# HeidelTime Standalone Version
# Manual

Julian Zell, Andreas Fay, Jannik Strötgen (Heidelberg University)
zell@informatik.uni-heidelberg.de, stroetgen@uni-hd.de

September 2013

**Abstract**

This document contains information about how to install and use the standalone version of HeidelTime. HeidelTime itself is a multilingual temporal tagger for the extraction and normalization of temporal expressions from documents, developed at the Heidelberg University from Strötgen and Gertz [6, 7, 8].
The original version of HeidelTime is designed to run within a proper UIMA-Pipeline [1]. With this standalone version the original version is wrapped such that it can be run with less prerequisites and especially without UIMA.
HeidelTime Standalone comes with resources for English, German, Spanish, Italian, Vietnamese, Arabic, and Dutch. Dutch resources were developed and kindly provided by Matje van de Camp (Tilburg University)[4]. French resources were provided by Véronique Moriceau (LIMSI-CNRS)[10].

## Contents

# 1 Preface

This document contains information about how to install and use the standalone version of HeidelTime. HeidelTime itself is a multilingual temporal tagger for the extraction and normalization of temporal expressions from documents, developed at the University of Heidelberg from Strötgen and Gertz [6, 7]
The original version of HeidelTime is designed to run within a proper UIMA-Pipeline [1]. With this standalone version the original version is wrapped such that it can be run with less prerequisites and especially without UIMA.

# 2 Quick Start

This section will briefly outline what is necessary in order to get HeidelTime Standalone going. See Section 3 for a more detailed description.

1. Install Java Runtime Environment [9] in order execute java programs.
2. Install TreeTagger [5] of the University of Stuttgart with the parameter files for English, German, Dutch, Spanish, Italian and French.
3. Ensure the path to your local TreeTagger installation is set correctly. Therefore, check the variable *treeTaggerHome* in `config.props`. It has to point to the root directory of your TreeTagger installation.
4. Change to the directory containing `de.unihd.dbs.heideltime.standalone.jar`.
5. Run HeidelTime Standalone using
   "java -jar de.unihd.dbs.heideltime.standalone.jar <file>"
   where *<file>* is the path to a text document.

# 3 Installation

This section explains the steps necessary to use HeidelTime Standalone.

## 3.1 Files

HeidelTime Standalone comes with three files and two folders:

- `de.unihd.dbs.heideltime.standalone.jar`
  Executable java file; see Section 4 for more information about possible command line arguments.

- `config.props`
  Configuration file; it has to be located in the same directory as the executable. See Section 3.3 for more information about the configuration of HeidelTime Standalone.

- **src/**
  Folder containing the source files that were used to generate the executable jar file `de.unihd.dbs.heideltime.standalone.jar`.

- **doc/**
  Folder containing the Javadoc files.

- `Manual.pdf`
  This file.

### 3.2 Prerequisites

HeidelTime Standalone requires the following two components to be installed:

1. The Java Runtime Environment [9] and
2. A compatible pre-processing tagger that is capable of identifying language tokens, part of speech and sentence boundaries in all languages supported by HeidelTime. We decided to use TreeTagger [5] of the University of Stuttgart for English, German, Dutch, Spanish and Italian. You will need to download and install the so called "parameter files" for those languages as well (all that are available, e.g., for German, download the Latin1 and the UTF-8 parameter files), to provide TreeTagger with the necessary functionality (see the TreeTagger Web-site for more information).
3. If you use HeidelTime Standalone to annotate documents in Vietnamese, you will need to get a copy of JVnTextPro [2]
4. For Arabic documents, you will need to download a full package of the Stanford POS Tagger [3]

   Note 2: If you use HeidelTime Standalone on Windows, please see Appendix A.

### 3.3 Configuration

After the installation of the prerequisites mentioned in Section 3.2, there are a few parameters to set up in the configuration file `config.props`:

### For most languages

- *treeTaggerHome*
  This variable has to point to the root directory of TreeTagger that you will need to use for most languages.

### For use with Vietnamese

- *word_model_path*
  This variable needs to point to the *folder* where JVnTextPro's segmentation model is stored.
  Example: `/opt/jvntextpro/models/jvnsegmenter`

- *sent_model_path*
  This variable needs to point to the *folder* where JVnTextPro's sentence segmentation model is stored.
  Example: `/opt/jvntextpro/models/jvnsensegmenter`

- *pos_model_path*
  This variable needs to point to the *folder* where JVnTextPro's part of speech model is stored.
  Example: `/opt/jvntextpro/models/jvnpostag/maxent`

### For use with Arabic

- *model_path*
  This variable needs to point to the path where StanfordPOSTagger's tagger model *file* is stored.
  Example: `/opt/stanfordpostagger/models/arabic-accurate.tagger`

- *config_path*
  This variable can be set to point to the path where StanfordPOSTagger's config model *file* is stored. This setting is optional and can be omitted (left empty).
  Example: `/opt/stanfordpostagger/tagger.config`

### General options

- *considerDate*
  Indicates whether HeidelTime should consider Timex3 expressions of type DATE.

- *considerDuration*
  Indicates whether HeidelTime should consider Timex3 expressions of type DURATION.

- *considerSet*
  Indicates whether HeidelTime should consider Timex3 expressions of type SET.
- *considerTime*
  Indicates whether HeidelTime should consider Timex3 expressions of type TIME.

All other options are not meant to be changed and therefore skipped in this section.

# 4 Usage

This section explains how to use HeidelTime Standalone both as a command line tool and as a component in other Java projects.

## 4.1 Command Line Usage

To use HeidelTime Standalone, open a command line terminal and switch to the directory containing `de.unihd.dbs.heideltime.standalone.jar`. You then are able to run it using the following command:
"java -jar de.unihd.dbs.heideltime.standalone.jar <file> [options]" where *<file>* is the path to a text document on your hard disk and *[options]* are possible options explained in Table 1.

### Extra steps for Arabic and Vietnamese tagging

To tag Arabic and Vietnamese documents, you will need to utilize a different command line scheme. First, you will have to set the `HT_CP` variable to include HeidelTime Standalone's class files as well as those of the languages' respective taggers:

Under Unix/Linux/Mac OS X:

```
"export HT_CP="<$1>:<$2>:<$3>:$CLASSPATH""
```

or under Windows:

```
"set HT_CP=<$1>;<$2>;<$3>;%CLASSPATH%"
```

where
<$1> is the path to JVnTextPro's `bin` folder, e.g. `/opt/jvntextpro/bin/`,
<$2> is the path to StanfordPOSTagger's `.jar` file, e.g.
`/opt/stanfordpostagger/stanford-postagger.jar` and
<$3> is `de.unihd.dbs.heideltime.standalone.jar`

Once you have this variable set, you can use the following command line:

```
java -cp $HT_CP de.unihd.dbs.heideltime.standalone.HeidelTimeStandalone
<file> [options]
```

where *<file>* is the path to a text document on your hard disk and *[options]* are possible options explained in Table 1.

Table 1: Command line arguments of HeidelTime Standalone.

| OPTION | NAME | DESCRIPTION |
|---|---|---|
| -dct | Document Creation Time | Date of the format YYYY-MM-DD when the document specified by *<file>* was created. This information is used only if "-t" is set to NEWS or COLLOQUIAL. It is used to resolve relative temporal expression such as "today". The default value is the current date on the local machine. |
| -l | Language | Language of the document. Possible values are: ENGLISH, GERMAN, DUTCH, ENGLISHCOLL (for -t COLLOQUIAL), ENGLISHSCI (for -t SCIENTIFIC), SPANISH, ITALIAN, ARABIC, VIETNAMESE, FRENCH. The default is ENGLISH. |
| -t | Type | Type of the document specified by *<file>*. Possible values are: NARRATIVES, NEWS, COLLOQUIAL and SCIENTIFIC. The default value is NARRATIVES. The major difference between these types is the consideration of "-dct" if type is set to NEWS or COLLOQUIAL. |
| -o | Output Type | Type of the result. Possible values are: XMI and TIMEML. The default value is TIMEML. |
| -e | Encoding | Encoding of the document that is to be processed, e.g., UTF-8, ISO-8859-1, ... Default value is UTF-8. |
| -c | Configuration file | Relative or absolute path to the configuration file. Default file is config.props |
| -v/-vv | Verbosity | Turns on verbose or very verbose logging. |
| -it | IntervalTagger | Enables the IntervalTagger and outputs recognized intervals. |

| OPTION | NAME | DESCRIPTION |
|---|---|---|
| -locale | Locale | Lets you set a custom locale to run HeidelTime under. Format is: X_Y, where X is from ISO 639 and Y is from ISO 3166, e.g.: "en_GB" |
| -pos | POS Tagger | Lets you choose a specific part of speech tagger; either STANFORDPOSTAGGER or TREETAGGER. Note that for Arabic or Vietnamese documents, we allow only StanfordPOSTagger and JVnTextPro respectively. Please take note of the prerequisites in Section 4.1. |
| -h | Help | Shows you a list of commands and usage information |

You may omit any of the options since they are optional. HeidelTime Standalone will however force you to enter a valid document path. It will output an XMI- or TimeML-document to the standard output stream containing all annotations made by HeidelTime. You may save the output to a file by using the following command:
`"java -jar de.unihd.dbs.heideltime.standalone.jar <file> [options] > <outputfile>"` where *<outputfile>* is the path to the document where the output will be saved into.

**Encoding settings:** HeidelTime Standalone can process files of different encodings. However, independent of the input encoding, the output is always encoded as UTF-8. If the default encoding of your Java Virtual Machine is not UTF-8, **you have to set the encoding to UTF-8** using the -Dfile.encoding option:

```
"java -Dfile.encoding=UTF-8 -jar de.unihd.dbs.heideltime.standalone.jar
<file> [options]"
```

If the encoding of the document that is to be processed is not UTF-8, you can specify the encoding with parameter “-e” as described in Table 1.

## 4.2 Component in other Projects

To use HeidelTime Standalone as a component in other projects, you have to prepare the executable jar file `de.unihd.dbs.heideltime.standalone.jar`: Add the configuration file `config.props` to the main directory of the executable using a proper archive tool. Once this is done you can copy the executable wherever you want and use it like a library. To run HeidelTime Standalone, instantiate an object of *HeidelTimeStandalone*. To do so, you simply have to provide the desired language and type that is to be processed (see Table 1 for further information). To actually run HeidelTime, you have to call *process* on the recently instantiated object of type *HeidelTimeStandalone* with the text to be processed. If this text is of type NEWS (remember your decision when instantiating a *HeidelTimeStandalone* object), you have to provide the document creation time as well. As a result you will get a string containing the TimeML document with all annotations made by HeidelTime for further treatment.

## 5 License

If you use HeidelTime, please cite one of the papers describing HeidelTime: [6, 8]. Thank you.

For details, see `http://dbs.ifi.uni-heidelberg.de/heideltime/` or `https://code.google.com/p/heideltime/`.

## References

[1] Apache Software Foundation. Apache UIMA, June 2011. URL `http://uima.apache.org/`.

[2] Thu-Trang Nguyen Cam-Tu Nguyen, Xuan-Hieu Phan. JVnTextPro, April 2013. URL `http://sourceforge.net/projects/jvntextpro/`.

[3] Stanford Natural Language Processing Group. Stanford POS Tagger, April 2013. URL `http://nlp.stanford.edu/software/tagger.shtml`.

[4] Matje van de Camp. Dutch resources, 2011. URL `http://www.tilburguniversity.edu/webwijs/show/?uid=m.m.v.d.camp`.

[5] Helmut Schmid. TreeTagger, July 2013. URL `http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/`.

[6] Jannik Strötgen and Michael Gertz. HeidelTime : High Quality Rule-based Extraction and Normalization of Temporal Expressions. In *Proceedings of the 5th International Workshop on Semantic Evaluation (SemEval)*, pages 321–324, Uppsala, Sweden, 2010.

[7] Jannik Strötgen and Michael Gertz. HeidelTime, May 2012. URL `http://dbs.ifi.uni-heidelberg.de/heideltime/`.

[8] Jannik Strötgen and Michael Gertz. Multilingual and cross-domain temporal tagging. *Language Resources and Evaluation*, 2012. doi: 10.1007/s10579-012-9179-y.

[9] Sun Microsystems. Java, March 2011. URL `http://www.java.com`.

[10] Véronique Moriceau. French resources, 2013. URL `http://vero.moriceau.free.fr/`.

# A Information for Windows Users

If you are using HeidelTime standalone on Windows, you have to download and install at least the following TreeTagger [5] scripts and resources, in addition to the Windows version of the TreeTagger:

- utf8-tokenize.perl
- german-abbreviations-utf8
- dutch-abbreviations

For this, download and extract the TreeTagger tagging scripts `http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/data/tagger-scripts.tar.gz` (be aware that this file should be extracted in an empty folder). Then, copy the following files to your TreeTagger folders:

- copy cmd/utf8-tokenize.perl to TREETAGGER_HOME/cmd/
- copy lib/german-abbreviations-utf8 to TREETAGGER_HOME/lib/
- copy lib/dutch-abbreviations to TREETAGGER_HOME/lib/

You should now be able to run HeidelTime standalone on a Windows machine.