

Supervised Learning Semantic Processor

Marcel Köster
Jonas Sunde
Universität des Saarlandes
winter semester 2008/2009

revised March 25, 2009

Chapter 1

What is a ... good for ?

1.1 ... Trainer Plugin ...

A trainer plugin creates a model that can be used to predict the meaning of a sentence.

1.2 ... Cluster Plugin ...

A cluster plugin does collect data about sentences which is later used by a trainer to create a model.

1.3 ... Corpus Word Generator Plugin ...

A corpus word generator plugin is used to create all possible inflected forms of a given word
e.g. burn → burn, burns, burned, burning

1.4 ... Corpus Generator Plugin ...

A corpus generator is used to create a number of sentences for a given word list.
e.g.(burn, burns, burned, burning) → The house burned to the ground., My skin burns like fire., ...

1.5 ... Parser Plugin ...

A parser plugin parses given sentences.

1.6 ... POS Tagger Plugin ...

A POS Tagger Plugin is used to enrich given sentences with part of speech tags.

1.7 ... war ...

absolutely nothing !
say it again yoah

Chapter 2

How to

2.1 ... write a Cluster Plugin

To write a new Clusterer all you have to do is open the `plugin.clusterPlugins` package and create a new class that extends `ClusterPlugin`.

The important thing here is that you have to place your class in this exact package or else the GUI won't be able to detect it.

2.2 ... write a Trainer Plugin

To write a Trainer all you have to do is open the `plugin.trainerPlugins` package and create a new class that extends `TrainerPlugin`.

The important thing here is that you have to place your class in this exact package or else the GUI won't be able to detect it.

2.3 ... write a Corpus Generator Plugin

To write a new corpus generator all you have to do is open the `plugin.corpusGeneratorPlugins` package and create a new class that extends `CorpusGeneratorPlugin`.

The important thing here is that you have to place your class in this exact package or else the GUI won't be able to detect it.

2.4 ... write a Parser Plugin

To write a new parser Plugin all you have to do is open the `plugin.parserPlugins` package and create a new class that extends `TrainerPlugin`.

The important thing here is that you have to place your class in this exact package or else the GUI won't be able to detect it.

2.5 ... extend the GUI ...

First of all if you just want to implement a new Plugin of a type that is already in use you don't have to change anything in the GUI ! If you do it right it will automatically detect the new Plugins.

If you want to extend the GUI for new features

Chapter 3

Ideas, TODOs and future work

3.1 automated translation of display language

The idea is to make it possible for people who don't speak a language for which a handwritten property file is available, to let the program automatically translate itself using web services (like babelfish).

In Order to do that you would have to:

- extend for example the OptionsDialog in the gui package with a new button or the like
- write a class that can extract all keys and values from a property file (or make use of the functionality of ResourceBundle) that is available and writes a new property file with the translated text using a web service API and save it to the gui.languageFiles package
- link the button with the class

3.2 automated search for cluster settings

The idea is to make it possible for the program to find the optimal (or at least good) settings for one (or more) cluster plugin(s).

In order to achieve this our idea would be:

- extend the clusterSettings class in the plugins.clusterPlugins package with a iteration functionality
- implement this iterator in all clusterPlugins (or only those you want one for)
- build some new GUI Panels
- the user should be able to load two annotated Data Files. one that is used to train and one that is used to test the model.
- now the program should iterate over all cluster settings, create a new model for each setting with the trainingsdata and test this model against the testdata given some metric
- with the aid of that metric you now can determine the clusterSetting that gives the best results (for that specific word and annotated data of course)