Multi-Lingual Noun Phrase Extractor (MuNPEx)

Guide for Users and Developers

René Witte

Release 2.0
July 26, 2015

Semantic Software Lab
Concordia University
Montréal, Canada
http://www.semanticsoftware.info

Contents

1 Multi-Lingual Noun Phrase Extractor (MuNPEx) 1
- 1.1 Overview . . . 1
- 1.2 Installation . . . 2
- 1.3 Usage . . . 3
  - 1.3.1 Dealing with Named Entities . . . 3
  - 1.3.2 Lemma Information . . . 3
  - 1.3.3 Runtime parameters . . . 3
- 1.4 Implementation notes . . . 4
  - 1.4.1 Named Entities . . . 4
  - 1.4.2 Slot position information . . . 4
  - 1.4.3 Pipeline Unit Tests . . . 4
- 1.5 Changelog . . . 5

About this document

This document contains documentation for the *Multi-Lingual Noun Phrase Extractor (MuNPeX)*. You can obtain the latest version from http://www.semanticsoftware.info/munpex.

Acknowledgments

Thanks to Michelle Khalifé for helping out in developing the French version.

License

Support

For question or general comments, please use the online forum at http://www.semanticsoftware.info/forums/tools-resources-forum/durm-corpus-wiki-tools

[1] LGPL3, https://www.gnu.org/licenses/lgpl-3.0.en.html

Chapter 1

Multi-Lingual Noun Phrase Extractor (MuNPEx)

MuNPEx is a multi-lingual noun phrase extraction component implemented in JAPE. Currently supported languages are English, German, and French, with additional Spanish support in beta.

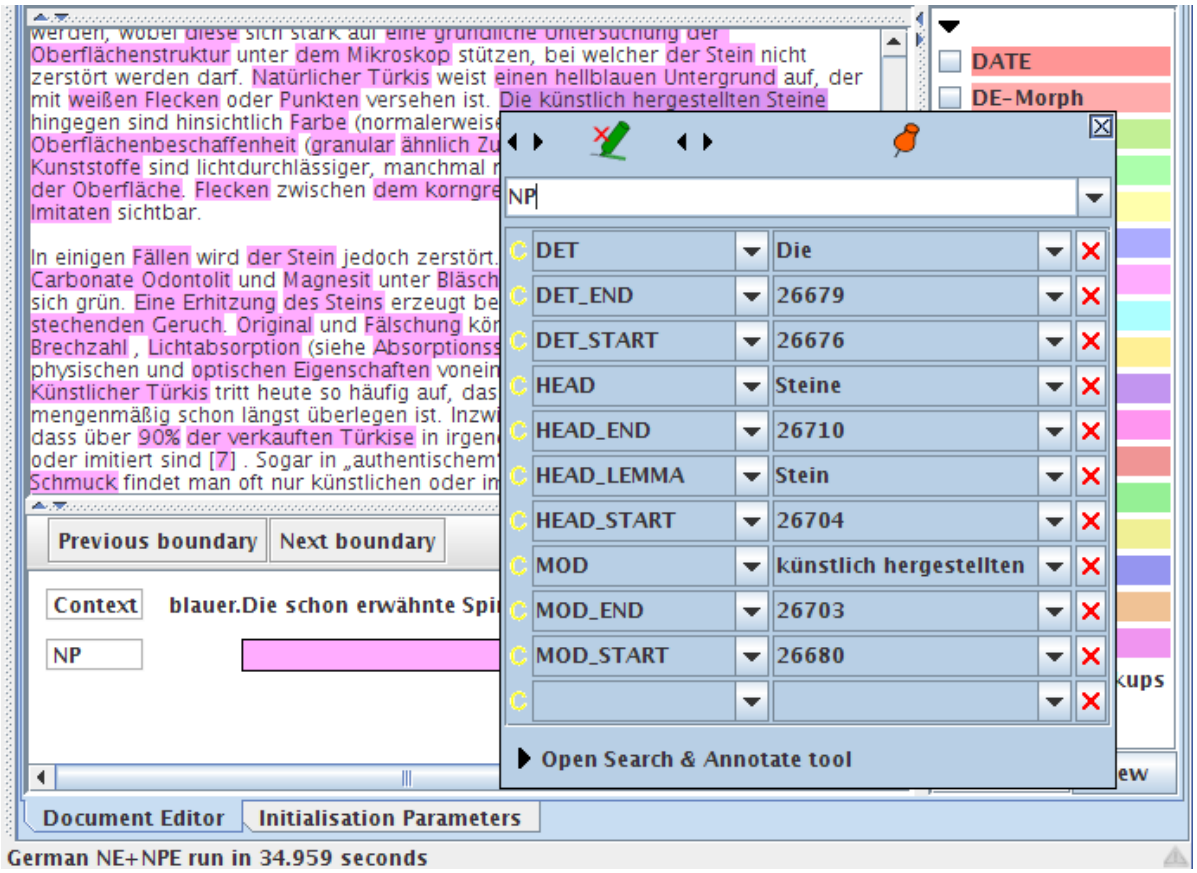

Figure 1.1: Example annotations generated by MuNPEx on a German document

1.1 Overview

MuNPEx is a base NP chunker, i.e., it does not deal with any kind of conjunctions, appositions, or PP-attachments. It supports several languages while attempting to re-use as much

code as possible between the different languages. Another feature is that it can makes use of previously detected named entities (NEs) to improve chunking performance.

For each detected NP, an annotation "NP" is added to the document, which includes several features (Figure 1.1):

DET the determiner of the NP

MOD a list of modifiers of the NP

HEAD the head noun of the NP

Pronoun boolean value {true, false} indicating a pronoun NP

MOD2 *(only for French and Spanish)* NP modifiers that appear *after* the HEAD noun

Using an additional grammar (see below), it can also add lemmatization information for the HEAD noun.

HEAD_LEMMA *(optional, with additional grammar)* lemmatized form of the HEAD noun

Optionally, it can generate additional features indicating the textual positions of the slots described above:

HEAD_START *(optional)* the position in the document where the NP's HEAD starts

HEAD_END *(optional)* the position in the document where the NP's HEAD ends

DET_START *(optional)* the position in the document where the NP's determiner starts

DET_END *(optional)* the position in the document where the NP's determiner ends

MOD_START *(optional)* the position in the document where the NP's first MOD starts

MOD_END *(optional)* the position in the document where the NP's last MOD ends

1.2 Installation

Prerequisites. To run MuNPEx, you must have GATE installed. Please refer to http://gate.ac.uk for instructions on how to download and install GATE. The automated install method described below requires GATE version 8 or better, but the NP grammars can be run on GATE version 5 or better.

Demo Pipeline. The MuNPEx distribution comes with a demo pipeline for English that you load directly from within GATE:

1. Start GATE, open the Plugin Manager *(File → Manage CREOLE Plugins...)*
2. On the fourth tab *(Configuration)*, select a writable directory for the local plugin installations and enable the "Semantic Software Lab" repository
3. On the third tab *(Available to install)*, you will now find MuNPEx. Select it and click on "Apply All".
4. On the first tab *(Installed Plugins)*, you will now find the MuNPEx plugin. Select it (either "Load Now" or "Load Always") and click on "Apply All".

To load the example pipeline, go to *File → Ready Made Applications → Semantic Software Lab → MuNPEx English NP Chunker Demo Application.*[1]

[1] Please refer to the GATE user's guide, http://gate.ac.uk/sale/tao for further details on working with pipelines in GATE.

1.3 Usage

To load the chunker component, simply create a new JAPE transducer (or JAPE-Plus transducer) component and load the main grammar file `xx-np_main.jape`, where `xx` is the language code (currently supported are *en* – English, *de* – German, *fr* – French, and *es* – Spanish). Alternatively, if you installed MuNPEx via the Plugin Manager, you can use the corresponding pre-configured PRs (which are simply JAPE transducers with the corresponding grammars).

Note that MuNPEx needs part-of-speech tags to function properly, so you have to run it after a POS tagger component. For English, you can use the Hepple tagger included in the GATE distribution (part of ANNIE); for German, French, and Spanish you can use the TreeTagger[2] which can be executed in GATE using the TaggerFramework PR (see the sample pipelines for the TreeTagger that come as part of the GATE distribution).

Alternative POS taggers are possible (e.g., using the Stanford POS tagger for German), but have not been tested extensively yet and require further updates to the language-specific tags.

1.3.1 Dealing with Named Entities

If an NLP pipeline contains other components for detecting Named Entities (NEs), MuNPEx can make use of those NEs for the chunking process. For example, the default ANNIE application shipped with GATE detects entities like *Person*, *Date*, or *Organization*. Instead of replicating code for detecting such entities within MuNPEx, it can simply use them for the appropriate HEAD or MOD slots within an NP (where exactly an NE can occur depends on its type and the language, see below for implementation details).

Of course, MuNPEx can also be used like any classical NP chunker oblivious to other NEs by placing it *before* any NE detection components (or removing the entities from the MuNPEx files). A mixed approach is also possible, where you first find some NEs for use within MuNPEx, and later detect more complex NEs using noun phrase chunks. The proper strategy here is highly application specific.

1.3.2 Lemma Information

If you need lemma information for the HEAD noun, you can include one of the following three grammars to extract the lemma information and add it as the HEAD_LEMMA feature (see Figure 1.1). This can make it easier for following components to work with the lemma information, as they do not have to know where the lemma is coming from.

en-lemma.jape for English: requires the GATE Morphological Analyser PR

de-lemma.jape for German: requires the Durm German Lemmatizer[3]

treetagger-lemma.jape works for all TreeTagger-processed documents

Simply load one of these grammars into a JAPE Transducer PR and add it *after* the MuNPEx transducer/PR.

1.3.3 Runtime parameters

As for every JAPE transducer component, you can set the

inputASName input annotation set; and

[2] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/DecisionTreeTagger.html
[3] http://www.semanticsoftware.info/durm-german-lemmatizer

outputASName output annotation set.

Make sure that the `inputASName` is the annotation set containing the `Token` annotations (and any optional named entities you want to use).

1.4 Implementation notes

MuNPEx contains language-specific and language-independent files. Every language-specific file has a two-letter prefix indicating the language it is used for.

MuNPEx is implemented as a set of grammars running in a multi-phase JAPE transducer. For each language, the file `xx-np_main.jape` contains the transducer definition, which generally contains five phases:

xx-np-parts.jape contains the language-specific definitions for the determiner, modifier, and head slots

np-entities.jape is a language-independent file defining which named entities to use for the HEAD slot of an NP (note that for English, NEs can also appear in the MOD slot)

check.jape is a language-independent file that handles the special case where the transducer's inputAS ! = outputAS

np.jape is a language-independent file that constructs NPs from the constituents detected in the previous phases

clean.jape cleans up temporary annotations

1.4.1 Named Entities

If you want to add new named entities to MuNPEx, you'll have to add the name of the NE to the file `np-entities.jape`. For example, if your application detects *IceCreamFlavours* as a named entity, you need to add this entity both to the `Input:` declaration and the `Rule: head`. If your NE can additionally appear within a MOD slot (this only happens in English), you also need to add it to the file `en-np-parts.jape`.

The distribution contains a (simple) JAPE example, the *Number Transducer*,[4] which you can add before MuNPEx to include number entities in the chunks (this works for all languages).

1.4.2 Slot position information

By default, MuNPEx creates several additional position features within each NP annotation, like `HEAD_START` and `HEAD_END`. This can help to speed up access to other enclosing/embedded annotations for these slots in subsequent components. If you don't like/need those, you can simply comment out the corresponding `features.put` lines within the file `np.jape`.

1.4.3 Pipeline Unit Tests

The `test/` directory contains a few JUnit sanity tests to check whether the demo pipeline is working. They can be run with the provided `buildtest.xml` ant script by calling the `test` target.

[4] in `optional/combineNumbers.jape`

1.5 Changelog

Version 2.0 (26.07.2015)

- New 'Pronoun' feature added to NP annotations
- License updated from GPL2 to LGPL3
- Basic JUnit tests for demo pipeline added

Version 1.2 (24.03.2012)

- Minor bugfix to English chunker

Version 1.1 (12.02.2012)

- Repackaged for new GATE 7 Plugin Manager
- Added example pipeline for English
- Added wrappers to load MuNPEx as PRs

Version 1.0 (16.08.2010)

- More robust on malformed input.
- Optional grammars for adding a HEAD_LEMMA slot.
- DET/MOD/HEAD/MOD2 now stored as strings instead of Content objects.
- Code cleanup and tweaks.

Version 0.2 (03.03.2006)

- Preliminary Spanish support added.
- Renamed from "NPE" to "MuNPEx".
- Small cleanups.
- Number transducer added.

Version 0.1 (21.11.2005)

Initial public release.