

1 Multilingual Noun Phrase Extractor (MuNPEX)

MuNPEX is a multi-lingual noun phrase extraction component implemented in JAPE. Currently supported languages are English, German, French, and Spanish (in beta).

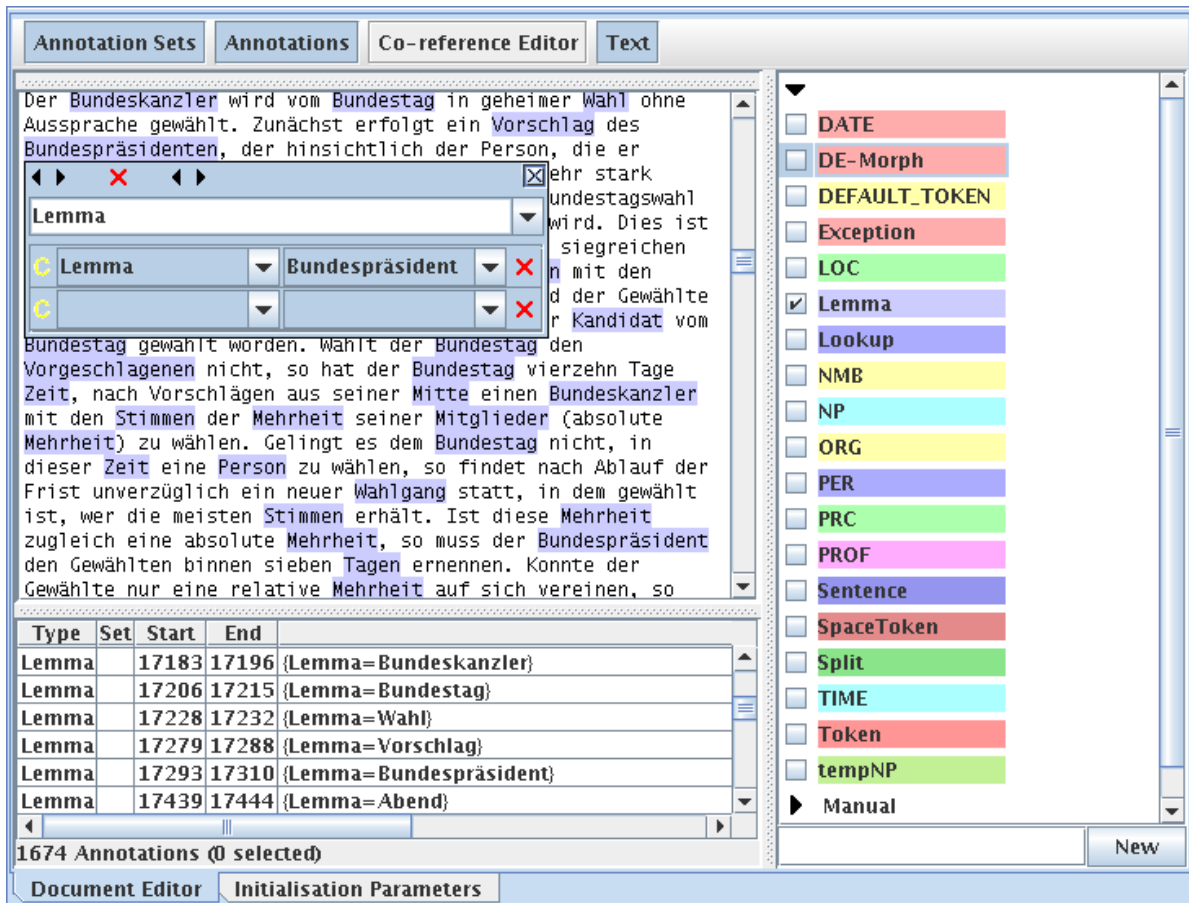


Figure 1.1: Example annotations generated by MuNPEX on a German document

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 make 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):

1 Multilingual Noun Phrase Extractor (MuNPEX)

DET the determiner of the NP

MOD a list of modifiers of the NP

HEAD the head noun of the NP

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

and similarly for the other slots.

1.2 Usage

To load the chunker, simply create a new JAPE transducer component and load the main grammar file `xx-np_main.jape`, where `xx` is the language (currently supported are *en* – English, *de* – German, *fr* – French, and *es* – Spanish).

Note that MuNPEX needs part-of-speech tags to function, so you have to run it after a POS tagger component. For English, you can use the Hepple tagger included in the GATE distribution; for German, French, and Spanish you can use the TreeTagger¹ with its GATE wrapper included in the distribution since version 3.

1.2.1 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.

1.2.2 Runtime parameters

As for every JAPE transducer component, you can set the

inputASName input annotation set; and

outputASName output annotation set.

Note: Currently, you'll need to set both to the same annotation set, or MuNPEX will not work.

¹<http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/DecisionTreeTagger.html>

1.3 Implementation notes

MuNPEX contains language-specific and language-independent files. All language-specific files have a two-letter prefix indicating the language its used for.

MuNPEX is implemented as a set of multi-phase JAPE transducer. For each language, the file `xx-np_main.jape` contains the transducer definition, which generally loads three sub-transducers:

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)

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

clean.jape cleans up temporary annotations

1.3.1 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`.

You can download a (simple) JAPE *Number Transducer* from the website to see a real-world example on how entities are used in the chunker (this works for all languages).

1.3.2 Slot position information

By default, MuNPEX creates two additional features within each NP annotation, `HEAD_START` and `HEAD_END`. This helps to speed up access to the "HEAD" slot in subsequent components. If you don't like/need those, you can simply comment out the two `features.put` lines within the file `np.jape`. Alternatively, if you'd also like to have the position information for the "MOD" and "DET" slots, you can un-comment the corresponding lines in `np.jape`.