Appendix B: Apache OpenNLP in R

Introduction

An interface to the Apache OpenNLP tools (version 1.5.3). The Apache OpenNLP library is a machine learning based toolkit for the processing of natural language text written in Java. It supports the most common NLP tasks, such as tokenization, sentence segmentation, part-of-speech tagging, named entity extraction, chunking, parsing, and coreference resolution. See OpenNLP for more information

Maxent_Chunk_Annotator

Apache OpenNLP based chunk annotators

Generate an annotator which computes chunk annotations using the Apache OpenNLP Maxent chunker.

```
require(rJava)
```

```
## Loading required package: rJava
```

```
require(NLP)
```

```
## Loading required package: NLP
```

```
require(openNLP)
```

```
## Loading required package: openNLP
```

```
## Requires package 'openNLPmodels.en' from the repository at
## <http://datacube.wu.ac.at>.
## Some text.
s <- paste(c("Pierre Vinken, 61 years old, will join the board as a ",
             "nonexecutive director Nov. 29.\n",
             "Mr. Vinken is chairman of Elsevier N.V., ","the Dutch publishing group."),
           collapse = "")
s <- as.String(s)
## Chunking needs word token annotations with POS tags.
sent_token_annotator <- Maxent_Sent_Token_Annotator()
word_token_annotator <- Maxent_Word_Token_Annotator()
pos_tag_annotator <- Maxent_POS_Tag_Annotator()
a3 <- annotate(s,
               list(sent_token_annotator,
                    word_token_annotator,
                    pos_tag_annotator))
annotate(s, Maxent_Chunk_Annotator(), a3)
```

```
##  id type     start end features
##   1 sentence     1  84 constituents=<<integer,18>>
##   2 sentence    86 153 constituents=<<integer,13>>
##   3 word         1   6 POS=NNP, chunk_tag=B-NP
##   4 word         8  13 POS=NNP, chunk_tag=I-NP
##   5 word        14  14 POS=,, chunk_tag=O
##   6 word        16  17 POS=CD, chunk_tag=B-NP
##   7 word        19  23 POS=NNS, chunk_tag=I-NP
##   8 word        25  27 POS=JJ, chunk_tag=B-ADJP
##   9 word        28  28 POS=,, chunk_tag=O
```

```
##  10 word        30  33 POS=MD, chunk_tag=B-VP
##  11 word        35  38 POS=VB, chunk_tag=I-VP
##  12 word        40  42 POS=DT, chunk_tag=B-NP
##  13 word        44  48 POS=NN, chunk_tag=I-NP
##  14 word        50  51 POS=IN, chunk_tag=B-PP
##  15 word        53  53 POS=DT, chunk_tag=B-NP
##  16 word        55  66 POS=JJ, chunk_tag=I-NP
##  17 word        68  75 POS=NN, chunk_tag=I-NP
##  18 word        77  80 POS=NNP, chunk_tag=B-NP
##  19 word        82  83 POS=CD, chunk_tag=I-NP
##  20 word        84  84 POS=., chunk_tag=O
##  21 word        86  88 POS=NNP, chunk_tag=B-NP
##  22 word        90  95 POS=NNP, chunk_tag=I-NP
##  23 word        97  98 POS=VBZ, chunk_tag=B-VP
##  24 word       100 107 POS=NN, chunk_tag=B-NP
##  25 word       109 110 POS=IN, chunk_tag=B-PP
##  26 word       112 119 POS=NNP, chunk_tag=B-NP
##  27 word       121 124 POS=NNP, chunk_tag=I-NP
##  28 word       125 125 POS=,, chunk_tag=O
##  29 word       127 129 POS=DT, chunk_tag=B-NP
##  30 word       131 135 POS=JJ, chunk_tag=I-NP
##  31 word       137 146 POS=NN, chunk_tag=I-NP
##  32 word       148 152 POS=NN, chunk_tag=I-NP
##  33 word       153 153 POS=., chunk_tag=O
```

```
annotate(s, Maxent_Chunk_Annotator(probs = TRUE), a3)
```

```
##  id type     start end features
##   1 sentence     1  84 constituents=<<integer,18>>
##   2 sentence    86 153 constituents=<<integer,13>>
##   3 word         1   6 POS=NNP, chunk_tag=B-NP, chunk_prob=0.9740431
##   4 word         8  13 POS=NNP, chunk_tag=I-NP, chunk_prob=0.9816025
##   5 word        14  14 POS=,, chunk_tag=O, chunk_prob=0.9863059
##   6 word        16  17 POS=CD, chunk_tag=B-NP, chunk_prob=0.9926662
##   7 word        19  23 POS=NNS, chunk_tag=I-NP, chunk_prob=0.9854421
##   8 word        25  27 POS=JJ, chunk_tag=B-ADJP, chunk_prob=0.9978292
##   9 word        28  28 POS=,, chunk_tag=O, chunk_prob=0.9909762
##  10 word        30  33 POS=MD, chunk_tag=B-VP, chunk_prob=0.979816
##  11 word        35  38 POS=VB, chunk_tag=I-VP, chunk_prob=0.9857121
##  12 word        40  42 POS=DT, chunk_tag=B-NP, chunk_prob=0.9932718
##  13 word        44  48 POS=NN, chunk_tag=I-NP, chunk_prob=0.9947529
##  14 word        50  51 POS=IN, chunk_tag=B-PP, chunk_prob=0.9717558
##  15 word        53  53 POS=DT, chunk_tag=B-NP, chunk_prob=0.9991619
##  16 word        55  66 POS=JJ, chunk_tag=I-NP, chunk_prob=0.9989155
##  17 word        68  75 POS=NN, chunk_tag=I-NP, chunk_prob=0.981308
##  18 word        77  80 POS=NNP, chunk_tag=B-NP, chunk_prob=0.8397682
##  19 word        82  83 POS=CD, chunk_tag=I-NP, chunk_prob=0.9913565
##  20 word        84  84 POS=., chunk_tag=O, chunk_prob=0.992369
##  21 word        86  88 POS=NNP, chunk_tag=B-NP, chunk_prob=0.9910283
##  22 word        90  95 POS=NNP, chunk_tag=I-NP, chunk_prob=0.9902959
##  23 word        97  98 POS=VBZ, chunk_tag=B-VP, chunk_prob=0.9888302
##  24 word       100 107 POS=NN, chunk_tag=B-NP, chunk_prob=0.993464
##  25 word       109 110 POS=IN, chunk_tag=B-PP, chunk_prob=0.9719827
##  26 word       112 119 POS=NNP, chunk_tag=B-NP, chunk_prob=0.9906478
##  27 word       121 124 POS=NNP, chunk_tag=I-NP, chunk_prob=0.9819624
```

```
##  28 word       125 125 POS=,, chunk_tag=O, chunk_prob=0.9897705
##  29 word       127 129 POS=DT, chunk_tag=B-NP, chunk_prob=0.995753
##  30 word       131 135 POS=JJ, chunk_tag=I-NP, chunk_prob=0.9758163
##  31 word       137 146 POS=NN, chunk_tag=I-NP, chunk_prob=0.9990291
##  32 word       148 152 POS=NN, chunk_tag=I-NP, chunk_prob=0.9973766
##  33 word       153 153 POS=., chunk_tag=O, chunk_prob=0.9986785
```

Maxent_Entity_Annotator

Apache OpenNLP based entity annotators

Generate an annotator which computes entity annotations using the Apache OpenNLP Maxent name finder.

```
## Requires package 'openNLPmodels.en' from the repository at
## <http://datacube.wu.ac.at>.
require("NLP")
## Some text.
s <- paste(c("Pierre Vinken, 61 years old, will join the board as a ",
"nonexecutive director Nov. 29.\n",
"Mr. Vinken is chairman of Elsevier N.V., ",
"the Dutch publishing group."),
collapse = "")
s <- as.String(s)
## Need sentence and word token annotations.
sent_token_annotator <- Maxent_Sent_Token_Annotator()
word_token_annotator <- Maxent_Word_Token_Annotator()
a2 <- annotate(s, list(sent_token_annotator, word_token_annotator))
## Entity recognition for persons.
entity_annotator <- Maxent_Entity_Annotator()
entity_annotator
```

```
## An annotator inheriting from classes
##   Simple_Entity_Annotator Annotator
## with description
##   Computes entity annotations using the Apache OpenNLP Maxent name
##   finder employing the default model for language 'en' and kind
##   'person'.
```

```
annotate(s, entity_annotator, a2)
```

```
##  id type     start end features
##   1 sentence     1  84 constituents=<<integer,18>>
##   2 sentence    86 153 constituents=<<integer,13>>
##   3 word         1   6
##   4 word         8  13
##   5 word        14  14
##   6 word        16  17
##   7 word        19  23
##   8 word        25  27
##   9 word        28  28
##  10 word        30  33
##  11 word        35  38
##  12 word        40  42
##  13 word        44  48
##  14 word        50  51
```

```
##  15 word        53  53
##  16 word        55  66
##  17 word        68  75
##  18 word        77  80
##  19 word        82  83
##  20 word        84  84
##  21 word        86  88
##  22 word        90  95
##  23 word        97  98
##  24 word       100 107
##  25 word       109 110
##  26 word       112 119
##  27 word       121 124
##  28 word       125 125
##  29 word       127 129
##  30 word       131 135
##  31 word       137 146
##  32 word       148 152
##  33 word       153 153
##  34 entity       1  13 kind=person
```

```
## Directly:
entity_annotator(s, a2)
```

```
##  id type   start end features
##  34 entity     1  13 kind=person
```

```
## And slice ...
s[entity_annotator(s, a2)]
```

```
## Pierre Vinken
```

```
## Variant with sentence probabilities as features.
annotate(s, Maxent_Entity_Annotator(probs = TRUE), a2)
```

```
##  id type     start end features
##   1 sentence     1  84 constituents=<<integer,18>>
##   2 sentence    86 153 constituents=<<integer,13>>
##   3 word         1   6
##   4 word         8  13
##   5 word        14  14
##   6 word        16  17
##   7 word        19  23
##   8 word        25  27
##   9 word        28  28
##  10 word        30  33
##  11 word        35  38
##  12 word        40  42
##  13 word        44  48
##  14 word        50  51
##  15 word        53  53
##  16 word        55  66
##  17 word        68  75
##  18 word        77  80
##  19 word        82  83
##  20 word        84  84
```

```
##  21 word        86  88
##  22 word        90  95
##  23 word        97  98
##  24 word       100 107
##  25 word       109 110
##  26 word       112 119
##  27 word       121 124
##  28 word       125 125
##  29 word       127 129
##  30 word       131 135
##  31 word       137 146
##  32 word       148 152
##  33 word       153 153
##  34 entity       1  13 kind=person, prob=0.9445758
```

Maxent_POS_Tag_Annotator

Apache OpenNLP based POS tag annotators

Generate an annotator which computes POS tag annotations using the Apache OpenNLP Maxent Part of Speech tagger.

```
require("NLP")
## Some text.
s <- paste(c("Pierre Vinken, 61 years old, will join the board as a ",
"nonexecutive director Nov. 29.\n",
"Mr. Vinken is chairman of Elsevier N.V., ",
"the Dutch publishing group."),
collapse = "")
s <- as.String(s)
## Need sentence and word token annotations.
sent_token_annotator <- Maxent_Sent_Token_Annotator()
word_token_annotator <- Maxent_Word_Token_Annotator()
a2 <- annotate(s, list(sent_token_annotator, word_token_annotator))
pos_tag_annotator <- Maxent_POS_Tag_Annotator()
pos_tag_annotator
```

```
## An annotator inheriting from classes
##   Simple_POS_Tag_Annotator Annotator
## with description
##   Computes POS tag annotations using the Apache OpenNLP Maxent
##   Part of Speech tagger employing the default model for language
##   'en'
```

```
a3 <- annotate(s, pos_tag_annotator, a2)
a3
```

```
##  id type     start end features
##   1 sentence     1  84 constituents=<<integer,18>>
##   2 sentence    86 153 constituents=<<integer,13>>
##   3 word         1   6 POS=NNP
##   4 word         8  13 POS=NNP
##   5 word        14  14 POS=,
##   6 word        16  17 POS=CD
##   7 word        19  23 POS=NNS
```

```
##   8 word        25  27 POS=JJ
##   9 word        28  28 POS=,
##  10 word        30  33 POS=MD
##  11 word        35  38 POS=VB
##  12 word        40  42 POS=DT
##  13 word        44  48 POS=NN
##  14 word        50  51 POS=IN
##  15 word        53  53 POS=DT
##  16 word        55  66 POS=JJ
##  17 word        68  75 POS=NN
##  18 word        77  80 POS=NNP
##  19 word        82  83 POS=CD
##  20 word        84  84 POS=.
##  21 word        86  88 POS=NNP
##  22 word        90  95 POS=NNP
##  23 word        97  98 POS=VBZ
##  24 word       100 107 POS=NN
##  25 word       109 110 POS=IN
##  26 word       112 119 POS=NNP
##  27 word       121 124 POS=NNP
##  28 word       125 125 POS=,
##  29 word       127 129 POS=DT
##  30 word       131 135 POS=JJ
##  31 word       137 146 POS=NN
##  32 word       148 152 POS=NN
##  33 word       153 153 POS=.
```

```
## Variant with POS tag probabilities as (additional) features.
head(annotate(s, Maxent_POS_Tag_Annotator(probs = TRUE), a2))
```

```
##  id type     start end features
##   1 sentence     1  84 constituents=<<integer,18>>
##   2 sentence    86 153 constituents=<<integer,13>>
##   3 word         1   6 POS=NNP, POS_prob=0.9476405
##   4 word         8  13 POS=NNP, POS_prob=0.9692841
##   5 word        14  14 POS=,, POS_prob=0.9884445
##   6 word        16  17 POS=CD, POS_prob=0.9926943
```

```
## Determine the distribution of POS tags for word tokens.
a3w <- subset(a3, type == "word")
tags <- sapply(a3w$features, `[[`, "POS")
tags
```

```
##  [1] "NNP" "NNP" ","   "CD"  "NNS" "JJ"  ","   "MD"  "VB"  "DT"  "NN"
## [12] "IN"  "DT"  "JJ"  "NN"  "NNP" "CD"  "."   "NNP" "NNP" "VBZ" "NN"
## [23] "IN"  "NNP" "NNP" ","   "DT"  "JJ"  "NN"  "NN"  "."
```

```
table(tags)
```

```
## tags
##   ,   .  CD  DT  IN  JJ  MD  NN NNP NNS  VB VBZ
##   3   2   2   3   2   3   1   5   7   1   1   1
```

```
## Extract token/POS pairs (all of them): easy.
sprintf("%s/%s", s[a3w], tags)
```

```
##  [1] "Pierre/NNP"      "Vinken/NNP"      ",/,"
```

```
##  [4] "61/CD"           "years/NNS"       "old/JJ"
##  [7] ",/,"             "will/MD"         "join/VB"
## [10] "the/DT"          "board/NN"        "as/IN"
## [13] "a/DT"            "nonexecutive/JJ" "director/NN"
## [16] "Nov./NNP"        "29/CD"           "./."
## [19] "Mr./NNP"         "Vinken/NNP"      "is/VBZ"
## [22] "chairman/NN"     "of/IN"           "Elsevier/NNP"
## [25] "N.V./NNP"        ",/,"             "the/DT"
## [28] "Dutch/JJ"        "publishing/NN"   "group/NN"
## [31] "./."
```

```
## Extract pairs of word tokens and POS tags for second sentence:
a3ws2 <- annotations_in_spans(subset(a3, type == "word"),
subset(a3, type == "sentence")[2L])[[1L]]
sprintf("%s/%s", s[a3ws2], sapply(a3ws2$features, `[[`, "POS"))
```

```
##  [1] "Mr./NNP"       "Vinken/NNP"    "is/VBZ"        "chairman/NN"
##  [5] "of/IN"         "Elsevier/NNP"  "N.V./NNP"      ",/,"
##  [9] "the/DT"        "Dutch/JJ"      "publishing/NN" "group/NN"
## [13] "./."
```

Maxent_Sent_Token_Annotator

Apache OpenNLP based sentence token annotators

Generate an annotator which computes sentence annotations using the Apache OpenNLP Maxent sentence detector.

```
require("NLP")
## Some text.
s <- paste(c("Pierre Vinken, 61 years old, will join the board as a ",
"nonexecutive director Nov. 29.\n",
"Mr. Vinken is chairman of Elsevier N.V., ",
"the Dutch publishing group."),
collapse = "")
s <- as.String(s)
sent_token_annotator <- Maxent_Sent_Token_Annotator()
sent_token_annotator
```

```
## An annotator inheriting from classes
##   Simple_Sent_Token_Annotator Annotator
## with description
##   Computes sentence annotations using the Apache OpenNLP Maxent
##   sentence detector employing the default model for language 'en'.
```

```
a1 <- annotate(s, sent_token_annotator)
a1
```

```
##  id type     start end features
##   1 sentence     1  84
##   2 sentence    86 153
```

```
## Extract sentences.
s[a1]
```

```
## [1] "Pierre Vinken, 61 years old, will join the board as a nonexecutive director Nov. 29."
## [2] "Mr. Vinken is chairman of Elsevier N.V., the Dutch publishing group."
```

```
## Variant with sentence probabilities as features.
annotate(s, Maxent_Sent_Token_Annotator(probs = TRUE))
```

```
##  id type     start end features
##   1 sentence     1  84 prob=0.9998197
##   2 sentence    86 153 prob=0.9968879
```

Maxent_Word_Token_Annotator

Apache OpenNLP based word token annotators

Generate an annotator which computes word token annotations using the Apache OpenNLP Maxent tokenizer

```
require("NLP")
## Some text.
s <- paste(c("Pierre Vinken, 61 years old, will join the board as a ",
"nonexecutive director Nov. 29.\n",
"Mr. Vinken is chairman of Elsevier N.V., ",
"the Dutch publishing group."),
collapse = "")
s <- as.String(s)
## Need sentence token annotations.
sent_token_annotator <- Maxent_Sent_Token_Annotator()
a1 <- annotate(s, sent_token_annotator)
word_token_annotator <- Maxent_Word_Token_Annotator()
word_token_annotator
```

```
## An annotator inheriting from classes
##   Simple_Word_Token_Annotator Annotator
## with description
##   Computes word token annotations using the Apache OpenNLP Maxent
##   tokenizer employing the default model for language 'en'.
```

```
a2 <- annotate(s, word_token_annotator, a1)
a2
```

```
##  id type     start end features
##   1 sentence     1  84 constituents=<<integer,18>>
##   2 sentence    86 153 constituents=<<integer,13>>
##   3 word         1   6
##   4 word         8  13
##   5 word        14  14
##   6 word        16  17
##   7 word        19  23
##   8 word        25  27
##   9 word        28  28
##  10 word        30  33
##  11 word        35  38
##  12 word        40  42
##  13 word        44  48
##  14 word        50  51
##  15 word        53  53
##  16 word        55  66
##  17 word        68  75
##  18 word        77  80
```

```
##  19 word        82  83
##  20 word        84  84
##  21 word        86  88
##  22 word        90  95
##  23 word        97  98
##  24 word       100 107
##  25 word       109 110
##  26 word       112 119
##  27 word       121 124
##  28 word       125 125
##  29 word       127 129
##  30 word       131 135
##  31 word       137 146
##  32 word       148 152
##  33 word       153 153
```

```
## Variant with word token probabilities as features.
head(annotate(s, Maxent_Word_Token_Annotator(probs = TRUE), a1))
```

```
##  id type     start end features
##   1 sentence     1  84 constituents=<<integer,18>>
##   2 sentence    86 153 constituents=<<integer,13>>
##   3 word         1   6 prob=1
##   4 word         8  13 prob=0.9770575
##   5 word        14  14 prob=1
##   6 word        16  17 prob=1
```

```
## Can also perform sentence and word token annotations in a pipeline:
a <- annotate(s, list(sent_token_annotator, word_token_annotator))
head(a)
```

```
##  id type     start end features
##   1 sentence     1  84 constituents=<<integer,18>>
##   2 sentence    86 153 constituents=<<integer,13>>
##   3 word         1   6
##   4 word         8  13
##   5 word        14  14
##   6 word        16  17
```

Parse_Annotator

Apache OpenNLP based parse annotator

Generate an annotator which computes Penn Treebank parse annotations using the Apache OpenNLP chunking parser for English.

```
## Requires package 'openNLPmodels.en' from the repository at
## <http://datacube.wu.ac.at>.
require("NLP")
## Some text.
s <- paste(c("Pierre Vinken, 61 years old, will join the board as a ",
"nonexecutive director Nov. 29.\n",
"Mr. Vinken is chairman of Elsevier N.V., ",
"the Dutch publishing group."),
collapse = "")
s <- as.String(s)
```

```
## Need sentence and word token annotations.
sent_token_annotator <- Maxent_Sent_Token_Annotator()
word_token_annotator <- Maxent_Word_Token_Annotator()
a2 <- annotate(s, list(sent_token_annotator, word_token_annotator))
parse_annotator <- Parse_Annotator()
## Compute the parse annotations only.
p <- parse_annotator(s, a2)
## Extract the formatted parse trees.
ptexts <- sapply(p$features, `[[`, "parse")
ptexts
```

```
## [1] "(TOP (S (NP (NP (NNP Pierre) (NNP Vinken))(, ,) (ADJP (NP (CD 61) (NNS years)) (JJ old)))(, ,)
## [2] "(TOP (S (NP (NNP Mr.) (NNP Vinken)) (VP (VBZ is) (NP (NP (NN chairman)) (PP (IN of) (NP (NP (NNI
```

```
## Read into NLP Tree objects.
ptrees <- lapply(ptexts, Tree_parse)
ptrees
```

```
## [[1]]
## (TOP
##   (S
##     (NP
##       (NP (NNP Pierre) (NNP Vinken))
##       (, ,)
##       (ADJP (NP (CD 61) (NNS years)) (JJ old)))
##     (, ,)
##     (VP
##       (MD will)
##       (VP
##         (VB join)
##         (NP (DT the) (NN board))
##         (PP
##           (IN as)
##           (NP
##             (NP (DT a) (JJ nonexecutive) (NN director))
##             (NP (NNP Nov.) (CD 29))))))
##     (. .)))
##
## [[2]]
## (TOP
##   (S
##     (NP (NNP Mr.) (NNP Vinken))
##     (VP
##       (VBZ is)
##       (NP
##         (NP (NN chairman))
##         (PP
##           (IN of)
##           (NP
##             (NP (NNP Elsevier) (NNP N.V.))
##             (, ,)
##             (NP (DT the) (JJ Dutch) (NN publishing) (NN group))))))
##     (. .)))
```