How to install and use NaturalOWL v.2.0

Gerasimos Lampouras and Ion Androutsopoulos
Natural Language Processing Group, Department of Informatics,
Athens University of Economics and Business, Greece

November 2013

1. Introduction

This document accompanies NaturalOWL version 2.0.[1] NaturalOWL v.2.0 is an open-source natural language generation engine written in Java. It produces English and Greek descriptions of individuals (e.g., products or museum exhibits) and classes (e.g., types of products) from OWL 2.0 ontologies. The ontologies can be optionally associated with linguistic and user modeling resources, also represented in OWL. NaturalOWL can be optionally used within the Protégé editor.[2]

NaturalOWL v.2.0 is described in the following article:

- I. Androutsopoulos, G. Lampouras and D. Galanis, "Generating Natural Language Descriptions from OWL Ontologies: the NaturalOWL System". *Journal of Artificial Intelligence Research*, 48:671-715, 2013.

A more detailed description of the system can be found in the following report:[3]

- I. Androutsopoulos, G. Lampouras and D. Galanis, *Generating Natural Language Descriptions from OWL Ontologies: A Detailed Presentation of the NaturalOWL System*. Technical Report, Natural Language Processing Group, Department of Informatics, Athens University of Economics and Business, 2012.

Apart from the pipeline architecture discussed in the documents above, NaturalOWL v.2.0 implements the alternative, global optimization architecture of the following articles:

- G. Lampouras and I. Androutsopoulos, "Using Integer Linear Programming in Concept-to-Text Generation to Produce More Compact Texts". Proceedings of the *51st Annual Meeting of the Association for Computational Linguistics* (ACL 2013), Sofia, Bulgaria, pp. 561-566 (short papers), 2013.
- G. Lampouras and I. Androutsopoulos, "Using Integer Linear Programming for Content Selection, Lexicalization, and Aggregation to Produce Compact Texts from OWL Ontologies". Proceedings of the *14th European Workshop on Natural Language Generation* (ENLG 2013), at the *51st Annual Meeting of the Association for Computational Linguistics* (ACL 2013), Sofia, Bulgaria, pp. 51-60, 2013.

NaturalOWL was initially developed in the Department of Informatics of the Athens University of Economics and Business (AUEB) during the Greek project XENIOS and was extended during the European project INDIGO.[4] NaturalOWL is heavily based on ideas from

[1] NaturalOWL v.2.0 is available from http://nlp.cs.aueb.gr/software.html.
[2] See http://protege.stanford.edu/.
[3] Available from http://nlp.cs.aueb.gr/pubs/, along with the other documents mentioned here.
[4] XENIOS was co-funded by the European Union and the Greek Secretariat of Research and Technology; see http://www.ics.forth.gr/xenios/. INDIGO was an FP6 IST project of the European Union; see http://www.ics.forth.gr/indigo/.

the European project M-PIRO, which was in turn based on Edinburgh's ILEX system.[5] Version 2.0 of NaturalOWL was developed at AUEB in the PhD thesis of G. Lampouras.[6]

In the rest of this document, we assume that you have already read the articles above.

NaturalOWL v.2.0 is released with a GNU General Public License (GPL); please consult the file GPL.txt for more information. Please note that NaturalOWL v.2.0 is a research prototype. It is provided with absolutely no guarantee and absolutely no support.[7]

2. How to install and use NaturalOWL v.2.0 in Protégé

To install NaturalOWL v.2.0 as a Protégé plug-in, follow the following steps:

Step 1: Installing a Java Virtual Machine

You need JDK 6 (or later) or JRE 6 (or later) installed on your computer to use NaturalOWL v.2.0; they can be downloaded from http://www.oracle.com/technetwork/java/. To check which version of JDK or JRE is currently installed on your computer, type the following in a command-line shell:

```
java -version
```

If this command reports version number 1.6 or later, you can move on to step 2 without installing a new JDK or JRE. Otherwise, you have to install a newer JDK or JRE.

Step 2: Installing Protégé

NaturalOWL v.2.0 was developed and tested with Protégé version 4.1, but it may work with later Protégé versions too. The following installation instructions are for Protégé 4.1. They may have to be modified for later Protégé versions.

Important: Versions 4.1 and 4.2 of Protégé currently have some compatibility issues with version 7 of Java. If you wish to use JDK 7 or JRE 7 with Protégé, we advise you to install Protégé 4.3.

Download Protégé from: http://protege.stanford.edu/. Select "*platform independent installer program*". *Do not* select an installer that includes a "Java VM", because this may install an older version of the Java Virtual Machine (VM) than the version required by NaturalOWL v.2.0.

After downloading Protégé, follow the installation instructions on the same page. You will be asked to specify an installation directory. Write down the installation directory you specified. When prompted to select a Java VM, select the Java VM you want to use. If the Java VM you want to use is not in the list, click "Choose Java Executable" and use the file browsing window that will appear to specify the path to the Java VM to be used; in Windows, it should be something like:

```
C:\Program Files\Java\jdk1.6.0_01\bin\java.exe
```

[5] M-PIRO was an FP5 IST project of the European Union; see http://www.ltg.ed.ac.uk/mpiro/.

[6] Co-financed by the European Union (European Social Fund) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework – Research Funding Program: Heracleitus II, Investing in Knowledge Society through the European Social Fund.

[7] However, please send bug reports to gerasimos.lampouras@gmail.com.

Step 3: Installing NaturalOWL v.2.0 as a Protégé plug-in

Once the installation of Protégé has been completed, go to the installation directory of step 2 and find the "plugins" sub-directory. Copy "gr.aueb.cs.nlg.NLOwlPlugin.jar" from NaturalOWL's distribution to "plugins".

Step 4 (optional): Installing the GNU Linear Programming Kit

If you want to use the optional global optimization architecture, you should first install the GNU Linear Programming Kit (GLPK). NaturalOWL v.2.0 was developed and tested with GLPK version 4.48, but it may work with later GLPK versions too.

If you are installing on a Windows system, you can download a precompiled version of GLPK from http://sourceforge.net/projects/winglpk/files/winglpk/. After you download and extract the distribution, go to the "w32" sub-directory if you are using a 32-bit version of Windows, or the "w64" sub-directory if you are using a 64-bit version of Windows. You need to copy the files "glpk_4_48.dll" and "glpk_4_48_java.dll"to the "system32" sub-directory of your Windows installation; it should be something like:

```
C:\Windows\System32\
```

Important: if you are running the 32-bit version of Protégé in a 64-bit version of Windows, you will have to copy the .dll files of the "w32" sub-directory to the "SysWOW64" sub-directory of your Windows installation instead.

If you are installing on a Linux system, you can follow the instructions in http://en.wikibooks.org/wiki/GLPK/Linux_OS to install GLPK.

If you are installing on a Mac system, you first need to download the GLPK distribution from http://www.gnu.org/software/glpk. Then use the terminal to navigate to the directory where the distribution has been downloaded (e.g. cd ~/downloads) and execute the following commands:

```
tar -xzf glpk-4.48.tar.gz
cd glpk-4.48
./configure
sudo make
make install
```

GLPK should now be installed.

Step 5: Loading a domain ontology in Protégé

Start Protégé (in Windows, double-click on "protege.exe" in the installation directory of step 2). In the "Welcome to Protégé" pop-up window select "Open OWL ontology" and locate the .owl file with the domain ontology you want to load. NaturalOWL's distribution includes a directory "Ontologies", which contains sample domain ontologies and their linguistic and user modeling resources; we use the term "NL ontology" and "NL resources" to refer to the linguistic resources of a domain ontology.[8] For example, you can load the file "Ontologies/MPIRO/mpiro.owl".

[8] The M-PIRO and INDIGO domain ontologies were originally based on information kindly provided by the Foundation of the Hellenic World (FHW, http://www.fhw.gr/). See our JAIR article for information on the origin of the Consumer Electronics ontology and its sample instances.

Step 6: Using NaturalOWL v.2.0 in Protégé

Once you have loaded the .owl file of a domain ontology, go to Protégé's "Window" menu, select the "Tabs" sub-menu and tick "Lexicon", "NL Names", "Sentence Plans", "Sections and Order", "User Modelling", and "Text Generation" to add them to the visible tabs. You can now use the six tabs to author linguistic and user modeling resources, as explained in our JAIR article and the technical report mentioned in the introduction.

You can save the linguistic resources you have authored by selecting Protégé's "NaturalOWL" menu, and then "Save NL resources" or "Save NL resources as…". In the same menu, you can also create a new NL resources ontology, open an existing one, or import the resources from an additional NL resources ontology to the currently loaded one. Only one NL resources ontology can be loaded at a time.

Important: *Only the linguistic resources are saved in the NL resources ontology. The user modeling and ordering annotations are saved in the domain ontology; to save the domain ontology, select the "File" menu, and then "Save" or "Save as…".*

If a NL resources ontology is saved in a file named "NLResources.owl" and placed in the same directory as the .owl file of the domain ontology, it will also automatically loaded when the domain ontology is loaded.

Step 7: Generating texts with NaturalOWL v.2.0 in Protégé

You can generate texts from an OWL domain ontology with or without NL resources. The quality of the texts will be much better if appropriate NL resources are provided.

To generate a text:

- Go to the "Text Generation" tab.
- Select a class and/or an individual (instance of a class) of the domain ontology by using Protégé's Class Browser and Individual Browser on the left. For example, if you have loaded M-PIRO's domain ontology, select "exhibit", then "vessel", and then "kylix" in the Class Browser; then select "exhibit22" in the Instance Browser.
- Select the target language and user type in the "Text Generation" tab.[9] If you tick "Generate comparisons", the generated texts will occasionally contain comparisons to individuals described in previous texts and/or other individuals of the ontology.
- Specify the "Maximum Graph Distance in Content Selection"; see the articles mentioned in the introduction for an explanation of this parameter. For M-PIRO's ontology, we recommend setting this value to 2.
- Press "Generate text". A textual description of the selected class or individual should appear.

If you generate texts for several individuals (e.g., try generating previews for the three kylikes of M-PIRO's ontology, then generate a preview for the lekythos "exhibit15"), you will notice that NaturalOWL v.2.0 avoids repeating information that has already been conveyed; if you have ticked "Generate comparisons", it will also generate comparisons to previous individuals. Clicking on "Reset interaction history" will force NaturalOWL v.2.0 to "forget" the previous descriptions it has produced.

[9] In M-PIRO's domain ontology, the texts for experts do not convey information that the experts would already know (e.g., what a hydria was used for), nor information they would be able to infer from the image of the exhibit (e.g., that a particular amphora was painted with the red-figure technique).

You can select a generation architecture through the dropdown menu "Use engine", where you can select "Pipeline", "ILP model" or "ILP model (approx.)". Consult our ACL and ENLG 2013 papers for more information.

Important: *"ILP model" can be very slow when generating texts longer than 3 sentences! We automatically override the user model to limit all texts to a maximum of 3 sentences. For longer texts, use "ILP model (approx.)"instead.*

Other general options include: "Show Syntactic and Semantic Annotations", which will show the generated texts with syntactic and semantic markup in XML, "Shape text paragraphs" which will shape the generated text into paragraphs corresponding to the authored sections, and "Markup use of default resources", which will color the sentence plans and NL names in the text that are generated via built-in NL resources.

3. Modifying the source code of NaturalOWL v.2.0 and using it within Protégé

The source code of NaturalOWL v.2.0 is included in the distribution (see directory "source"). Provided that you respect its GPL license, you may modify the source code to create a tailored version of NaturalOWL. To use the new version within Protégé, follow the following steps:

Step 1

Extract all files of the distribution into the same directory.

Step 2

Modify the Java files of the source code.

Step 3

To create a new file "gr.aueb.cs.nlg.NLOwlPlugin.jar" we recommend you also download the source code of Protégé and add the plugin there. You can find relevant instructions here: http://protegewiki.stanford.edu/wiki/CompileProtege4InEclipseFromSvn. Following them, you can build the new "gr.aueb.cs.nlg.NLOwlPlugin.jar" through Eclipse.

Step 4

Copy "gr.aueb.cs.nlg.NLOwlPlugin.jar" to the "plugins" subdirectory of Protégé's installation directory.

4. How to call NaturalOWL v.2.0's generation engine from your software

You may call NaturalOWL v.2.0 directly from another application. File "TestNLGEngine.java" provides an example of how to do this.