System PPAttach tackles the problem of prepositional phrase attachment by incorporating semantic knowledge derived from the lexico-semantic ontologies such as VER ! E" and # $R% ! E"& "he system assumes input in the form of set of tuples

T1' (verb) noun) preposition) noun*

+or a given set of tuples PPAttach will return its decision on each tuple on whether it triggers verb or noun attachment& PPAttach uses machine learning methods to implement its decision procedure& , achine learning methods are commonly used for implementing classification procedures called classifiers& -n supervised learning) the classifier is first trained on a set of labeled data (training data* that is representative of the domain of interest& "ypically labeled data consists of pairs of input ob.ects and a desired output& An input ob.ect is often summari/ed by so called feature vector& "he trained classifier is then used to carry out classification decisions for unseen data (testing data*& PPAttach uses classic 0Ratanaparkhi1 dataset) composed of labeled2annotated tuples of the form ("3*) for 0training1 and 0testing1& #eka 4 a machine learning tool of the 5niversity of #aikato http'22www&cs&waikato&ac&n/2ml2weka2 4 is used within the framework to carry out the classification&

Site

http'22www&unomaha&edu2nlpkr2software2ppattach2

is the pro.ect6s website which contains a link to the paper on

738 0Prepositional Phrase Attachment Problem Revisited: How VERBNET Can Help" b !aniel Baile "#$li a %ierler" Ben&amin '$sman" -n Proceedin(s o) the 11th *nternational Con)erence on Comp$tational 'emantics +*, C'-) 9:3;&

"his paper is the best resource for details on the implemented techni<ues&

"his document provides directions on setting up) running) and extending the PPattach system& "he PPattach system is composed of two main components& $ne component is responsible for building feature vectors for given tuples of the form (T1*) another component is responsible for processing these feature vectors and performing the classification itself& "he former component is written in python by the authors of the pro.ect& "he latter component relies on #eka&

"he pro.ect uses Python 9&= with !>"? 9&:&@& -f the destination machine does not currently have !>"? or is running !>"? A) !>"? 9 will need to be installed& -nstructions for doing so can be found at (under the NLTK heading) Buestion2Answers 3C-3=*'

http'22www&pitt&edu2Dnaraehan2python92fa<&html

-nstructions are provided for >inux users (but modulo command line commands these instructions can easily be adapted on #indows*&

+or P?-->inux lab users'

```
E cp -R 2nlpkr2ppattach2 FFtheFdirectoryFofFyourFchoiceFF
```

+or general public'

%ownload and un/ip the following file'

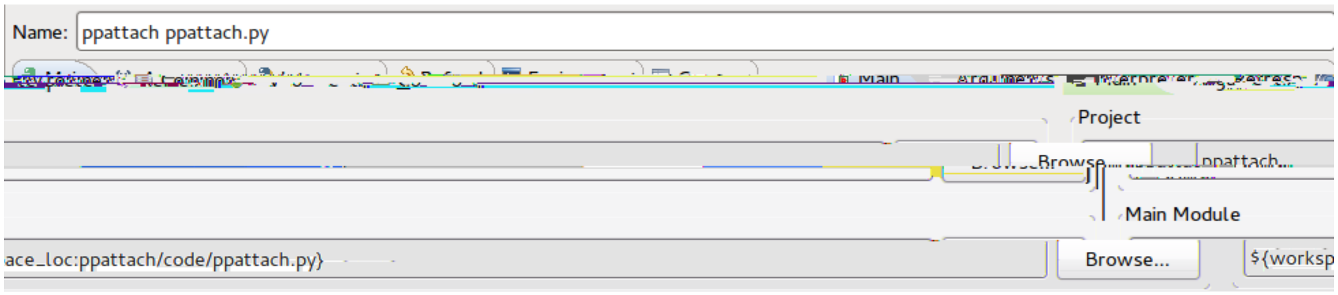
```
E cd 2home2ylierler2ppattach
E python code2ppattach&py -h
```

A& Jo to' Run K Run As K Python Run K !ew

Change !ame to ppattach ppattach1p

Change Pro.ect to ppattach

Change ,ain ,odule to 34wor/space.loc:ppattach2code2ppattach1p 5



Name: ppattach ppattach.py

Project

Browse...

ppattach...

Main Module

ace_loc:ppattach/code/ppattach.py}

Browse...

${worksp

H& !ow go to the Arguments "ab K #orking directory

Change #orking directory to Other: 34wor/space.loc:ppattach5
C ick Apply button and then Glose button

Start the system by

Jo to' Run K Run As K 3 Python Run K Select code2ppattach&py

Gommand line arguments can be added by going to the menu'

Run K Run Gonfigurations -L Arguments "ab K Program arguments

-n this area) for example) you can type 0-h1) then click 0Apply1 and 0Run1

An explanation of valid command line arguments should be listed in the console& "his is your main way to interface with PPattach&

! " " #

$ % &

All development should be done in code2additional+eatures&py& A dummy feature has been given in this file& Mou may call the feature(s* whatever you want) but ensure that the res$lts dictionary uses the features6 name as a key& "he python dictionary res$lts is an instance variable of code2features&py and is inherited by code2additional+eatures&py

-deas for feature development may include'

Analy/ing a specific preposition and creating relevant features to capture this analysis (what was done with 6with6 738*'

- in
- for
- on
- from
- to

5tili/ing or improving on existing lexical ontologies in creating new features

- #ordnet)
- !omlex)
- !om ank)
- Propbank &&&