embeddings_vs_ontomeasures_final_tables.R*" with the aim of automating this post-processing. The latest version of the aforementioned post-processing script should be obtained from HESML GitHub distribution, as detailed in Tables 7 and 8, or from our companion reproducibility dataset [39]. This aforementioned script includes the source code of the *mat.sort* function provided by the BioPhysConnectoR package [67], which is no longer available in CRAN server.

In order to carry-out the aforementioned post-processing, you should setup either the R statistical program[6] or RStudio[7] in your workstation and follow the steps detailed in Table 15. Then, you need to install the "knitr" and "readr" packages using the functionality provided for this task by any of the two aforementioned programs. Table 16 shows the output files which are generated from the raw output files listed in Table 4 by running our aforementioned post-processing script, as well as their corresponding data tables in our primary paper [27]. In addition, our post-processing script generates a collection of HTML files which contain all data tables reported in our primary paper [27].

6 https://www.r-project.org/
7 https://rstudio.com/

Table 13
Detailed instructions on how to reproduce the packaged experiments once Reprounzip has been installed. We use *screen* program with the aim of allowing the execution of Reprounzip in background whilst main program console is detached and closed.

Step	Detailed setup and running instructions for our Reprozip-based experiments
(1)	Setup the Reprounzip program onto any supported platform (Linux, Windows and MacOS) by following the step-by-step guide detailed in Table 12 (see ReproZip installation page for further information).

(2)	Move to the home directory and create a working directory as follows
	$ cd /home
	$ mkdir REPROEXPS
	$ cd REPROEXPS

(3)	Download the *WN_ontology_measures_vs_embeddings.rpz* (12.4 Gb) from its repository [39]. For instance, you can execute the command below. The download of this file could takes several minutes. $ wget http://dx.doi.org/10.21950/wn_ontology_measures_vs_embeddings.rpz
(4)	Next, we must setup the docker container as detailed below which could take up to 45 min depending of your testing platform. Thus, we recommend to create a "screen" session to run in background both setup and running of the Reprozip-based experiment. You can detach from "screen" console by pressing "Ctrl+a,d". user@server$ screen -S REPROUNZIP user@server$ reprounzip docker setup wn_ontology_measures_vs_embeddings.rpz docker_folder
(5)	Next, we will run the Reprozip-based experiment. Note that Reprounzip execution could take up to two weeks depending on your hardware setup (see Table 6). We strongly recommend to keep open the *screen* console to run the experiment in background as detailed below. user@screen$ reprounzip docker run docker_folder - We detach from "screen" before to close the server main console user@screen$ CTRL+a, d - We reattach to the screen console to check the completion of Reprounzip user@server$ screen -r REPROUNZIP - We destroy the "screen" console once finished Reprounzip execution user@server$ screen -X -S REPROUNZIP quit

Table 14
Detailed instruction to recover the output files generated by our Reprozip-based experiments. The first instruction shows a list with the output files generated by the experiments, whilst the second one extracts all the output files from the container and downloads them onto the current folder. You should obtain all raw output files listed in Table 4.

Step	Recovering the output files generated by our Reprozip-based experiments
1	user@server$ reprounzip showfiles docker_folder
2	user@server$ sudo reprounzip docker download −−all docker_folder

Table 15
Detailed instructions for the post-processing of the raw output files generated by our experiments. R-language script computes all Pearson, Spearman and Harmonic score metrics and generates a HTML report file reproducing all data tables reported by our primary paper Lastra-Díaz et al. [27].

Step	Detailed post-processing instructions based on a R-language script
(1)	Launch the R statistical program and install *knitr* and *readr* packages.
(2)	Launch the R statistical program (you could also use R-Studio).
(3)	Select the menu option "*File->Open script*". Then, load the R-language script file called *embeddings_vs_ontomeasures_final_tables.R* contained in the folder shown in figure Fig. 2. The latest version of the aforementioned script should be obtained from HESML GitHub distribution or our companion reproducibility dataset [39].
(4)	Edit the *rootDir*, *inputDir* and *outputDir* variables at the beginning of the script in order to set the directory which contains the raw output files onto your hard drive, as well as the directory in which will be saved the final assembled data tables as reported in our primary paper [27]. IMPORTANT NOTE: *inputDir* and *outputDir* variables should end with slash '/' symbol.
(5)	Select the menu option "*Edit->Run all*". The final assembled data tables will be saved in the output directories defined above, as detailed in Table 16. In addition, the aforementioned R script creates a and opens a collection of HTML files which show all data tables in our primary paper [27] and detailed in Table 16.

Finally, raw data files and processed data files shown in Tables 4 and 16 respectively could be loaded into any spreadsheet software to carry-out any further data analysis or confirming the reproducibility of the experiments and results reported by our primary paper [27].

4. Extending and reusing our reproducible experiments

Our reproducible experiments are based on the XML-based HESML experiment file format (*.exp) whose specification is detailed by the "*WordNetBasedExperiments.xsd*" schema file distributed with HESML library as shown in Fig. 2. Both *.exp experiment files and *.xsd schema file were created with XML Spy editor. Next paragraphs provide a detailed description of the main objects encoded by the HESML XML-based experiment file format, and how they could be used to create new experiments from scratch like those introduced herein.

HESML XML-Based experiment file format. Fig. 3 shows a sample file which has been extracted from the "benchmark_survey.exp" file encoding all reproducible experiments introduced herein. *WordNetBasedExperiments* is the root node, which contains the collection of word similarity or relatedness benchmarks to be evaluated, whilst *SingleDatasetSimilarityValuesExperiment* encodes a specific word similarity benchmark defined by a dataset, an output directory, and a collection of WordNet-based similarity measures and pre-trained word embedding models. *SimilarityMeasures* nodes encode ontology-based semantic similarity measures based on WordNet which could require a further Information Content (IC) model for its implementation, being declared below the *WordNetMeasures* node. Likewise, *RawWordVectorFiles* encodes the collection of pre-trained word embedding files to be evaluated in the same dataset. Both *SimilarityMeasures* and *RawWordVectorFiles* could be declared independently, and they could contain an unlimited number of methods to be

Table 16
Collection of processed output files generated by the execution of the "embeddings_vs_ontomeasures_final_tables.R" script file onto the *outputDir* directory and their corresponding tables in our primary work [27].

File#	Post-processing output files saved at "outputDir" directory	In primary paper [27]
1	table_Pearson_SimDatasets.csv	table 4 (full precision)
2	table_Pearson_SimDatasets_rounded.csv	table 4
3	table_Spearman_SimDatasets.csv	table 5 (full precision)
4	table_Spearman_SimDatasets_rounded.csv	table 5
5	table_Pearson_RelDatasets.csv	table 6 (full precision)
6	table_Pearson_RelDatasets_rounded.csv	table 6
7	table_Spearman_RelDatasets.csv	table 7 (full precision)
8	table_Spearman_RelDatasets_rounded.csv	table 7
9	table_joined_allEmbeddings_similarity.csv	table 8 (full precision)
10	table_joined_allEmbeddings_similarity_rounded.csv	table 8
11	table_joined_allEmbeddings_relatedness.csv	table 9 (full precision)
12	table_joined_allEmbeddings_relatedness_rounded.csv	table 9
13	table_pvalues_AttractReppel_allembeddings_similarity.csv	table A.1
14	table_pvalues_Paragramws_allembeddings_relatedness.csv	table A.2
15	table_AvgMeasures_Pearson_SimDatasets.csv	table A.3 (full precision)
16	table_AvgMeasures_Pearson_SimDatasets_rounded.csv	table A.3
17	table_AvgMeasures_Spearman_SimDatasets.csv	table A.4 (full precision)
18	table_AvgMeasures_Spearman_SimDatasets_rounded.csv	table A.4
19	table_AvgMeasures_Pearson_RelDatasets.csv	table A.5 (full precision)
20	table_AvgMeasures_Pearson_RelDatasets_rounded.csv	table A.5
21	table_AvgMeasures_Spearman_RelDatasets.csv	table A.6 (full precision)
22	table_AvgMeasures_Spearman_RelDatasets_rounded.csv	table A.6

XML-based HESML experiment file sample

```
<WordNetBasedExperiments>
    <SingleDatasetSimilarityValuesExperiment>
        <OutputFileName>raw_similarity_values_MC28_dataset.csv</OutputFileName>
        <DatasetDirectory>../WN\_Datasets</DatasetDirectory>
        <DatasetFileName>Miller_Charles_28_dataset.csv</DatasetFileName>
        <WordNetMeasures>
            <WordNetDatabaseFileName>data.noun</WordNetDatabaseFileName>
            <WordNetDatabaseDirectory>../Wordnet-3.0/dict</WordNetDatabaseDirectory>
            <SimilarityMeasures>
                <SpecificSimilarityMeasure>
                    <MeasureType>JiangConrath</MeasureType>
                    <IntrinsicICModel>Sanchez2011</IntrinsicICModel>
                </SpecificSimilarityMeasure>
                <SpecificSimilarityMeasure>
                    <MeasureType>Rada</MeasureType>
                </SpecificSimilarityMeasure>
            </SimilarityMeasures>
        </WordNetMeasures>
        <RawWordVectorFiles>
            <EmbVectorFiles>
                <VectorFile>../WordEmbeddings/attract-reppel.emb</VectorFile>
            </EmbVectorFiles>
            <UKBVectorFiles>
                <VectorFile>../WordEmbeddings/wordnet-ukb.ppv</VectorFile>
            </UKBVectorFiles>
            <NasariVectorFiles>
                <NasariVectorFile>
                    <SensesFile>../WordEmbeddings/nasari/en_wordsenses_BN.txt</SensesFile>
                    <VectorFile>../WordEmbeddings/nasari/nasari-unified</VectorFile>
                </NasariVectorFile>
            </NasariVectorFiles>
        </RawWordVectorFiles>
    </SingleDatasetSimilarityValuesExperiment>
</WordNetBasedExperiments>
```

Fig. 3. XML source code above shows an example of a HESML reproducible experiment on word similarity and relatedness. Source code above has been extracted from the "benchmark_survey.exp" file which encodes all experiments reported in our primary paper [27].

evaluated. The latest HESML version supports three different pre-trained word embeddings file formats which are defined by the *EmbVectorFiles*, *UKBVectorFiles* and *NasariVectorFiles* nodes. Raw output files which are generated by *SingleDatasetSimilarityValuesExperiment* procedures contain a matrix of values encoding the raw similarity value reported by each method for each word pair in the similarity dataset being evaluated.

Extending or modifying our experiments. Anyone could use our main aforementioned experiment file as a template to create new experiments from scratch by evaluating other sets of available ontology-based semantic similarity measures based on WordNet, pre-trained word embedding models, or word similarity datasets not considered herein. For instance, the reader could

Table 17
Pearson correlation (r) values for each Ontology-based semantic Topological similarity Measure (OTM), Ontology-based Vector Model (OVM) or Word Embedding (WE/WEC) model in the five noun similarity datasets. Measures (rows) are ranked according to their average value shown in Avg column. Best value for each dataset is shown in bold. (*) Last column shows p-values for an one-side t-Student distribution between Attract–repel [50] model and the remaining methods using the performance in the five noun similarity datasets as paired random sample with the aim of testing the hypothesis that Attract–repel significantly outperforms remaining methods in Pearson correlation. (**) The values reported in this column correspond to an evaluation of the MC28 dataset with its original similarity scores, which is carried-out in this work to fix a mismatch in the MC28 benchmark file used in our primary paper [27]. Our primary paper reports the values obtained in the evaluation of the MC28 dataset with the similarity scores of the RG65 dataset.

Corrigendum of our primary paper [27, table 4]			Pearson correlation (r) in noun similarity datasets						
Family	Measure	IC model	MC28 (**)	RG65	PS_{full}	Agirre201	SL665	Avg (r)	p-value(*)
WEC	Attract–repel [50]		0.847	0.840	0.893	0.720	0.691	**0.798**	–
OTM	coswJ&C [1]	Sánchez et al. [68]	**0.885**	**0.877**	0.885	0.695	0.592	0.787	0.340
OTM	$Cai_{strategy2}$ [69]	Cai et al. [69]	0.854	0.872	0.901	0.687	0.608	0.784	0.270
OTM	Hadj Taieb et al. [4]		0.826	0.867	**0.907**	0.708	0.609	0.784	0.240
OTM	Zhou et al. [70]	Seco et al. [71]	0.848	0.873	0.895	0.672	0.624	0.782	0.220
OTM	cosJ&C [1]	Sánchez et al. [68]	0.852	0.875	0.900	0.682	0.594	0.781	0.240
WEC	Counter-fitting [72]		0.824	0.806	0.866	0.701	**0.697**	0.779	0.023
WEC	Paragram-ws [46]		0.801	0.810	0.849	0.765	0.662	0.778	0.140
OTM	Pirró&Seco [73]	Seco et al. [71]	0.849	0.862	0.897	0.679	0.597	0.777	0.180
WEC	Paragram-sl [46]		0.799	0.798	0.854	0.748	0.682	0.776	0.100
OTM	Gao_{strat3} [74]	CPRefHypo [10]	0.826	0.865	0.891	0.674	0.614	0.774	0.120
OTM	Meng and Gu [75]	Seco et al. [71]	0.809	0.860	0.903	0.692	0.605	0.774	0.130
OTM	FaITH [76]	Seco et al. [71]	0.803	0.856	0.904	0.692	0.605	0.772	0.120
OTM	Lin [77]	Seco et al. [71]	0.819	0.861	0.894	0.680	0.601	0.771	0.110
OTM	Li_{strat3} [78]		0.828	0.862	0.885	0.664	0.606	0.769	0.099
OTM	Jiang&Conrath [79]	Sánchez et al. [68]	0.858	0.862	0.876	0.652	0.584	0.766	0.130
OTM	Leacock&Chodorow [80]		0.813	0.851	0.871	0.647	0.605	0.757	0.040
OTM	Sánchez et al. [81]		0.794	0.848	0.870	0.669	0.594	0.755	0.034
OVM	WN-RandowWalks [14]		0.811	0.797	0.843	0.773	0.543	0.753	0.120
OTM	Resnik [82]	CPRefLeSub-Rat [10]	0.810	0.823	0.874	0.669	0.512	0.738	0.057
OVM	Nasari [49]		0.880	0.791	0.812	0.708	0.489	0.736	0.095
OTM	Meng et al. [83]	Seco et al. [71]	0.796	0.849	0.837	0.613	0.571	0.733	0.023
OTM	Mubaid&Nguyen [84]		0.780	0.807	0.853	0.645	0.576	0.732	0.005
WE	FastText [47]		0.860	0.793	0.818	0.775	0.411	0.731	0.160
WE	GloVe [44]		0.807	0.770	0.759	**0.797**	0.467	0.720	0.097
WE	CBOW [43]		0.807	0.772	0.786	0.763	0.461	0.718	0.073
OTM	Pedersen et al. [85]		0.746	0.781	0.840	0.605	0.551	0.705	0.002
OTM	Garla and Brandt [86]	Sánchez et al. [68]	0.721	0.769	0.847	0.572	0.512	0.684	0.005
OTM	Rada et al. [87]		0.707	0.771	0.751	0.558	0.565	0.670	0.001
OTM	$Wu\&Palmer_{fast}$ [88]		0.630	0.720	0.715	0.568	0.473	0.621	0.000
WEC	SymPatterns-500d [45]		0.648	0.690	0.709	0.454	0.435	0.587	0.000
OVM	WordNet UKB [18]		0.515	0.548	0.629	0.375	0.361	0.486	0.000

evaluate any ontology-based methods by declaring it in any *SimilarityMeasures* node whenever this method have been previously implemented in HESML and its keyname being specified by the *SimilarityMeasureType* enumeration in the "*WordNetBasedExperiments.xsd*" schema file. Likewise, currently supported IC models are specified by the *IntrinsicICModelType* enumeration in the aforementioned XML schema file. On the other hand, the reader could evaluate any unconsidered pre-trained word embedding model by declaring a new method in the *RawWordVectorFiles*, whenever its corresponding pre-trained model being provided in any of the three file formats which are currently supported, otherwise it would be needed to extend HESML to support a new pre-trained word embedding file format. Finally, the reader could define any new benchmark considering a different set of word similarity datasets by declaring further *SingleDatasetSimilarityValuesExperiment* nodes with their corresponding dataset files in comma-separated file format. For a detailed list of the methods currently implemented by HESML V1R4, we refer the readers to its release notes [28].

5. Corrigendum of the mismatch in the MC28 similarity scores

Despite the reproducibility protocol detailed in Section 3 allows to reproduce all the results in our primary work [27] exactly, there was an unintentional and unfortunate mismatch in the preparation of the MC28 [41] benchmark file used in our experiments, which was detected during the review of this work. The MC28 dataset is made up by a subset of the word pairs in the RG65 dataset [52]; however, the similarity scores for the same word pairs are slightly different. Because of a manipulation error, our aforementioned MC28 dataset file, called *Miller_Charles_28_dataset.csv* as detailed in primary work [27,

Table 18
Spearman rank correlation (ρ) values for each Ontology-based semantic Topological similarity Measure (OTM), Ontology-based Vector Model (OVM) or Word Embedding (WE/WEC) model in the five similarity datasets. Measures (rows) are ranked according to their average value shown in Avg column. Best value for each dataset is shown in bold. (*) Last column shows p-values for an one-side t-Student distribution between Attract–repel [50] model and the remaining methods using the performance in the five noun similarity datasets as paired random sample with the aim of testing the hypothesis that Attract–repel significantly outperforms remaining methods in Spearman rank correlation. (**) The values reported in this column correspond to an evaluation of the MC28 dataset with its original similarity scores, which is carried-out in this work to fix a mismatch in the MC28 benchmark file used in our primary paper [27]. Our primary paper reports the values obtained in the evaluation of the MC28 dataset with the similarity scores of the RG65 dataset.

Corrigendum of our primary paper [27, table 5]			Spearman correlation (ρ) in noun similarity datasets						
Family	Measure	IC model	MC28 (**)	RG65	PS_{full}	Agirre201	SL665	Avg (ρ)	p-value(*)
WEC	Attract–repel [50]		0.864	0.825	**0.843**	0.738	0.690	**0.792**	–
WEC	Paragram-ws [46]		0.859	0.813	0.821	**0.808**	0.645	0.789	0.440
WEC	Counter-fitting [72]		0.864	0.808	0.831	0.695	**0.698**	0.779	0.110
WEC	Paragram-sl [46]		0.808	0.775	0.794	0.778	0.679	0.767	0.120
OVM	WN-RandowWalks [14]		**0.882**	0.823	0.814	0.784	0.529	0.766	0.260
OVM	WordNet UKB [18]		0.878	**0.858**	0.841	0.718	0.524	0.763	0.230
OTM	coswJ&C [1]	Sánchez et al. [68]	0.874	0.835	0.822	0.666	0.587	0.757	0.099
OTM	Zhou et al. [70]	Seco et al. [71]	0.832	0.824	0.814	0.655	0.610	0.747	0.024
OTM	$Cai_{strategy2}$ [69]	Cai et al. [69]	0.864	0.804	0.794	0.662	0.595	0.744	0.025
OTM	Pirró&Seco [73]	Seco et al. [71]	0.875	0.801	0.792	0.656	0.586	0.742	0.036
OTM	Meng et al. [83]	Seco et al. [71]	0.793	0.820	0.815	0.655	0.610	0.739	0.014
OTM	cosJ&C [1]	Sánchez et al. [68]	0.847	0.803	0.800	0.650	0.591	0.738	0.017
OTM	Jiang&Conrath [79]	Sánchez et al. [68]	0.847	0.803	0.800	0.650	0.591	0.738	0.017
OTM	Garla and Brandt [86]	Sánchez et al. [68]	0.847	0.803	0.800	0.650	0.591	0.738	0.017
OTM	Gao_{strat3} [74]	CPRefHypo [10]	0.821	0.801	0.791	0.641	0.595	0.730	0.007
OTM	Mubaid&Nguyen [84]		0.807	0.812	0.807	0.645	0.578	0.729	0.013
OTM	Hadj Taieb et al. [4]		0.791	0.797	0.797	0.660	0.596	0.728	0.003
OTM	Meng and Gu [75]	Seco et al. [71]	0.813	0.797	0.791	0.647	0.589	0.727	0.004
OTM	FaITH [76]	Seco et al. [71]	0.813	0.797	0.791	0.647	0.589	0.727	0.004
OTM	Lin [77]	Seco et al. [71]	0.813	0.797	0.791	0.647	0.589	0.727	0.004
WE	FastText [47]		0.843	0.801	0.801	0.777	0.410	0.726	0.150
OTM	Li_{strat3} [78]		0.802	0.810	0.798	0.625	0.588	0.724	0.010
OTM	Pedersen et al. [85]		0.802	0.810	0.798	0.625	0.588	0.724	0.010
OTM	Leacock&Chodorow [80]		0.802	0.810	0.798	0.625	0.588	0.724	0.010
OTM	Rada et al. [87]		0.802	0.810	0.798	0.625	0.588	0.724	0.010
OTM	Sánchez et al. [81]		0.786	0.784	0.789	0.643	0.578	0.716	0.002
WE	GloVe [44]		0.832	0.769	0.755	0.795	0.429	0.716	0.110
WE	CBOW [43]		0.822	0.760	0.767	0.772	0.454	0.715	0.078
OTM	Resnik [82]	CPRefLeSub-Rat [10]	0.851	0.763	0.757	0.638	0.511	0.704	0.016
OVM	Nasari [49]		0.790	0.745	0.752	0.684	0.488	0.692	0.009
OTM	$Wu\&Palmer_{fast}$ [88]		0.578	0.712	0.716	0.600	0.482	0.618	0.003
WEC	SymPatterns-500d [45]		0.670	0.663	0.674	0.483	0.460	0.590	0.000

table 3], contains the similarity scores of the RG65 dataset. Thus, despite the experiments, results, and conclusions of our primary work are valid and fully reproducible, the results reported for the MC28 dataset cannot be directly compared with others reported in the literature. For this reason, we introduce here a corrigendum for those tables in our primary work that report results in the evaluation of the MC28 dataset.

5.1. New evaluation of the MC28 dataset

We have included a new file called *Miller_Charles_28_canonic_dataset.csv* with the original MC28 similarity scores, both in the HESML GitHub repository and the latest HESML version [42]. The MC28 similarity scores are also reported by Bollegala et al. [89] and Yang and Powers [90] among others. We have evaluated again the MC28 dataset by running the reproducibility protocol detailed in the Section 3.4 with a experiment file called *benchmark_MC28_canonic.exp*. This later benchmark file only evaluates the MC28 dataset generating the *raw_similarity_values_MC28_canonic_dataset.csv* file as raw output. All these aforementioned files are provided as supplementary material with this work, being also available at the master branch of the HESML GitHub repository, with the aim of allowing the exact replication of the new results in the evaluation of the MC28 dataset which are reported in this section.

Tables 17, 18, 19, 20, and 21 correct the mismatch in the MC28 results reported in tables 3, 4, 8, A.3 and A.4 of our primary work [27] respectively. Fortunately, MC28 similarity scores are highly correlated with their corresponding RG65 scores and our conclusions were based on a large collection of datasets. For this reason, the new MC28 results only introduce minor changes which do not invalid any significant conclusion reported in our

Table 19
Pearson (r), Spearman (ρ) and Harmonic score metrics obtained by each Word Embedding (WE) or Ontology-based Vector (OVM) model in all similarity datasets. Word embedding (WE/WEC) and ontology-based vector models (columns) are ranked in descending order from left to right according to their average harmonic score shown in last row. Best value for each dataset and metric is shown in bold. (*) The values reported in these rows correspond to an evaluation of the MC28 dataset with its original similarity scores, which is carried-out in this work to fix a mismatch in the MC28 benchmark file used in our primary paper [27]. Our primary paper reports the values obtained in the evaluation of the MC28 dataset with the similarity scores of the RG65 dataset.

Corrigendum of our primary paper [27, table 8]											
Method and family	Attract–repel [50]	Counter-fitting [72]	Paragram-ws [46]	Paragram-sl [46]	WN-Randow Walks [14]	CBOW [43]	Nasari [49]	GloVe [44]	FastText [47]	Sym Patterns -500d [45]	WordNet UKB [18]
	WEC	WEC	WEC	WEC	OVM	WE	OVM	WE	WE	WEC	OVM
Dataset	Pearson (r) correlation values in all similarity datasets										
MC28 (*)	0.847	0.824	0.801	0.799	0.811	0.807	**0.880**	0.807	0.860	0.648	0.515
RG65	**0.840**	0.806	0.810	0.798	0.797	0.772	0.791	0.770	0.793	0.690	0.548
PS_{full}	**0.893**	0.866	0.849	0.854	0.843	0.786	0.812	0.759	0.818	0.709	0.629
Agirre201	0.720	0.701	0.765	0.748	0.773	0.763	0.708	**0.797**	0.775	0.454	0.375
SimLex665	0.691	**0.697**	0.662	0.682	0.543	0.461	0.489	0.467	0.411	0.435	0.361
SimLex111	**0.877**	0.857	0.844	0.815	0.637	0.598	0.498	0.614	0.484	0.700	0.443
SimLex222	**0.777**	0.713	0.574	0.605	0.464	0.349	0.428	0.220	0.247	0.537	0.338
SimLex999	**0.745**	0.728	0.676	0.689	0.532	0.454	0.466	0.437	0.385	0.493	0.370
SimVerb3500	**0.666**	0.613	0.524	0.546	0.549	0.375	0.336	0.294	0.263	0.327	0.387
Avg(r)	**0.784**	0.756	0.723	0.726	0.661	0.596	0.601	0.574	0.560	0.555	0.441
Dataset	Spearman (ρ) correlation values in all similarity datasets										
MC28 (*)	0.864	0.864	0.859	0.808	**0.882**	0.822	0.790	0.832	0.843	0.670	0.878
RG65	0.825	0.808	0.813	0.775	0.823	0.760	0.745	0.769	0.801	0.663	**0.858**
PS_{full}	**0.843**	0.831	0.821	0.794	0.814	0.767	0.752	0.755	0.801	0.674	0.841
Agirre201	0.738	0.695	**0.808**	0.778	0.784	0.772	0.684	0.795	0.777	0.483	0.718
SimLex665	0.690	**0.698**	0.645	0.679	0.529	0.454	0.488	0.429	0.410	0.460	0.524
SimLex111	**0.872**	0.847	0.825	0.795	0.643	0.592	0.473	0.622	0.508	0.676	0.555
SimLex222	**0.783**	0.727	0.562	0.590	0.446	0.322	0.414	0.196	0.231	0.544	0.367
SimLex999	**0.751**	0.736	0.667	0.685	0.525	0.442	0.450	0.408	0.380	0.513	0.497
SimVerb3500	**0.672**	0.628	0.514	0.540	0.545	0.364	0.287	0.283	0.258	0.328	0.499
Avg(ρ)	**0.782**	0.759	0.724	0.716	0.666	0.588	0.565	0.565	0.556	0.557	0.637
Dataset	Harmonic score (h) values in all similarity datasets										
MC28 (*)	**0.856**	0.843	0.829	0.803	0.845	0.814	0.832	0.820	0.852	0.659	0.649
RG65	**0.833**	0.807	0.811	0.786	0.810	0.766	0.767	0.770	0.797	0.676	0.669
PS_{full}	**0.867**	0.848	0.835	0.823	0.828	0.777	0.781	0.757	0.809	0.691	0.720
Agirre201	0.729	0.698	0.786	0.762	0.779	0.767	0.696	**0.796**	0.776	0.468	0.493
SimLex665	0.690	**0.697**	0.653	0.681	0.536	0.457	0.489	0.447	0.411	0.447	0.427
SimLex111	**0.874**	0.852	0.835	0.805	0.640	0.595	0.485	0.618	0.496	0.688	0.493
SimLex222	**0.780**	0.720	0.568	0.597	0.455	0.335	0.421	0.207	0.239	0.540	0.352
SimLex999	**0.748**	0.732	0.671	0.687	0.528	0.448	0.458	0.422	0.383	0.503	0.424
SimVerb3500	**0.669**	0.620	0.519	0.543	0.547	0.369	0.310	0.288	0.261	0.328	0.436
Avg(h)	**0.783**	0.758	0.723	0.721	0.663	0.592	0.582	0.569	0.558	0.556	0.518

primary work. However, we introduce a detailed discussion below to report the minor impact of this new MC28 results on the conclusions reported in our primary work.

Reproducing our corrigendum. You could reproduce this new evaluation of the MC28 dataset by substituting the experiment file detailed in step 2 of the Tables 10 or 11 by the *benchmark_MC28_canonic.exp* file, which is provided as supplementary material. Subsequently, you should edit the post-processing file "*embeddings_vs_ontomeasures_final_tables.R*" detailed in Section 3.6 to substitute the source code in line 32 by the new raw output file as follows: *raw_MC28_file* ← *"raw_similarity_values_MC28_canonic_dataset.csv"*. Finally, you should run the aforementioned post-processing file (see Section 3.6) to generate all corrigendum tables reported herein.

5.2. Impact of the new MC28 results

The new results for the evaluation of the MC28 dataset reported herein invalidate none major conclusion in our primary work. However, they modify some minor conclusions which are enumerated below.

1. *CoswJ&C [27] similarity measure obtains the highest average Pearson correlation value in all similarity datasets among the family of OTM measures instead of Cai et al. [69] measure*. This conclusion can be drawn by looking the average column in Table 17. However, Cai et al. [69] similarity measure obtained this same aforementioned result in our primary work [27, table 4](see Table 18).
2. *Nasari [49] model obtains the highest Pearson correlation value in the MC28 dataset among the family of WE and OVM models instead of GloVe [44]*. This conclusion can be drawn by looking at Table 19. However, GloVe obtained this aforementioned same result in our primary paper [27, table 8].
3. *Attract–repel [50] model obtains the highest harmonic score in the MC28 dataset among the family of WE and OVM models instead of WN-RandomWalks [14]*. This conclusion can be drawn by looking at Table 19. However, WN-RandomWalks obtained this same aforementioned result in our primary paper [27, table 8](see Table 20).
4. *CoswJ&C [27] obtains the highest Spearman correlation value in the MC28 dataset when it is combined with Attract-reppel*. This conclusion can be drawn by looking results in Table 21. However, the coswJ&C [27] and Zhou et al. [70] measures obtained the same Spearman correlation value in our primary work [27, table A.4].

Table 20
Pearson correlation (r) values for the combined measures defined by the arithmetic mean of the similarity values returned by the Attract–repel model and each remaining base measure. Combined measures (rows) are ranked according to their average value shown in Avg column. Best value for each dataset is shown in bold. (*) Last column shows p-values for an one-side t-Student distribution between all combined measures as regard their corresponding base method in Measure column using the performance in the five noun similarity datasets as paired random sample with the aim of testing the hypothesis that each combined measure significantly outperforms base methods in Pearson correlation. (**) The values reported in this column correspond to an evaluation of the MC28 dataset with its original similarity scores, which is carried-out in this work to fix a mismatch in the MC28 benchmark file used in our primary paper [27]. Our primary paper reports the values obtained in the evaluation of the MC28 dataset with the similarity scores of the RG65 dataset.

Corrigendum of our primary paper [27, table A.3]									
Measures combined with Attract–repel model			Pearson correlation (r) in noun similarity datasets						
Fam	Measure	IC model	MC28 (**)	RG65	PS_{full}	Agirre201	SL665	Avg (r)	p-value(*)
OTM	coswJ&C [1]	Sánchez et al. [68]	**0.933**	0.912	0.942	0.784	0.721	**0.858**	0.007
OTM	Zhou et al. [70]	Seco et al. [71]	0.915	0.905	0.946	0.777	**0.739**	0.856	0.005
OTM	$Cai_{strategy2}$ [69]	Cai et al. [69]	0.911	0.902	0.947	0.786	0.736	0.856	0.008
OTM	cosJ&C [1]	Sánchez et al. [68]	0.915	0.910	**0.949**	0.778	0.720	0.854	0.005
OTM	Hadj Taieb et al. [4]		0.894	0.896	0.943	0.782	0.722	0.847	0.006
OTM	Gao_{strat3} [74]	CPRefHypo [10]	0.899	0.899	0.939	0.767	0.732	0.847	0.004
OTM	Meng et al. [83]	Seco et al. [71]	0.908	**0.915**	0.933	0.749	0.723	0.846	0.001
OTM	Lin [77]	Seco et al. [71]	0.890	0.898	0.942	0.774	0.724	0.845	0.004
OTM	Li_{strat3} [78]		0.902	0.895	0.934	0.763	0.728	0.845	0.005
OTM	Sánchez et al. [81]		0.888	0.896	0.935	0.767	0.731	0.843	0.002
OTM	FaITH [76]	Seco et al. [71]	0.877	0.888	0.940	0.774	0.718	0.839	0.006
OTM	Meng and Gu [75]	Seco et al. [71]	0.869	0.889	0.937	0.763	0.698	0.831	0.004
OTM	Pirró&Seco [73]	Seco et al. [71]	0.891	0.892	0.932	0.744	0.679	0.828	0.003
WE	GloVe [44]		0.866	0.855	0.883	**0.817**	0.691	0.823	0.021
OVM	WN-RandowWalks [14]		0.863	0.848	0.899	0.791	0.709	0.822	0.027
OVM	Nasari [49]		0.901	0.855	0.893	0.780	0.679	0.822	0.019
WE	Fastext [47]		0.881	0.858	0.902	0.793	0.673	0.821	0.058
WE	CBOW [43]		0.863	0.851	0.890	0.799	0.679	0.817	0.018
OTM	Leacock&Chodorow [80]		0.872	0.883	0.913	0.722	0.690	0.816	0.002
WEC	Paragram-ws [46]		0.837	0.844	0.890	0.770	0.716	0.811	0.007
OTM	Pedersen et al. [85]		0.856	0.848	0.907	0.732	0.700	0.809	0.002
WEC	Paragram-sl [46]		0.833	0.833	0.888	0.758	0.721	0.807	0.002
WEC	Counter-fitting [72]		0.845	0.836	0.894	0.730	0.722	0.805	0.000
OTM	WuPalmerFast		0.830	0.864	0.890	0.732	0.691	0.802	0.000
WEC	Attract–repel [50]		0.847	0.840	0.893	0.720	0.691	0.798	—
WEC	SymPatterns-500d [45]		0.843	0.850	0.894	0.713	0.691	0.798	0.000
OTM	Mubaid&Nguyen [84]		0.844	0.846	0.895	0.719	0.668	0.794	0.002
OTM	Garla and Brandt [86]	Sánchez et al. [68]	0.835	0.838	0.907	0.712	0.671	0.793	0.002
OTM	Jiang&Conrath [79]	Sánchez et al. [68]	0.872	0.872	0.889	0.670	0.606	0.782	0.001
OVM	WordNet UKB [18]		0.787	0.796	0.868	0.663	0.666	0.756	0.000
OTM	Resnik [82]	CPRefLeSub-Rat [10]	0.821	0.831	0.883	0.683	0.535	0.750	0.004
OTM	Rada et al. [87]		0.758	0.810	0.799	0.610	0.623	0.720	0.000

6. Conclusions and future work

This work introduces, for the first time, a large set of reproducible experiments on word similarity and relatedness including most methods in the families of ontology-based semantic similarity measures based on WordNet and word embedding models. This aforementioned set of experiments allow that all the experiments and results introduced by Lastra-Díaz et al. [27] to be reproduced exactly. Likewise, our reproducible experiments could be easily extended, or modified, to create new benchmarks from scratch, which evaluate a different set of methods and word similarity and relatedness datasets from those considered herein. For this reason, we hope that this set of reproducible benchmarks to become into a de facto standard experimentation platform for any future research on word similarity and relatedness.

Finally, this work introduces a corrigendum for a mismatch in the MC28 similarity scores used in the experiments in our primary work, which was detected during the review of this work. We accidentally included the RG65 similarity scores in the MC28 benchmark file, despite the MC28 similarity scores are slightly different. In order to bridge this gap, this work introduces an updated version of those data tables which report MC28 results in our primary work. Because the MC28 dataset is highly correlated with the RG65 dataset and our conclusions were based on a large collection of datasets, the new MC28 results impact none major conclusion of our primary work. However, we report the changes in four minor conclusions.

As forthcoming activities, we plan the study and proposal of new distributional similarity and relatedness measures, as well as their use in the definition of sentence and short-text similarity measures.

Table 21
Spearman rank correlation (ρ) values for the combined measures defined by the arithmetic mean of the similarity values returned by the Attract–repel model and each remaining base measure. Combined measures (rows) are ranked according to their average value shown in Avg column. Best value for each dataset is shown in bold. (*) Last column shows p-values for an one-side t-Student distribution between all combined measures as regard their corresponding base method in Measure column using the performance in the five noun similarity datasets as paired random sample with the aim of testing the hypothesis that each combined measure significantly outperforms base methods in Spearman correlation. (**) The values reported in this column correspond to an evaluation of the MC28 dataset with its original similarity scores, which is carried-out in this work to fix a mismatch in the MC28 benchmark file used in our primary paper [27]. Our primary paper reports the values obtained in the evaluation of the MC28 dataset with the similarity scores of the RG65 dataset.

Corrigendum of our primary paper [27, table A.4]									
Measures combined with Attract–repel model			Spearman correlation (ρ) in noun similarity datasets						
Fam	Measure	IC model	MC28 (**)	RG65	PS_{full}	Agirre201	SL665	Avg (ρ)	p-value(*)
OTM	$Cai_{strategy2}$ [69]	Cai et al. [69]	0.912	0.900	**0.903**	0.774	0.727	**0.843**	0.001
OTM	Zhou et al. [70]	Seco et al. [71]	0.912	**0.902**	0.902	0.761	0.731	0.842	0.000
OTM	coswJ&C [1]	Sánchez et al. [68]	**0.927**	0.887	0.881	0.755	0.702	0.830	0.002
OVM	WN-RandowWalks [14]		0.898	0.854	0.858	0.832	0.703	0.829	0.046
OTM	cosJ&C [1]	Sánchez et al. [68]	0.909	0.895	0.889	0.746	0.702	0.828	0.000
OTM	Pedersen et al. [85]		0.893	0.869	0.878	0.763	**0.734**	0.827	0.002
OTM	Hadj Taieb et al. [4]		0.901	0.875	0.877	0.759	0.710	0.824	0.000
OTM	Li_{strat3} [78]		0.887	0.878	0.878	0.744	0.719	0.821	0.001
WEC	Paragram-ws [46]		0.887	0.837	0.851	0.811	0.708	0.819	0.018
OTM	Garla and Brandt [86]	Sánchez et al. [68]	0.882	0.854	0.869	0.764	0.722	0.818	0.006
OTM	FaITH [76]	Seco et al. [71]	0.882	0.870	0.870	0.759	0.709	0.818	0.000
WE	GloVe [44]		0.876	0.856	0.859	**0.835**	0.662	0.818	0.022
WE	CBOW [43]		0.884	0.853	0.864	0.817	0.664	0.816	0.012
OTM	Meng et al. [83]	Seco et al. [71]	0.891	0.876	0.870	0.733	0.710	0.816	0.001
OTM	Gao_{strat3} [74]	CPRefHypo [10]	0.890	0.868	0.863	0.738	0.715	0.815	0.001
WEC	Counter-fitting [72]		0.879	0.843	0.863	0.748	0.726	0.812	0.003
OTM	Lin [77]	Seco et al. [71]	0.874	0.864	0.861	0.746	0.706	0.810	0.001
WE	Fastext [47]		0.888	0.846	0.854	0.807	0.653	0.809	0.053
OVM	WordNet UKB [18]		0.889	0.839	0.853	0.759	0.698	0.808	0.130
WEC	Paragram-sl [46]		0.852	0.825	0.844	0.792	0.723	0.807	0.002
OTM	Mubaid&Nguyen [84]		0.882	0.866	0.865	0.721	0.684	0.804	0.001
OTM	Sánchez et al. [81]		0.848	0.849	0.852	0.743	0.714	0.801	0.002
OTM	Meng and Gu [75]	Seco et al. [71]	0.868	0.858	0.853	0.729	0.683	0.798	0.000
OVM	Nasari [49]		0.868	0.831	0.836	0.780	0.673	0.798	0.003
OTM	Pirró&Seco [73]	Seco et al. [71]	0.895	0.852	0.848	0.717	0.661	0.795	0.002
OTM	Leacock&Chodorow [80]		0.881	0.861	0.852	0.697	0.678	0.794	0.000
WEC	Attract–repel [50]		0.864	0.825	0.843	0.738	0.690	0.792	
WEC	SymPatterns-500d [45]		0.814	0.805	0.821	0.730	0.688	0.772	0.001
OTM	$Wu\&Palmer_{fast}$ [88]		0.805	0.820	0.829	0.721	0.671	0.769	0.002
OTM	Rada et al. [87]		0.845	0.834	0.825	0.665	0.645	0.763	0.001
OTM	Jiang&Conrath [79]	Sánchez et al. [68]	0.874	0.826	0.824	0.671	0.612	0.761	0.000
OTM	Resnik [82]	CPRefLeSub-Rat [10]	0.863	0.783	0.775	0.651	0.531	0.720	0.000

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We are grateful of Fernando González and Juan Corrales for setting up our UNED Dataverse dataset, Alicia Lara-Clares for developing the HESML web site, Yuanyuan Cai for answering kindly our questions to replicate their IC-based similarity measures and IC models in HESML, and http://clouding.io for their technical support to set up our experimental platform. We are also very thankful to José Camacho-Collados for providing the weighting overlap source code which we have integrated into HESML for measuring the similarity between the NASARI vectors. Finally, we are sincerely grateful of the reviewer for his valuable comments to improve the quality of the paper and his careful and rigorous review.

This work has been partially supported by the Spanish project VEMODALEN (TIN2015-71785-R), the Basque Government (type A IT1343-19), BBVA BigKnowledge bigknowledge project, and the Spanish Research Agency LIHLITH project (PCIN-2017-118/AEI) in the framework of EU ERA-Net CHIST-ERA.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.is.2020.101636.

References

[1] J.J. Lastra-Díaz, A. García-Serrano, A novel family of IC-based similarity measures with a detailed experimental survey on WordNet, Eng. App. Artif. Intell. 46 (2015) 140–153.

[2] M.A. Hadj Taieb, M. Ben Aouicha, M. Tmar, A. Ben Hamadou, New WordNet-based semantic relatedness measurement, in: GWC 2012 6th International Global Wordnet Conference, academia.edu, 2012, p. 126.

[3] M.A. Hadj Taieb, M. Ben Aouicha, Y. Bourouis, FM3S: Features-based measure of sentences semantic similarity, in: E. Onieva, I. Santos, E. Osaba, H. Quintián, E. Corchado (Eds.), Proceedings of the 10th International Conference on Hybrid Artificial Intelligent Systems, HAIS 2015, in: LNCS, vol. 9121, Springer, Bilbao, Spain, 2015, pp. 515–529.

[4] M.A. Hadj Taieb, M. Ben Aouicha, A. Ben Hamadou, Ontology-based approach for measuring semantic similarity, Eng. Appl. Artif. Intell. 36 (2014) 238–261.

[5] M.A. Hadj Taieb, M. Ben Aouicha, A. Ben Hamadou, A new semantic relatedness measurement using WordNet features, Knowl. Inf. Syst. 41 (2) (2014) 467–497.

[6] M. Ben Aouicha, M.A. Hadj Taieb, Computing semantic similarity between biomedical concepts using new information content approach, J. Biomed. Inform. 59 (2016) 258–275.

[7] M. Ben Aouicha, M. Hadj Taieb, G2WS: Gloss-based WordNet and Wiktionary semantic Similarity measure, in: 2015 IEEE/ACS 12th International Conference of Computer Systems and Applications, AICCSA, 2015, pp. 1–7.

[8] M. Ben Aouicha, M.A. Hadj Taieb, A. Ben Hamadou, Taxonomy-based information content and wordnet-wiktionary-wikipedia glosses for semantic relatedness, Appl. Intell. (2016) 1–37.

[9] J.J. Lastra-Díaz, A. García-Serrano, A new family of information content models with an experimental survey on WordNet, Knowl.-Based Syst. 89 (2015) 509–526.

[10] J.J. Lastra-Díaz, A. García-Serrano, A Refinement of the Well-Founded Information Content Models with a Very Detailed Experimental Survey on WordNet, Technical Report, (TR-2016-01) UNED, 2016, http://e-spacio.uned.es/fez/view/bibliuned:DptoLSI-ETSI-Informes-Jlastra-refinement.

[11] M. Hadj Taieb, M. Ben Aouicha, M. Tmar, A. Hamadou, New information content metric and nominalization relation for a new WordNet-based method to measure the semantic relatedness, in: Cybernetic Intelligent Systems (CIS), 2011 IEEE 10th International Conference on, 2011, pp. 51–58.

[12] M.A. Hadj Taieb, M. Ben Aouicha, M. Tmar, A. Ben Hamadou, Wikipedia category graph and new intrinsic information content metric for word semantic relatedness measuring, in: Proc. of the Third International Conference on Data and Knowledge Engineering, ICDKE, in: Lecture Notes in Computer Science, Springer Berlin Heidelberg, Wuyishan, Fujian, China, 2012, pp. 128–140.

[13] J. Goikoetxea, E. Agirre, A. Soroa, Exploring the use of word embeddings and random walks on wikipedia for the cogalex shared task, in: Proc. of the 4th Workshop on Cognitive Aspects of the Lexicon, 2014, pp. 31–34.

[14] J. Goikoetxea, A. Soroa, E. Agirre, Random walks and neural network language models on knowledge bases, in: Proc. of HLT-NAACL, 2015, pp. 1434–1439.

[15] J. Goikoetxea, E. Agirre, A. Soroa, Single or multiple? Combining word representations independently learned from text and wordnet, in: Proc. of AAAI, 2016, pp. 2608–2614.

[16] J. Goikoetxea, A. Soroa, E. Agirre, Bilingual embeddings with random walks over multilingual wordnets, Knowl.-Based Syst. 150 (15) (2018) 218–230.

[17] E. Yeh, D. Ramage, C.D. Manning, E. Agirre, A. Soroa, Wikiwalk: Random walks on wikipedia for semantic relatedness, in: Proc. of the 2009 Workshop on Graph-Based Methods for Natural Language Processing, in: TextGraphs-4, ACL, Stroudsburg, PA, USA, 2009, pp. 41–49.

[18] E. Agirre, E. Alfonseca, K. Hall, J. Kravalova, M. Paşca, A. Soroa, A study on similarity and relatedness using distributional and wordnet-based approaches, in: Proc. of Human Language Technologies: The 2009 Annual Conf. of the North American Chapter of the Association for Computational Linguistics, NAACL '09, ACL, Stroudsburg, PA, USA, 2009, pp. 19–27.

[19] M. Ben Aouicha, M. Hadj Taieb, S. Beyaoui, Distributional semantics study using the co-occurrence computed from collaborative resources and WordNet, in: 2016 International Symposium on INnovations in Intelligent SysTems and Applications, INISTA, 2016, pp. 1–8.

[20] J.J. Lastra-Díaz, A. García-Serrano, M. Batet, M. Fernández, F. Chirigati, HESML: a scalable ontology-based semantic similarity measures library with a set of reproducible experiments and a replication dataset, Inf. Syst. 66 (2017) 97–118.

[21] M. Ben Aouicha, M.A. Hadj Taieb, A. Ben Hamadou, SISR: System for integrating semantic relatedness and similarity measures, Soft Comput. (2016) 1–25.

[22] J.J. Lastra-Díaz, A. García-Serrano, WNSimRep: a framework and replication dataset for ontology-based semantic similarity measures and information content models, 2016, http://dx.doi.org/10.17632/mpr2m8pycs.1, Mendeley Data v1.

[23] J.J. Lastra-Díaz, J. Goikoetxea, M.A. Hadj Taieb, A. García-Serrano, M. Ben Aouicha, E. Agirre, Reproducibility dataset for a large experimental survey on word embeddings and ontology-based methods for word similarity, Data Brief (2019).

[24] http://e-spacio.uned.es/fez/view/tesisuned:ED-Pg-SisInt-Jjlastra J.J. Lastra-Díaz, in: A. García-Serrrano (Ed.), Recent Advances in Ontology-based Semantic Similarity Measures and Information Content Models based on WordNet (Ph.D. thesis), Universidad Nacional de Educación a Distancia (UNED), 2017.

[25] J.J. Lastra-Díaz, A. García-Serrano, WordNet-based word similarity reproducible experiments based on HESML V1R1 and ReproZip, 2016, http://dx.doi.org/10.17632/65pxgskhz9.1, Mendeley Data, v1.

[26] J.J. Lastra-Díaz, A. García-Serrano, HESML_vs_SML: scalability and performance benchmarks between the HESML V1R2 and SML 0.9 semantic measures libraries, 2016, http://dx.doi.org/10.17632/5hg3z85wf4.1, Mendeley Data, v1.

[27] J.J. Lastra-Díaz, J. Goikoetxea, M.A. Hadj Taieb, A. García-Serrano, M. Ben Aouicha, E. Agirre, A reproducible survey on word embeddings and ontology-based methods for word similarity: linear combinations outperform the state of the art, Eng. Appl. Artif. Intell. 85 (2019) 645–665.

[28] J.J. Lastra-Díaz, A. García Serrano, HESML V1R4 Java Software library of ontology-based semantic similarity measures and information content models, 2018, http://dx.doi.org/10.17632/t87s78dg78.4, Mendeley Data, v4.

[29] G. Miller, WordNet: A lexical database for english, Commun. ACM 38 (11) (1995) 39–41.

[30] F. Chirigati, R. Rampin, D. Shasha, J. Freire, ReproZip: computational reproducibility with ease, in: Proc. of the ACM Intl. Conf. on Management of Data, SIGMOD, Vol. 16, 2016, pp. 2085–2088.

[31] F. Chirigati, R. Capone, R. Rampin, J. Freire, D. Shasha, A collaborative approach to computational reproducibility, Inf. Syst. 59 (2016) 95–97.

[32] A. Wolke, M. Bichler, F. Chirigati, V. Steeves, Reproducible experiments on dynamic resource allocation in cloud data centers, Inf. Syst. 59 (2016) 98–101.

[33] A. Fariña, M.A. Martínez-Prieto, F. Claude, G. Navarro, J.J. Lastra-Díaz, N. Prezza, D. Seco, On the reproducibility of experiments of indexing repetitive document collections, Inf. Syst. 83 (2019) 181–194.

[34] T. Pedersen, Empiricism is not a matter of faith, Comput. Linguist. 34 (3) (2008) 465–470.

[35] A. Fokkens, M. Van Erp, M. Postma, T. Pedersen, P. Vossen, N. Freire, Offspring from reproduction problems: What replication failure teaches us, in: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL, Sofia, Bulgaria, 2013, pp. 1691–1701.

[36] A. Branco, K.B. Cohen, P. Vossen, N. Ide, N. Calzolari, Replicability and reproducibility of research results for human language technology: introducing an LRE special section, Lang. Resour. Eval. 51 (1) (2017) 1–5.

[37] M. Wieling, J. Rawee, G. van Noord, Reproducibility in computational linguistics: Are we willing to share? Comput. Linguist. 44 (4) (2018) 641–649.

[38] M.R. Munafò, B.A. Nosek, D.V. Bishop, K.S. Button, C.D. Chambers, N.P. du Sert, U. Simonsohn, E.-J. Wagenmakers, J.J. Ware, J.P.A. Ioannidis, A manifesto for reproducible science, Nat. Hum. Behav. 1 (2017) 0021.

[39] J.J. Lastra-Díaz, J. Goikoetxea, M.A. Hadj Taieb, A. García-Serrano, M. Ben Aouicha, E. Agirre, Word Similarity Benchmarks of Recent Word Embedding Models and Ontology-Based Semantic Similarity Measures, e-cienciaDatos, 2019, http://dx.doi.org/10.21950/AQ1CVX, e-cienciaDatos, v1.

[40] S. Harispe, S. Ranwez, S. Janaqi, J. Montmain, The semantic measures library and toolkit: fast computation of semantic similarity and relatedness using biomedical ontologies, Bioinformatics 30 (5) (2014) 740–742.

[41] G.A. Miller, W.G. Charles, Contextual correlates of semantic similarity, Lang. Cogn. Process. 6 (1) (1991) 1–28.

[42] J.J. Lastra-Díaz, A. Lara-Clares, A. García-Serrano, HESML V1R5 Java Software library of ontology-based semantic similarity measures and information content models, 2020, http://dx.doi.org/10.21950/1RRAWJ, e-cienciaDatos, v1.

[43] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, 2013, arXiv:1301.3781.

[44] J. Pennington, R. Socher, C.D. Manning, Glove: Global vectors for word representation, Proc. EMNLP 12 (2014) 1532–1543.

[45] R. Schwartz, R. Reichart, A. Rappoport, Symmetric pattern based word embeddings for improved word similarity prediction, in: Proc. of the Conf. on Computational Natural Language Learning, 2015, pp. 258–267.

[46] J. Wieting, M. Bansal, K. Gimpel, K. Livescu, D. Roth, From paraphrase database to compositional paraphrase model and back, Trans. ACL 3 (2015) 345–358.
[47] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriching word vectors with subword information, 2016, arXiv:1607.04606.
[48] E. Agirre, A. Soroa, Personalizing pagerank for word sense disambiguation, in: Proc. of the 12th Conf. of the European Chapter of the Association for Computational Linguistics, ACL, 2009, pp. 33–41.
[49] J. Camacho-Collados, M.T. Pilehvar, R. Navigli, Nasari: Integrating explicit knowledge and corpus statistics for a multilingual representation of concepts and entities, Artificial Intelligence 240 (2016) 36–64.
[50] N. Mrkšić, I. Vulić, D.Ó. Séaghdha, I. Leviant, R. Reichart, M. Gašić, A. Korhonen, S. Young, Semantic specialisation of distributional word vector spaces using monolingual and cross-lingual constraints, Trans. ACL 5 (2017) 309–324.
[51] D. Merkel, Docker: Lightweight linux containers for consistent development and deployment, Linux J. 2014 (239) (2014) Article No. 2.
[52] H. Rubenstein, J.B. Goodenough, Contextual correlates of synonymy, Commun. ACM 8 (10) (1965) 627–633.
[53] G. Pirró, A semantic similarity metric combining features and intrinsic information content, Data Knowl. Eng. 68 (11) (2009) 1289–1308.
[54] F. Hill, R. Reichart, A. Korhonen, SimLex-999: Evaluating semantic models with (genuine) similarity estimation, Comput. Linguist. 41 (4) (2015) 665–695.
[55] G. Halawi, G. Dror, E. Gabrilovich, Y. Koren, Large-scale learning of word relatedness with constraints, in: Proc. of ACM SIGKDD, ACM, New York, NY, USA, 2012, pp. 1406–1414.
[56] K. Radinsky, E. Agichtein, E. Gabrilovich, S. Markovitch, A word at a time: computing word relatedness using temporal semantic analysis, in: Proc. of the Intl. Conf. on WWW, ACM, 2011, pp. 337–346.
[57] L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, E. Ruppin., Placing search in context: the concept revisited, ACM Trans. Inf. Syst. 20 (1) (2002) 116–131.
[58] S.R. Szumlanski, F. Gomez, V.K. Sims, A new set of norms for semantic relatedness measures, in: Proc. of the 51st Annual Meeting of the Association for Computational Linguistics, Vol. 2, ACL'2013, aclweb.org, Sofia, Bulgaria, 2013, pp. 890–895.
[59] D. Gerz, I. Vulić, F. Hill, R. Reichart, A. Korhonen, SimVerb-3500: A large-scale evaluation set of verb similarity, in: Proc. of EMNLP, Austin, Texas, 2016, pp. 2173–2182.
[60] E. Bruni, N.-K. Tran, M. Baroni, Multimodal distributional semantics, J. Artificial Intelligence Res. 49 (1) (2014) 1–47.
[61] D. Yang, D.M. Powers, Verb similarity on the taxonomy of WordNet, in: Proc. of the 3th Intl. WordNet Conf., GWC, Masaryk University, 2006, pp. 121–128.
[62] T. Luong, R. Socher, C.D. Manning, Better word representations with recursive neural networks for morphology, in: Proc. of CoNLL, 2013, pp. 104–113.
[63] E.H. Huang, R. Socher, C.D. Manning, A.Y. Ng, Improving word representations via global context and multiple word prototypes, in: Proc. of the Annual Meeting of the ACL, Vol. 1, 2012, pp. 873–882.
[64] O. Sefraoui, M. Aissaoui, M. Eleuldj, OpenStack: toward an open-source solution for cloud computing, Int. J. Comput. Appl. Technol. 55 (3) (2012) 38–42.
[65] T. Pedersen, WordNet-infocontent-3.0.tar dataset repository, 2008, https://www.researchgate.net/publication/273885902_WordNet-InfoContent-3.0.tar.
[66] T. Pedersen, Measuring the Similarity and Relatedness of Concepts: a MICAI 2013 Tutorial, Mexico City, Mexico, 2013.
[67] F. Hoffgaard, P. Weil, K. Hamacher, BioPhysConnectoR: Connecting sequence information and biophysical models, BMC Bioinf. 11 (2010) 199.
[68] D. Sánchez, M. Batet, D. Isern, Ontology-based information content computation, Knowl.-Based Syst. 24 (2) (2011) 297–303.
[69] Y. Cai, Q. Zhang, W. Lu, X. Che, A hybrid approach for measuring semantic similarity based on IC-weighted path distance in WordNet, J. Intell. Inf. Syst. (2017) 1–25.
[70] Z. Zhou, Y. Wang, J. Gu, New model of semantic similarity measuring in WordNet, in: Proc. of the 3rd Intl. Conf. on Intelligent System and Knowledge Engineering, Vol. 1, IEEE, 2008, pp. 256–261.
[71] N. Seco, T. Veale, J. Hayes, An intrinsic information content metric for semantic similarity in WordNet, in: Proc. of ECAI, Vol. 16, IOS Press, Valencia, Spain, 2004, pp. 1089–1094.
[72] N. Mrkšić, D. Ó Séaghdha, B. Thomson, M. Gašić, L. Rojas-Barahona, P.-H. Su, D. Vandyke, T.-H. Wen, S. Young, Counter-fitting Word Vectors to Linguistic Constraints, in: Proc. of HLT-NAACL, 2016.
[73] G. Pirró, N. 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, in: LNCS, vol. 5332, Springer, 2008, pp. 1271–1288.
[74] J.B. Gao, B.W. Zhang, X.H. Chen, A wordnet-based semantic similarity measurement combining edge-counting and information content theory, Eng. App. Artif. Intell. 39 (2015) 80–88.
[75] L. Meng, J. Gu, A new model for measuring word sense similarity in WordNet, in: Proc. of the 4th Intl. Conf. on Advanced Communication and Networking, ASTL, Vol. 14, 2012, pp. 18–23.
[76] G. Pirró, J. Euzenat, A feature and information theoretic framework for semantic similarity and relatedness, in: Proc. of ISWC, in: LNCS, vol. 6496, Springer, Shangai, China, 2010, pp. 615–630.
[77] D. Lin, An information-theoretic definition of similarity, in: Proc. of ICML, Vol. 98, Madison, WI, 1998, pp. 296–304.
[78] Y. Li, Z. Bandar, D. McLean, An approach for measuring semantic similarity between words using multiple information sources, IEEE Trans. Knowl. Data Eng. 15 (4) (2003) 871–882.
[79] J.J. Jiang, D.W. Conrath, Semantic similarity based on corpus statistics and lexical taxonomy, in: Proc. of Intl. Conf. Research on Computational Linguistics, ROCLING X, 1997, pp. 19–33.
[80] C. Leacock, M. Chodorow, Combining local context and WordNet similarity for word sense identification, in: WordNet: An Electronic Lexical Database, MIT Press, 1998, pp. 265–283.
[81] D. Sánchez, M. Batet, D. Isern, A. Valls, Ontology-based semantic similarity: A new feature-based approach, Expert Syst. Appl. 39 (9) (2012) 7718–7728.
[82] P. Resnik, Using information content to evaluate semantic similarity in a taxonomy, in: Proc. of IJCAI, Vol. 1, 1995, pp. 448–453.
[83] L. Meng, R. Huang, J. Gu, Measuring semantic similarity of word pairs using path and information content, Intl. J. Fut. Gener. Commun. Netw. 7 (3) (2014) 183–194.
[84] H. Al-Mubaid, H. Nguyen, Measuring semantic similarity between biomedical concepts within multiple ontologies, IEEE Trans. Syst. Man Cybern. 39 (4) (2009) 389–398.
[85] T. Pedersen, S.V.S. Pakhomov, S. Patwardhan, C.G. Chute, Measures of semantic similarity and relatedness in the biomedical domain, J. Biomed. Inform. 40 (3) (2007) 288–299.
[86] V.N. Garla, C. Brandt, Semantic similarity in the biomedical domain: an evaluation across knowledge sources, BMC Bioinf. 13 (2012) 261.
[87] R. Rada, H. Mili, E. Bicknell, M. Blettner, Development and application of a metric on semantic nets, IEEE Trans. Syst. Man Cybern. 19 (1) (1989) 17–30.
[88] Z. Wu, M. Palmer, Verbs semantics and lexical selection, in: Proc. of the Annual Meeting of ACL, ACL, 1994, pp. 133–138.
[89] D. Bollegala, Y. Matsuo, M. Ishizuka, Measuring semantic similarity between words using web search engines, in: Proc. of the 16th International Conference on World Wide Web (WWW'07), WWW '07, ACM, New York, NY, USA, 2007, pp. 757–766.
[90] D. Yang, D.M.W. Powers, Measuring semantic similarity in the taxonomy of WordNet, in: Proc. of the Twenty-Eighth Australasian Conference on Computer Science, Vol. 38, ACSC '05, Australian Computer Society, Inc., Darlinghurst, Australia, Australia, 2005, pp. 315–322.