SysNERV
User Documentation

Jindřich Helcl, Petr Jankovský, Ondřej Košarko,
Jan Mašek, Sara van de Moosdijk

supervisor: Zdeněk Žabokrtský

June 5, 2013

Contents

1 Introduction 3

2 Recognizers 4

2.1 Prerequisites 4

2.2 Installation 4

2.3 Instructions for use 5

3 Browser Extension 6

3.1 System requirements 6

3.2 Installation 6

3.3 Instructions for use 8

4 Web service 10

4.1 Prerequisites 10

4.2 Installation 10

4.3 Settings 11

5 Licensing 14

1 Introduction

This document describes the system requirements, installation processes, and instructions needed to use the applications that form Project SysNERV, a system for named entity recognition and visualization. All together this project includes named entity recognizers for English and Czech, a standalone application for applying the recognizers to text offline, and a web service with a corresponding Chrome browser extension for applying the recognizers to text online.

SysNERV aims to effectively recognize named entities in a text, where a named entity is defined as a word or phrase that refers to one particular and fixed thing. For example *Prague* is a named entity referring to the capital city of the Czech Republic. Our system provides recognizers for both English and Czech texts, which recognize and categorize the named entities. The categorization system differs for each language, but some examples of named entity categories are persons, locations, and institutions. Besides recognizing the named entities and their types, the SysNERV system aims to visualize the named entities by displaying additional information gained from Wikipedia[1] or Google Maps[2], relating to each entity.

Section 2 will focus on the installation of the recognizers and the standalone application. It will list the system requirements and explain how the standalone application can be applied. Section 3 explains how to find and download the Chrome browser extension in the Chrome Web Store. It provides details about setting up the extension according to your preferences, and instructions for using the extension to access the recognizers while browsing the web. The extension can be used without setting up your own web service, but section 4 clarifies how to set up and install the web service should you wish to host it yourself.

[1] `http://www.wikipedia.org/`

[2] `https://developers.google.com/maps/documentation/staticmaps/`

2 Recognizers

2.1 Prerequisites

For using SysNERV as a standalone application, you need a computer with a Linux operating system, and the following programs, utilities and libraries installed:

- libxml2-dev
- zlib1g-dev
- SWIG-2.0.4 or higher
- tar
- make
- bash
- perl 5.14.2 or higher
- g++
- gcc
- jre (*only for English*)

The standalone application also needs 460 MB of free space in the installation folder, and additional 540 MB during the installation in the user's home directory.

2.2 Installation

For installation of our software run script *install* placed in the installation folder. There are several available options:

- **-t** Module and library tests will be run.
- **-p** [*PATH*] The software will be installed in the specified path.
- **-v** Version of the SysNERV installer will be displayed.

By default, the software is installed to folder *sysnerv* to user's home directory. In case that folder with this name already exists, name of the new folder is extended by number suffix (eg. *sysnerv-1*). The default path can be changed by **-p** option. A temporary installation folder *.sysnerv-install* is created under the user's home directory. This folder is used just during the installation time and after successful installation is removed automatically.

In case of installation failure, all log files are stored in the temporary installation folder under the `log` directory.

To uninstall the software, just remove the program folder. All settings required by software are set during the run-time and are just temporal. All user settings are preserved.

2.3 Instructions for use

To run the standalone NER application, run `./bin/sysnerv-run` in the installation directory. Alternatively, if you add the "bin" folder to your `$PATH` variable (for example in your `/.bashrc` file), you can just run `sysnerv-run`.

The script analyses the input text for named entities and prints the HTML-formatted result to standard output. If no file is provided, the text is read from standard input.

Usage of the script:

```
sysnerv-run [OPTIONS] [FILE ...]
```

Available options are:

- `--lang, --language=cs|en` – specify the language used for NE recognition. Defaults to the Czech language.
- `--treex` – output the result in the Treex format. (Implies option `--to=noname001.treex.gz`)
- `--to[=FILE]` – output the result into `FILE`. (Can handle also "-" as standard input.)
- `--help` – display help and exit
- `--version` – display version information and exit

3 Browser Extension

3.1 System requirements

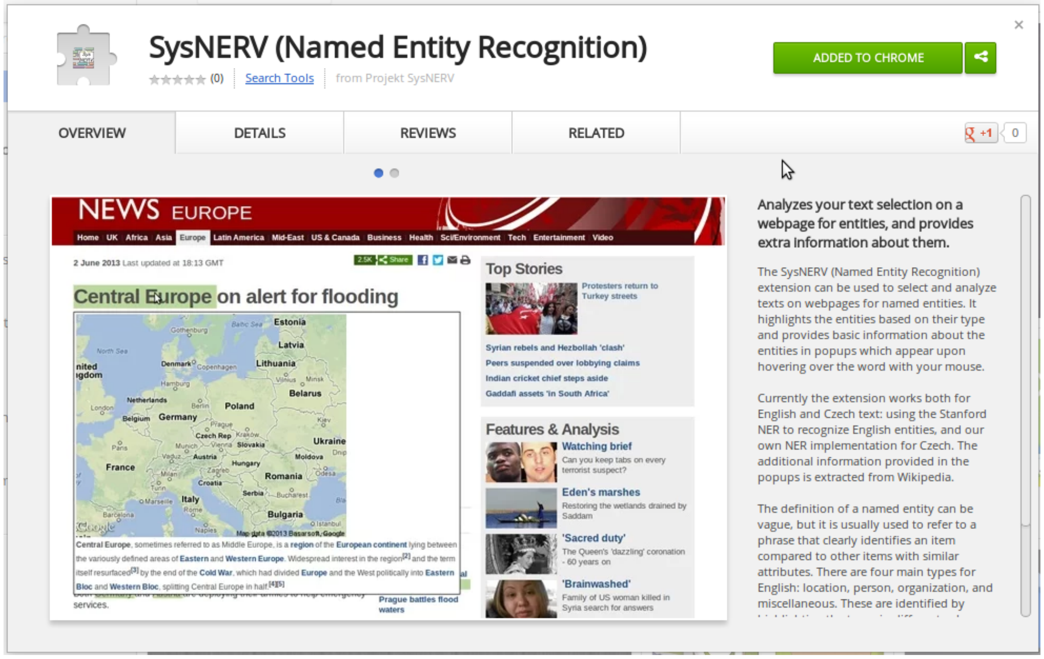
Installation is possible only in Google Chrome browsers version 18 and up, or the corresponding open-source Chromium project. It has been tested up to version 26, and requires a stable internet connection, so it is not suitable for use offline. There are no additional plugins required.

3.2 Installation

The extension can be acquired directly from the Chrome Web Store. It is listed under the name "SysNERV (Named Entity Recognition)" by the user Projekt SysNERV. And it can only be accessed by following the direct link[3]. Figure 1 shows the store page. Downloading an extension from the web store requires the user to log into a Google account. Once logged in, the extension can be installed by simple clicking the "Add to Chrome" button on the extension's store page. Chrome will complete the rest of the installation process for you, and the extension will ready to use after downloading the configuration. This process is explained in the next section.

Entity recognition can be applied to pieces of text on a webpage. First make sure that the extension has been correctly installed. You can view your all of your installed extensions on the Tools, Extensions page of the Google Chrome browser, as shown in Figure 2. On this page you will also see a link to the options page of the Sysnerv extension to download the web service configuration and set your language preferences. Currently there is only one central location from which you can download the configuration, unless you host the web service yourself. In that case the location of your privately hosted web service can be specified in the extension options page. So the first step is to set the location of the web service (don't make any changes if you wish to use the default location) and click "Download the configuration". You should see a message appear to confirm that the configuration has been downloaded successfully, and three Recognizer options relating to

[3] `https://chrome.google.com/webstore/detail/sysnerv-named-entity-reco/fpinegdcacehidkphifhbfnlajfnggmn`



SysNERV (Named Entity Recognition)

(0) | Search Tools | from Projekt SysNERV

2 June 2013 Last updated at 18:13 GMT

Central Europe on alert for flooding

Central Europe, sometimes referred to as Middle Europe, is a region of the European continent lying between the variously defined areas of Eastern and Western Europe. Widespread interest in the region[2] and the term itself resurfaced[3] by the end of the Cold War, which had divided Europe and the West politically into Eastern Bloc and Western Bloc, splitting Central Europe in half.[4][5]

services.

Prague battles flood waters

Top Stories

Protesters return to Turkey streets

Syrian rebels and Hezbollah 'clash'

Peers suspended over lobbying claims

Indian cricket chief steps aside

Gaddafi assets 'in South Africa'

Features & Analysis

Watching brief

Can you keep tabs on every terrorist suspect?

Eden's marshes

Restoring the wetlands drained by Saddam

'Sacred duty'

The Queen's 'dazzling' coronation - 60 years on

'Brainwashed'

Family of US woman killed in Syria search for answers

Analyzes your text selection on a webpage for entities, and provides extra information about them.

The SysNERV (Named Entity Recognition) extension can be used to select and analyze texts on webpages for named entities. It highlights the entities based on their type and provides basic information about the entities in popups which appear upon hovering over the word with your mouse.

Currently the extension works both for English and Czech text: using the Stanford NER to recognize English entities, and our own NER implementation for Czech. The additional information provided in the popups is extracted from Wikipedia.

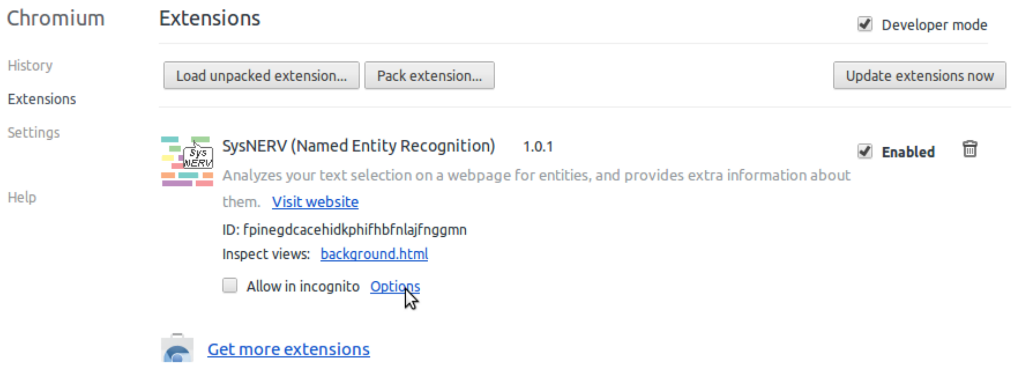
The definition of a named entity can be vague, but it is usually used to refer to a phrase that clearly identifies an item compared to other items with similar attributes. There are four main types for English: location, person, organization, and miscellaneous. These are identified by

Figure 1: Screenshot of the web store page

your language preferences will appear in a drop-down list. The three options are:

- AutoDetect: The extension will automatically detect the language of the selected text and send it to the corresponding recognizer. This is the default setting.
- English: Any selected text will automatically be sent to the English entity recognizer. Choose this option if you never plan to use the extension on Czech text.
- Czech: Any selected text will automatically be sent to the Czech entity recognizer. Choose this option if you never plan to use the extension on English text.

Choose the option most applicable to your desired use of the extension. Once again a confirmation message will appear. after which you can close the options page. The extension is now ready to use!



SysNERV (Named Entity Recognition) 1.0.1

Analyzes your text selection on a webpage for entities, and provides extra information about them. Visit website

ID: fpinegdcacehidkphifhbfnlajfnggmn

Inspect views: background.html

□ Allow in incognito Options

Figure 2: Screenshot of the extensions page in Chrome

3.3 Instructions for use

Once your preferences have been set, the extension is ready to be used. This can be done by navigating to the webpage of interest and selecting the text on that webpage for which you are interested in the entities it contains. After the text has been selected, right click on it and choose the "Analyze for entities" option from the context menu. The extension will send the text to the recognizers in separate paragraphs if your selection is relatively large, and a loading image will appear temporarily until at least one of the paragraphs has been completed. Once the loading image disappears you are free to view the information which has already been returned while waiting for the analysis of the remaining paragraphs to finish. If you select only small pieces of text the entire analysis will be done as a whole.

When the analysis of a piece of text is complete, any recognized entities will be highlighted with the designated color of their entity type. Table 1 shows which colors correspond to specific entity types. Popups containing extra information about the entities will appear when hovering the mouse over these highlighted words. This process of selecting and submitting text to the recognizer can be repeated as often as you wish for different text on a webpage.

There are still a few bugs which prevent the selection from always running smoothly. An error will appear to inform you when the extension is unable to recognize your selection. When this happens please reselect the text in

Tag	Meaning	Color
a	Numbers in addresses	🟨
c	Bibiographic items	🟩
g	Geographical names	🟩
i	Institutions	🟩
m	Media names	🟪
n	Specific number usage	🟥
o	Artifact names	🟥
p	Personal names	🟦
q	Quantitative expressions	🟦
t	Time expressions	🟧

Table 1: Colors corresponding to the entity types

a slightly different way. Refreshing the page might also help if the error is caused by a preceding error from the webpage itself.

To prevent as many of these selection errors as possible, there are a few guidelines you can follow when making your selection:

- Make selections containing only text, without images, tables, or other such elements.
- Make smaller selections (a few paragraphs instead of the whole page). This not only reduces the risk of selection errors, but also takes less time. You can analyse the page in parts and read the finished analyses while waiting for the next part to finish.
- Avoid selections that start or end near a header, caption, or other area where HTML DOM elements are likely to start or end.

4 Web service

4.1 Prerequisites

In order for the service to work you have to have the standalone package operational. That means you'll need the ”full Perl environment”. It might be possible to run it in the mod_perl environment, but that is currently not supported and probably would require some modifications in the source code.

As for the ”hardware” requirements, it depends on how many worker processes you choose to run. But note that the English NER runs under Java with -Xmx610m (uses quite a lot of memory for its model). The ufallab machine runs with two cores and 4G of memory.

4.2 Installation

Installing the service should be quite easy; just extract the supplied service.tar.gz. All the extra dependencies should be available through CPAN, but especially for Mojolicious we can not guarantee compatibility with future versions; that's why all the extra modules are available in packages/.

If you have cpanm and decide to install to the same directory as the standalone, you can proceed as follows in listing 1:

```
cd sysnerv-installed
tar -xvf service.tar.gz
cpanm -L ./lib/perl5 service/packages/Mojolicious-4.11.tar.gz

cpanm -L lib/perl5/ service/packages/Lingua-YALI-0.014.tar.gz

cpanm -L lib/perl5/ service/packages/HTML-TokeParser-Simple
   -3.15.tar.gz

cpanm -L lib/perl5/ service/packages/AnyEvent-7.04.tar.gz

cpanm -L lib/perl5/ service/packages/AnyEvent-HTTP-2.15.tar.gz
```

Listing 1: Installing extra modules

Wherever you decide to install the service, before you can actually run it, a proper environment must be set. Once again in the standalone install directory you could do something similar like in listing 2. You basically need to set the correct paths for Treex and other executables, tell Perl where it should look for the libraries and set where the configuration and the shared items (pretrained models) should go. Just note that the order of the paths in PERL5LIB does matter and if you use the suggested ROOT and CONFIG paths, you'll use the packed models. One final note; the web service won't work without the CzechMorpho data (those are supplied in the CONFIG/share).

```
CWD=$(pwd)

export PATH=”$CWD/lib/perl5/bin:$CWD/lib/libsvm/:$CWD/lib/treex/
    bin:$PATH”

export PERL5LIB=”$CWD/lib/treex/lib:$CWD/lib/treex/oldlib:$CWD/
    lib/perl5/lib/perl5:$PER5LIB”

export TMT_ROOT=$CWD/lib/treex

export TREEX_CONFIG=$CWD/lib/treex
```

Listing 2: Setting environment

You should be able to run ”treex -h” and ”mojo version” without errors, if everything went well.

Now proceed to the settings section to finish the configuration.

4.3 Settings

The service comes with its own server; the server configuration is at the beginning of dispatch.pl. Generally you can follow the documentation at `http://mojolicio.us/perldoc/Mojo/Server/Hypnotoad#SETTINGS`. Basically you should make sure the log file and pid file are writable, choose a port to listen on, and set the service_url.

The service_url is the baseurl of the service which will handle the extension's requests. You need to change that in order to use the service you've just installed. This config should be enough to run it locally:

```
##################### CONFIG START
    #####################################
```

```
app->config(hypnotoad => {listen => ['http://*:3000'], pid_file
    => '/tmp/hypnotoad.pid', workers => 2} );
my $log = Mojo::Log->new(path=> '/tmp/sysnerv.log', level=>'
    debug');
my $service_url = "http://localhost:3000/";
###################### CONFIG END
    ######################################
```

There is a small run script (run.sh) provided. You can use it as an inspiration for creating your own and/or creating an init script for the service.

```
source /etc/profile

case "$@" in
        -d | --debug)
                #MOJO_USERAGENT_DEBUG=1 MOJO_IOLOOP_DEBUG=1
                #morbo -l 'http://*:6000' dispatch.pl
                morbo dispatch.pl
        ;;
        -s | --stop)
                hypnotoad -s dispatch.pl
        ;;
        --start)
                nohup hypnotoad dispatch.pl  >/tmp/sysnerv_nohup
                    .out 2>&1 &
        ;;
        *)
                USAGE=1;
        ;;
esac
```

The sourcing of /etc/profile ensures that the proper paths are set. 'morbo' is a debug version of the server, it reloads as soon as you touch the file you've loaded with it. It ignores most of the configuration provided above (hypnotoad specific). So if you don't want to use the default (:3000) port you have to provide the -l option.

The 'hypnotoad' is a server implementation that's meant to be used for production. It tries to provide zero downtime so if you run start again (when the server is already running) it starts "hot deployment". For more options, see the official Mojolicious documentation.

We strongly recommend using morbo for the first run.

To quickly check your configuration you could try:

```
wget –O − −−tries=1 −−post−data='{"selection":"Miroslav Kalousek
    "}' http://localhost:3000/ner/ces
wget –O − −−tries=1 −−post−data='{"selection":"James Bond"}'
    http://localhost:3000/ner/eng
```

5 Licensing

This project uses the Perl module CzechMorpho in the training of the Czech named entity recognizer. The module and all of its files are copyrighted (C) Jan Hajič, 1989-2001. It is used in this project under the common license for PDT 1.0, for research and/or educational purposes[4]. Therefore, until further notice, the same license extends to all of the content and files directly connected to the Czech named entity recognizer of the SysNERV project. If the CzechMorpho ever becomes freely available for use in the future, this project will be open-source and free to use.

For English input this project uses the Stanford Named Entity Recognizer (NER), licensed under the GNU General Public License. Therefore all content and files directly related to the English named entity recognizer of the SysNERV project fall under this license.

The additional packages from CPAN included in the installation of SysNERV fall under the same license of Perl 5 programming language, and are therefore free to use.

[4] http://ufal.mff.cuni.cz/corp-lic/pdt10-ord.html