Natural Language Tools for Information Extraction for Soft Target Exploitation and Fusion[*] Final Report for Letter Subcontract No. S690000049

Stuart C. Shapiro and Shane Axtell
Department of Computer Science and Engineering
and Center for Cognitive Science
201 Bell Hall
University at Buffalo, The State University of New York
Buffalo, NY 14260-2000
shapiro@cse.buffalo.edu
axtell@cse.buffalo.edu

February 2, 2007

[*] The work reported herein was supported, in part, by SRA International Inc. through Letter Subcontract No. S690000049

Contents

1 Introduction

The task of Information Extraction for Soft Target Exploitation and Fusion (STEF) is to input a set of messages originating from human intelligence (HUMINT), and output a sequence of database records or frames. For example, from the input messages

> 03/17/07 - Khalid Sattar arrived at 0930.
> Mohammed Janabi arrived at 1035.

output the sequence

> (m2 (date (m1 (day 17) (month 3) (year 7))))
> (m3! (act arrive) (agent Sattar Khalid) (time 930))
> (m4! (act arrive) (agent Janabi Mohammed) (time 1035))

We believe that in order to do this properly, each "word" in the input messages must be reduced to its lexical root (for example, "arrived" was reduced to "arrive" above), and its part of speech must be identified. Then, the resulting message must be parsed into its principal parts (such as *agent, act,* and *time.*)

This paper reports on our study of the capabilities and features of a number of natural language processing (NLP) tools for the text analysis that is required for the information extraction task outlined above. Discussion and evaluation of the individual tools is done immediately following the presentation of their capabilities. For §**??** and §**??** we have included demonstration runs of the given tools. These tools are freely available, which allowed us to download and test them. The tools in §**??** are all proprietary software and, therefore, were not available to us for this project. The tools in §**??** and §**??** are also freely available but since they do not automatically process text, we did not download and test them.

The small corpora used in the demonstration runs are contained in `/projects/shapiro/STEF/SampleData/SampleData2.txt` and `/projects/shapiro/STEF/SampleData/SampleData3.txt`. `SampleData3.txt` consists of the sample sentences provided by Richard Antony, of SAIC. This file has been manually tokenized and formatted with one sentence per line. It contains a total of 410 tokens and 361 words (we included dates as "words"). `SampleData2.txt` consists of a small list of inflected and uninflected words used in testing the stemmers.

In the body of this paper, we first discuss part of speech taggers (POS taggers). POS taggers take in a string of text and label (a.k.a. "tag") each word with its part of speech. In this task we used the 410 tokens of `SampleData3.txt` to measure recall, precision, and overall performance of each tool. Because these tools tag punctuation as well as words, it was appropriate to include all of the tokens.

Second, we discuss stemmers and lemmatizers. Stemmers are designed to take in a wordform as text and reduce it character-by-character to its stem (where a "stem" is the main morpheme of a word [**?**]). For the stemmers we used the 361 words in `SampleData3.txt` to measure recall, precision, and overall performance, because these tools do not stem punctuation, nor is it possible to do so. Lemmatizers take in a wordform as text and return it to its basic form (where a lemma is a set of word forms that have the same stem, the same part of speech, and the same word-sense [**?**]; by "basic" we mean the uninflected and underived form of a word such as "beginning" → "begin", or "rode" → "ride", for example). We used the 361 words in `SampleData3.txt` to measure recall, precision, and overall performance because, like the stemmers, these tools do not lemmatize punctuation, nor is it possible to do so. The lemmatizers also tag words with their part of speech and punctuation as punctuation marks. For these capabilities, we used the 410 tokens of `SampleData3.txt` to measure recall, precision, and overall performance.

Third, we discuss tools that process text automatically. The descriptions of each of these tools consist of information that was taken from the individual websites. Fourth, we discuss tools that are trained by humans that then process text automatically. These tools require the annotator to manually tag the first instance of each word in a text; they then search the rest of the corpus and label subsequent instances of that text according to the manually tagged words. We have not demonstrated these tools in this document because they require manual tagging as a preliminary stage to the processing of the text (the desired output for the STEF project requires that the text be automatically tagged and analyzed). Fifth, we discuss a few tools that are used for human (manual) annotation of corpora. These tools are not demonstrated in this document because, once again, they require manual annotation. Finally, since some of the tools discussed in this report assume access to a dictionary, we discuss one large online dictionary.

List of tools to be discussed

POS taggers:

Brill Tagger
CLAWS4

Stemmers and Lemmatisers:

Lovins Stemmer
Porter Stemmer
Lingua::Stem
MontyLingua
FreeLing
The Joy Lemmatizer

Tools that automatically process text:

AeroText
Global Name Recognition
IdentiFinder
Intelligent Miner for Text
NetOwl
Thing Finder

Human trained tools:

Alembic Natural Language Processing System
Annotate

Tools for manual annotation of text:

ACE
Callisto
Ellogon
GATE

Online Dictionaries:

WordNet

2 Part of speech taggers

2.1 Transformation-based Part of Speech Tagger

The Transformation-based Part of Speech Tagger (a.k.a. Brill Tagger) is available for free download at `http://www.cs.jhu.edu/˜brill/`. As the name implies, this tool is only used to tag parts of speech.

The Transformation-based Part of Speech Tagger is a fully automatic, trainable tagger that takes a corpus as input and outputs the corpus tagged with the parts of speech. Each lexical item is given the most likely tag first. This tag is derived from a partially ordered list of possible tags for each word (the list was compiled using a training corpus). Then, lexical rules and contextual transformations determine the accuracy of the tag given. Unknown words are first tagged as nouns and then affixes and adjacency cooccurence are used to determine the most likely/appropriate tag. [**?**]

2.1.1 Demo run

To execute the tagger, enter the directory in which the tagger is installed and do the following:

- `cd RULE_BASED_TAGGER_V1.14/Bin_and_Data`
- `tagger LEXICON` *`some sample file to tag`* `BIGRAMS LEXICALRULEFILE CONTEXTUALRULEFILE`

There are three different lexica provided with the distribution of this tagger: `LEXICON.BROWN.AND.WSJ`, `LEXICON.BROWN`, and `LEXICON.WSJ`. These are lists of words in the form [word tag$_1$ tag$_2$ ...tag$_n$] where tag$_1$ is the most likely tag for an encountered "word" and tag$_2$ ...tag$_n$ are other taggings of the "word" in the training corpora [**?**]. `LEXICON` in the command line is a link to `LEXICON.BROWN.AND.WSJ` which is a combination of the Brown corpus and the Penn Treebank tagging of *The Wall Street Journal* (a three million word corpus) [**?**]. The `tagger` uses the information in the lexica to apply tag$_1$ to each encountered word (i.e., each newly encountered word initially receives tag$_1$; the rest of the process is explained below).

`Some sample file to tag` is the corpus or small file that receives the tags. This corpus must be tokenized and formatted so that each sentence is on a single line. Otherwise, the tagger will tag periods (.), and other punctuation, along with the preceding word.

`BIGRAMS` is a list of adjacency pairs contained in the training corpus which are used to apply adjacency transformations of the type: "change the tag from X to Y if word Z ever appears to the right" [**?**]. However, in this distribution `BIGRAMS` is set to a dummy list as a place holder in the command line because `BIGRAMS` are only used when an encountered word is unkown. Since there is no portion of the Brown and WSJ text in the Penn Treebank that is not already tagged, `BIGRAMS` cannot be listed [**?**]. This is not explained well in the documentation. `BIGRAMS` (as well as the lexical rule file and contextual rule file) can be modified when more training material is available [**?**].

There are two lexical rule files provided with this distribution: `LEXICALRULEFILE.BROWN` and `LEXICALRULEFILE.WSJ`. These are lists of lexical rules used to apply initial tags on unknown words. The `LEXICALRULEFILE` in the command line is a link to `LEXICALRULEFILE.WSJ` [**?**]. The following are examples of lexical rules:

> If a word ends in "ed" give it the tag "VBN" (VBN = simple past tense verb).
> If a word ends in "s" give it the tag "NN" (NN = noun).

`CONTEXTUALRULEFILE` is a list of the contextually triggered transformations used to make the initial tags on unknown words more accurate [**?**]. The rules seem to specify in which environments a lexical item with a particular part of speech is allowed to occur. There are three such files provided with this distribution: `CONTEXTUALRULEFILE.BROWN`, `CONTEXTUALRULEFILE.WSJ`, and `CONTEXTUALRULEFILE.WSJ.NOLEX`. `CONTEXTUALRULEFILE` is a link to `CONTEXTUALRULEFILE.WSJ`.

The following is the output from the demo run (the formatting has been changed to make it more readable)[1]:

```
START STATE TAGGER::LEXICON READ
START STATE TAGGER::CORPUS READ
START STATE TAGGER::RULEFILE READ
START STATE TAGGER::BIGRAMS READ
ssssssssssssssssssssssssssssssssssssssssssssss
FINAL STATE TAGGER::READ IN OUTPUT FROM START
FINAL STATE TAGGER::READ IN LEXICON
ffffffffffffffffffffffffffffffffffffffffffffff


02/03/07/CD -/:  Source/NN says/VBZ that/IN Khalid/NNP Sattar/NNP ,/, a/DT
young/NN ,/, unemployed/JJ Sunni/NNP who/WP has/VBZ spoken/VBN angrily/RB
about/IN the/DT presence/NN of/IN U.S./NNP troops/NNS in/IN Iraq/NNP ,/,
has/VBZ been/VBN visiting/VBG a/DT house/NN on/IN Dhubat/NNP Street/NNP
```

[1] The tags are based on the Penn Treebank tagset and are shown in Appendix **??** below; a copy of the manual containing the glossary of the tags is available at `http://www.cis.upenn.edu/˜treebank/home.html`.

in/IN Adhamiya/NNP near/IN the/DT al-Assaf/JJ mosque/NN several/JJ times/NNS during/IN the/DT past/JJ week/NN ./.

Source/NN does/VBZ not/RB know/VB who/WP lives/VBZ there/RB ./.

02/12/07/CD -/: There/EX have/VBP been/VBN a/DT large/JJ number/NN of/IN "/" hits/NNS "/" for/IN the/DT web/NN site/NN of/IN Yarmuk/NNP Hospital/NNP in/IN the/DT Yarmuk/NNP district/NN of/IN Baghdad/NNP ./.

02/24/07/CD -/: Source/NN says/VBZ that/IN Khalid/NNP Sattar/NNP is/VBZ applying/VBG for/IN a/DT job/NN as/IN an/DT ambulance/NN driver/NN at/IN Yarmuk/NNP Hospital/NNP in/IN Baghdad/NNP ./.

Source/NN said/VBD that/IN Sattar/NNP had/VBD never/RB shown/VBN any/DT prior/JJ interest/NN in/IN health/NN care/NN or/CC emergency/NN services/NNS ./.

2/27/07/CD -/: The/DT principal/NN of/IN Ali/NNP Primary/NNP School/NNP on/IN Bayaa/NNP Street/NNP in/IN Bayaa/NNP reports/VBZ a/DT suspicious/JJ visit/NN by/IN parents/NNS of/IN a/DT prospective/JJ student/NN ./.

The/DT parents/NNS were/VBD not/RB interested/JJ in/IN the/DT school/NN curriculum/NN but/CC instead/RB were/VBD asking/VBG specific/JJ questions/NNS about/IN the/DT building/NN ./.

03/02/07/CD -/: Detainee/NNP Mohammed/NNP Janabi/NNP ,/, an/DT al-Qaeda/NN in/IN Iraq/NNP member/NN ,/, reports/NNS that/WDT suspected/VBD terrorist/JJ cell/NN leader/NN Ziyad/NNP al-Obeidi/NN talked/VBD last/JJ fall/NN about/IN using/VBG liquid/JJ bombs/NNS for/IN IED/VBN attacks/NNS against/IN coalition/NN forces/NNS ./.

03/14/07/CD -/: A/DT soldier/NN who/WP speaks/VBZ Arabic/NNP overheard/VBD a/DT conversation/NN in/IN a/DT restaurant/NN in/IN Baghdad/NNP concerning/VBG a/DT "/" new/JJ "/" type/NN of/IN weapon/NN ./.

Not/RB clear/JJ what/WP this/DT was/VBD or/CC who/WP may/MD have/VB it/PRP ./.

03/15/07/CD -/: Source/NN reports/NNS boasted/VBD to/TO him/PRP about/IN being/VBG in/IN charge/NN of/IN "/" something/NN very/RB big/JJ ./. "/"

03/17/07/CD -/: Surveillance/NN of/IN house/NN #23/CD on/IN Dhubat/NNP Street/NNP found/VBD that/IN Sufian/NNP Mashhad/NNP entered/VBD at/IN 0700/CD ./. Two/CD unknown/JJ males/NNS ,/, approximately/RB 18/CD to/TO 20/CD years/NNS old/JJ ,/, visited/VBD the/DT house/NN at/IN 0932/CD ./.

Another/DT male/NN ,/, identified/VBN as/IN Khalid/NNP Sattar/NNP ,/, arrived/VBD at/IN 1035/CD ./.

One/CD of/IN the/DT unidentified/JJ men/NNS left/VBN at/IN 1600/CD ,/, the/DT other/NN at/IN 1640/CD ./.

Sattar/NNP left/VBD at/IN 2130/CD ./.

03/20/07/CD -/: E-mail/NN message/NN from/IN an/DT unknown/JJ user/NN of/IN an/DT Internet/NNP cafe/NN in/IN Baghdad/NNP to/TO an/DT unknown/JJ recipient/NN in/IN an/DT Internet/NNP cafe/NN in/IN Karachi/NNP requested/VBD information/NN on/IN "/" that/DT stuff/NN "/" ./.

```
03/22/07/CD -/:  Two/CD unidentified/JJ males/NNS were/VBD seen/VBN walking/VBG
around/IN the/DT Ali/NNP Primary/NNP School/NNP in/IN Bayaa/NNP
taking/VBG pictures/NNS and/CC writing/VBG in/IN a/DT notepad/NN ./.

They/PRP ran/VBD away/RB when/WRB approached/VBN by/IN security/NN
guards/NNS ./.

03/24/07/CD -/:  A/DT map/NN found/VBD in/IN a/DT search/NN of/IN the/DT
home/NN of/IN Ziyad/NNP al-Obeidi/NN had/VBD red/JJ circles/NNS drawn/VBN
around/IN the/DT Ali/NNP Primary/NNP School/NNP in/IN Bayaa/NNP and/CC
Yarmuk/NNP Hospital/NNP in/IN Yarmuk/NNP ./.
```

2.1.2 Performance

Recall, precision, and overall performance percentages for the Brill Tagger's tagging of this sample data are given below:

- recall (the percentage of occurences that should have been tagged that were tagged): 100%
- precision (the percentage of occurrences that were tagged correctly): 96.83%
- overall (2*P*R/P+R): 98.39%

Brill (1994) [**?**] reports 97.2% accuracy after training the transformation-based tagger on 600,000 words of the Penn Treebank Wall Street Journal Corpus. The trained tagger was tested against 150,000 untagged words of the same corpus.

2.2 CLAWS4

CLAWS (the Constituent Likelihood Automatic Word-tagging System), has been under continuous production by UCREL since the early 1980s [**?**]. The latest version of CLAWS, CLAWS4, was used to tag nearly 100 million words of the British National Corpus. This tagger is not available as open source, but a free trial version is available through the internet on the CLAWS website [**?**]. The CLAWS taggers have consistently performed at 96-97% accuracy with an error rate of 1.5% and circa 3.3% of ambiguities unresolved [**?**]. There is no mention on the website of stemming and/or stemming capabilities.

2.2.1 Trial run

The input was copied from a pico version of `SampleData3.txt` and pasted into the appropriate window on the CLAWS website. After entering a personal email address, one must select either the smaller C5 output tagset or the larger C7 output tagset. The larger C7 tagset consists of a larger variety of tags (160, plus punctuation tags) than the smaller C5 tagset (only 60 tags; kept small to accomodate larger corpora), the larger version was, therefore, selected for this trial run[2]. One must also choose an output format; the authors chose the horizontal format for this demo.

The following is the output from the trial run (it has been formatted to make it more readable):

```
02/03/07_MF -_- Source_NN1 says_VVZ that_CST Khalid_NP1 Sattar_NP1 ,_, a_AT1 young_JJ
,_, unemployed_JJ Sunni_NN1 who_PNQS has_VHZ spoken_VVN angrily_RR about_II the_AT
presence_NN1 of_IO U.S._NP1 troops_NN2 in_II Iraq_NP1 ,_, has_VHZ been_VBN
visiting_VVG a_AT1 house_NN1 on_II Dhubat_NP1 Street_NNL1 in_II Adhamiya_NP1 near_II
the_AT al-Assaf_JJ mosque_NN1 several_DA2 times_NNT2 during_II the_AT past_JJ
week_NNT1 ._.

Source_NN1 does_VDZ not_XX know_VVI who_PNQS lives_VVZ there_RL ._.
```

[2] A glossary of the tags is in Appendix **??** below or available at `http://www.comp.lancs.ac.uk/ucrel/claws7tags.html`

02/12/07_MF -_- There_EX have_VH0 been_VBN a_AT1 large_JJ number_NN1 of_IO "_" hits_NN2 "_" for_IF the_AT web_NN1 site_NN1 of_IO Yarmuk_NP1 Hospital_NN1 in_II the_AT Yarmuk_NN1 district_NN1 of_IO Baghdad_NP1 ._.

02/24/07_MF -_- Source_NN1 says_VVZ that_CST Khalid_NP1 Sattar_NP1 is_VBZ applying_VVG for_IF a_AT1 job_NN1 as_II an_AT1 ambulance_NN1 driver_NN1 at_II Yarmuk_NP1 Hospital_NN1 in_II Baghdad_NP1 ._.

Source_NN1 said_VVD that_CST Sattar_NP1 had_VHD never_RR shown_VVN any_DD prior_JJ interest_NN1 in_II health_NN1 care_NN1 or_CC emergency_NN1 services_NN2 ._.

2/27/07_MF -_- The_AT principal_NN1 of_IO Ali_NP1 Primary_JJ School_NN1 on_II Bayaa_NP1 Street_NNL1 in_II Bayaa_NP1 reports_VVZ a_AT1 suspicious_JJ visit_NN1 by_II parents_NN2 of_IO a_AT1 prospective_JJ student_NN1 ._.

The_AT parents_NN2 were_VBDR not_XX interested_JJ in_II the_AT school_NN1 curriculum_NN1 but_CCB instead_RR were_VBDR asking_VVG specific_JJ questions_NN2 about_II the_AT building_NN1 ._.

03/02/07_MF -_- Detainee_NP1 Mohammed_NP1 Janabi_NP1 ,_, an_AT1 al-Qaeda_JJ in_II Iraq_NP1 member_NN1 ,_, reports_VVZ that_CST suspected_VVD terrorist_JJ cell_NN1 leader_NN1 Ziyad_NP1 al-Obeidi_NP1 talked_VVD last_MD fall_NN1 about_II using_VVG liquid_JJ bombs_NN2 for_IF IED_JJ attacks_NN2 against_II coalition_NN1 forces_NN2 ._.

03/14/07_MF -_- A_AT1 soldier_NN1 who_PNQS speaks_VVZ Arabic_NN1 overheard_VVD a_AT1 conversation_NN1 in_II a_AT1 restaurant_NN1 in_II Baghdad_NP1 concerning_II a_AT1 "_" new_JJ "_" type_NN1 of_IO weapon_NN1 ._.

Not_XX clear_RR what_DDQ this_DD1 was_VBDZ or_CC who_PNQS may_VM have_VHI it_PPH1 ._.

03/15/07_MF -_- Source_NN1 reports_NN2 boasted_VVD to_II him_PPHO1 about_II being_VBG in_II31 charge_II32 of_II33 "_" something_PN1 very_RG big_JJ ._. "_"

03/17/07_MF -_- Surveillance_NN1 of_IO house_NN1 #23_NNU on_II Dhubat_NP1 Street_NNL1 found_VVD that_CST Sufian_JJ Mashhad_NP1 entered_VVD at_II 0700_MC ._.

Two_MC unknown_JJ males_NN2 ,_, approximately_RR 18_MC to_II 20_MC years_NNT2 old_JJ ,_, visited_VVD the_AT house_NN1 at_II 0932_MC ._.

Another_DD1 male_NN1 ,_, identified_VVN as_CSA Khalid_NP1 Sattar_NP1 ,_, arrived_VVD at_II 1035_MC ._.

One_MC1 of_IO the_AT unidentified_JJ men_NN2 left_VVN at_II 1600_MC ,_, the_AT other_JJ at_II 1640_MC ._.

Sattar_NP1 left_VVD at_II 2130_MC ._.

03/20/07_MF -_- E-mail_NN1 message_NN1 from_II an_AT1 unknown_JJ user_NN1 of_IO an_AT1 Internet_NN1 cafe_NN1 in_II Baghdad_NP1 to_II an_AT1 unknown_JJ recipient_NN1 in_II an_AT1 Internet_NN1 cafe_NN1 in_II Karachi_NP1 requested_VVD information_NN1 on_II "_" that_DD1 stuff_NN1 "_" ._.

```
  03/22/07_MF -_- Two_MC unidentified_JJ males_NN2 were_VBDR seen_VVN walking_VVG
around_II the_AT Ali_NP1 Primary_JJ School_NN1 in_II Bayaa_NP1 taking_VVG pictures_NN2
and_CC writing_VVG in_II a_AT1 notepad_NN1 ._.

They_PPHS2 ran_VVD away_RL when_CS approached_VVN by_II security_NN1 guards_NN2 ._.

03/24/07_MF -_- A_AT1 map_NN1 found_VVN in_II a_AT1 search_NN1 of_IO the_AT home_NN1
of_IO Ziyad_NP1 al-Obeidi_NP1 had_VHD red_JJ circles_NN2 drawn_VVN around_II the_AT
Ali_NP1 Primary_JJ School_NN1 in_II Bayaa_NP1 and_CC Yarmuk_NP1 Hospital_NN1 in_II
Yarmuk_NP1 ._.
```

2.2.2 Performance

Recall, precision, and overall performance percentages for the CLAWS4 tagging of this sample data are given below:

- recall (the percentage of occurences that should have been tagged that were tagged): 100%
- precision (the percentage of occurrences that were tagged correctly): 98.05%
- overall (2*P*R/P+R): 99.02%

3 Stemmers and Lemmatizers

3.1 Lovins Stemmer

J.B. Lovins' stemmer is a very strong stemmer, meaning that it will, on average, remove more characters from a derived word than other stemmers (see [**?**] for a further explanation of stemmer strength). The perl code for this stemmer contains a list of possible suffixes in English and a pattern matching subroutine that subtracts away the suffixes according to a set of conditions.

3.1.1 Execution

To access and run the perl code for the Lovins stemmer, do the following:

- `cd` into the directory containing the code
- `perl Lovins-Perl.pl` *`some sample data to stem`*

3.1.2 Trial run

The input for this trial run was (`SampleData3.txt`). The output from the stemmer was the following:

```
02/03/07 - sourc say that khalis sattar , a young , unemploy sun wh ha spok angr
about th pres of u.s.  troop in iraq , ha been visit a hous on dhubat strees in
adhamiy near th al-assaf mosqu sever tim dur th past week .

sourc do not know wh liv ther .

02/12/07 - ther hav been a larg number of " hit " for th web sit of yarmuk hos-
pit in th yarmuk district of baghdad .

02/24/07 - sourc say that khalis sattar is appl for a job as an ambl driver at yarmuk
hospit in baghdad .
```

```
sourc said that sattar had never shown any prior interest in health car or emerg
servic .

2/27/07 - th princip of al prim school on baya strees in baya report a suspic visit
by parent of a prospect stud .

th parent wer not interest in th school curricl but instead wer ask specif quest
about th build .

03/02/07 - detaine moham janab , an al-qaed in iraq member , report that suspect
terror cel leader ziyad al-obeid talk last fal about using liquid bomb for ied at-
tack against coalit forc .

03/14/07 - a sold wh speak arab overheard a convers in a restaur in baghdad con-
cern a " new " typ of weapon .

not clear what th wa or wh may hav it .

03/15/07 - sourc report boast to him about being in charg of " someth ver big .
"

03/17/07 - surveil of hous #23 on dhubat strees found that suf mashhad enter at
0700 .  tw unknown mal , approxim 18 to 20 year old , visit th hous at 0932 .

another mal , identif as khalis sattar , arriv at 1035 .

on of th unidentif men left at 1600 , th other at 1640 .

sattar left at 2130 .

03/20/07 - e-mail mes from an unknown user of an internet caf in baghdad to an un-
known recipi in an internet caf in karach request inform on " that stuff " .

03/22/07 - tw unidentif mal wer seen walk around th al prim school in baya tak pic-
tur and writ in a notepad .

the ran awa when approach by secur guard .

03/24/07 - a map found in a search of th hom of ziyad al-obeid had red circl drawn
around th al prim school in baya and yarmuk hospit in yarmuk .
```

3.1.3 Performance

The Lovins stemmer, as mentioned earlier, is a very strong stemmer in that it often cuts off too many characters of a derived word in order to return it to its stem. Once again, [**?**] ranked it as a stronger stemmer than the Paice/Husk and Porter stemmers. The following are the performance scores for the Lovins Stemmer:

- recall (the percentage of occurences that should have been stemmed that were stemmed): 95.84%
- precision (the percentage of occurrences that were stemmed correctly): 76.45%
- overall (P*R*2/P+R): 85.05%

3.2 Porter Stemmer

The Porter stemmer (originally defined in [**?**]) is a less agressive stemmer than the Lovins stemmer (see [**?**] for detailed evaluations of each). Like the Lovins stemmer, the Porter stemmer code (in perl) has a list of possible derivational affixes and then subroutines that determine how much of the affixes to stem. In what follows it is apparent that the Porter stemmer trims fewer characters in the stemming process than the Lovins stemmer.

3.2.1 Trial run

In order to access the Porter stemmer and then execute the stemming process, do the following:

- `cd` into the directory containing the stemmer code
- `perl PorterStemmer.pl` *`some file to be stemmed`*

The input for this trial run was `SampleData3.txt`. The following is the output from the first trial run:

```
02/03/07 - sourc sai that khalid sattar , a young , unemploi sunni who ha spoken
angrili about the presenc of u.s.  troop in iraq , ha been visit a hous on dhu-
bat street in adhamiya near the al-assaf mosqu sever time dure the past week .

sourc doe not know who live there .

02/12/07 - there have been a larg number of " hit " for the web site of yarmuk hos-
pit in the yarmuk district of baghdad .

02/24/07 - sourc sai that khalid sattar is appli for a job as an ambul driver at
yarmuk hospit in baghdad .

sourc said that sattar had never shown ani prior interest in health care or emerg
servic .

2/27/07 - the princip of ali primari school on bayaa street in bayaa report a sus-
pici visit by parent of a prospect student .

the parent were not interest in the school curriculum but instead were ask specif
question about the build .

03/02/07 - detaine moham janabi , an al-qaeda in iraq member , report that sus-
pect terrorist cell leader ziyad al-obeidi talk last fall about us liquid bomb for
i attack against coalit forc .

03/14/07 - a soldier who speak arab overheard a convers in a restaur in baghdad
concern a " new " type of weapon .

not clear what thi wa or who mai have it .

03/15/07 - sourc report boast to him about be in charg of " someth veri big .  "

03/17/07 - surveil of hous #23 on dhubat street found that sufian mashhad enter
at 0700 .  two unknown male , approxim 18 to 20 year old , visit the hous at 0932
.

anoth male , identifi as khalid sattar , arriv at 1035 .
```

```
on of the unidentifi men left at 1600 , the other at 1640 .

sattar left at 2130 .

03/20/07 - e-mail messag from an unknown user of an internet cafe in baghdad to
an unknown recipi in an internet cafe in karachi request inform on " that stuff
" .

03/22/07 - two unidentifi male were seen walk around the ali primari school in bayaa
take pictur and write in a notepad .

thei ran awai when approach by secur guard .

03/24/07 - a map found in a search of the home of ziyad al-obeidi had red circl
drawn around the ali primari school in bayaa and yarmuk hospit in yarmuk .
```

3.2.2 Performance

- recall (the percentage of occurences that should have been stemmed that were stemmed): 93.91%
- precision (the percentage that were stemmed correctly): 82.55%
- overall (P*R*2/P+R): 87.86%

3.3 Lingua::Stem

Lingua::Stem was originally created as an external library to be added to other applications. However, by modifying the perl code it is also possible to simply output to STDOUT a list of stemmed words. Since the stemming of `SampleData3.txt` (and any sample file) would require rewriting the entire file directly into the perl code, we do not provide a trial run with `SampleData3.txt` but only of `SampleData2.txt` given in §**??**.

3.3.1 Performance

- Recall (the percentage of occurences that should have been stemmed that were stemmed): 100%
- Precision (the percentage of occurrences that were stemmed correctly): 62.50%
- Overall (P*R*2/P+R): 76.92%

3.4 The Porter Stemmer, Lovins Stemmer, and Lingua::Stem compared

The table below shows the output from the second trial run for both the Lovins and Porter Stemmers and the single trial run for Lingua::Stem (the data file used was `SampleData2.txt`):

Lovins	Lingua::Stem	Porter
bought	bought	bought
begin	begin	begin
went	went	went
going	go	go
team	team	team
wiv	wive	wive
fl	fli	fli
now	now	now
ar	ar	are
on	on	on
negoti	negoti	negoti
typ	type	typ
hap	happi	happi

As is apparent from the output of all three trial runs, the Lovins Stemmer seems to be the strongest followed by Lingua::Stem and the Porter Stemmer, respectively. While the Porter Stemmer is not as strong as the other two it still seems to over stem to some degree.

3.5 MontyLingua

MontyLingua is an automatic NLP tool that first tags input data with a tagger that the creator (Hugo Liu) claims exceeds the accuracy of the Transformation-based Part of Speech Tagger [**?**, **?**]. MontyLingua includes a ‘lemmatiser’ that strips the suffixes from plurals and verbs and returns the root form of the verb or noun. MontyLingua extracts verb-argument structures and phrases and returns a “digest” (i.e., a semantic interpretation) of the original sentence. This semantic interpretation digest includes the following:

- subject/verb/object tuples
- adjectives
- NPs and VPs
- people’s names, places, events, dates, and times
- and other semantic information [**?**]

3.5.1 Trial run 1

To execute the MontyLingua program do the following:

- `cd` into the directory where MontyLingua is installed
- `cd montylingua-2.1/python`
- either `./run.bat` *<some input file >some output file* for a large file or `./run.bat` to input single sentences at the command prompt

The program will run, tag the input file with POS tags, create its semantic digest, and output the result to whatever file the user specified. `SampleData3.txt` was the input file for this trial run and `/project/shapiro/STEF/MontyLingua/SampleData3.mrf` was the output file. The output includes, first, a tagged version of each input sentence (according to the Penn Treebank tagset); second, the semantic digest (which includes the lemmatised predicates; and third, a “Generated Summary” which is a reduced version of the input sentence (the documentation is fairly silent on the general structure of the output).

The following is the output from the trial run (it has been formatted to make it more readable). In order to conserve space, we have only included the first five sentences from `SampleData3.mrf`:

```
****** MontyLingua v.2.1 ******
**** by hugo@media.mit.edu *****
Lemmatiser OK!
Custom Lexicon Found!  Now Loading!
Fast Lexicon Found!  Now Loading!
Lexicon OK!
LexicalRuleParser OK!
ContextualRuleParser OK!
Commonsense OK!
Semantic Interpreter OK!
Loading Morph Dictionary!
********************************



 >
(NX 02/CD NX) :/: (NX 03/CD NX) :/: (NX 07/CD NX) -/: (NX Source/NN
NX) (VX says/VBZ VX) that/IN (NX Khalid/NNP Sattar/NNP NX) ,/, (NX
a/DT young/JJ ,/, unemployed/JJ Sunni/NNP NX) (NX who/WP NX) (VX
has/VBZ spoken/VBN angrily/RB VX) about/IN (NX the/DT presence/NN NX)
of/IN (NX U.S./NNP NX) (NX troops/NNS NX) in/IN (NX Iraq/NNP NX) ,/,
(VX has/VBZ been/VBN visiting/VBG VX) (NX a/DT house/NN NX) on/IN (NX
Dhubat/NNP Street/NNP NX) in/IN (NX Adhamiya/NNP NX) near/IN (NX
the/DT al-Assaf/JJ mosque/NN several/JJ times/NNS NX) during/IN (NX
the/DT past/JJ week/NN NX) ./.

 SENTENCE #1 DIGEST:

             adj_phrases: []

      adj_phrases_tagged: []

               modifiers: [’young’, ’unemployed’, ’angrily’]

        modifiers_tagged: [’young/JJ’, ’unemployed/JJ’, ’angrily/RB’]

            noun_phrases: [’02’, ’03’, ’07’, ’Source’, ’Khalid Sattar’,
                           ’young , unemployed Sunni’, ’who’,
                           ’presence’, ’U.S.’]

     noun_phrases_tagged: [’02/CD’, ’03/CD’, ’07/CD’, ’Source/NN’,’Khalid/NNP
                           Sattar/NNP’, ’young/JJ ,/, unemployed/JJ Sunni/NNP’,
                           ’who/WP’, ’presence/NN’, ’U.S./NNP’]

parameterized_predicates: [  [  [’say’, [’past_tense’]], [’Source’, []],
                                [’that Khalid Sattar’, [’prep=that’]]],
                             [  [’speak’, [’past_tense’]],
                                [’who’,[]],
                                [’about presence’,
                                  [’prep=about’, ’determiner=the’]],
                                [’of U.S.’, [’prep=of’]]]]

           prep_phrases: [’that Khalid Sattar’, ’about the presence’,
                          ’of U.S.’]
```

```
    prep_phrases_tagged: [’that/IN Khalid/NNP Sattar/NNP’, ’about/IN the/DT
                          presence/NN’, ’of/IN U.S./NNP’]

    verb_arg_structures: [  [’says/VBZ’, ’Source/NN’,
                              [’that/IN Khalid/NNP Sattar/NN P’]],
                            [’has/VBZ spoken/VBN angrily/RB’, ’who/ WP’,
                              [’about/IN presence/NN’, ’of/IN U.S./NNP’]]]

verb_arg_structures_concise: [’("say" "Source" "that Khalid Sattar")’,
                              ’("speak" "who" "about presence" "of
                              U.S.")’]

           verb_phrases: [’says’, ’has spoken angrily’]

    verb_phrases_tagged: [’says/VBZ’, ’has/VBZ spoken/VBN angrily/RB’]

 SENTENCE #2 DIGEST:

             adj_phrases: []

      adj_phrases_tagged: []

               modifiers: [’al-Assaf’, ’several’, ’past’]

        modifiers_tagged: [’al-Assaf/JJ’, ’several/JJ’, ’past/JJ’]

            noun_phrases: [’troops’, ’Iraq’, ’house’, ’Dhubat Street’,
                           ’Adhamiya’, ’al-Assaf mosque several
                           times’, ’past week’]

     noun_phrases_tagged: [’troops/NNS’, ’Iraq/NNP’, ’house/NN’,
                           ’Dhubat/NNP Street/NNP’, ’Adhamiya/NNP’,
                           ’al-Assaf/JJ mosque/NN several/JJ
                           times/NNS’, ’past/JJ week/NN’]

parameterized_predicates: [  [  [’visit’, [’past_tense’, ’passive_voice’]],
                                [’’, []],
                                [’house’, [’determiner=a’]],
                                [’on Dhubat Street’, [’prep=on’]],
                                [’in Adhamiya, [’prep=in’]],
                                [’near al-Assaf mosque several time’,
                                  [’prep=near’, ’determiner=the’, ’plural’]],
                                [’during past week’,
                                  [’prep=during’, ’determiner=the’]]]]

            prep_phrases: [’in Iraq’, ’on Dhubat Street’, ’in
                           Adhamiya’, ’near the al-Assaf mosque
                           several times’, ’during the past week’]

     prep_phrases_tagged: [’in/IN Iraq/NNP’, ’on/IN Dhubat/NNP
                           Street/NNP’, ’in/IN Adhamiya/NNP’, ’near/IN
                           the/DT al-Assaf/JJ
                           mosque/NN several/JJ times/NNS’, ’during/IN
                           the/DT past/JJ week/NN’]

     verb_arg_structures: [  [’has/VBZ been/VBN visiting/VBG’, ’’,
                               [’house/NN’, ’on/IN Dhubat/NNP Street/NNP’,
                                ’in/IN Adhamiya/NNP’,
                                ’near/IN al-Assaf/JJ mosque/NN several/JJ
                                times/NNS’, ’during/IN past/JJ
                                week/NN’]]]

verb_arg_structures_concise: [’("visit" "" "house" "on Dhubat Street"
                              "in Adhamiya" "near al-Assaf mosque
                              several time" "during past week")’]

            verb_phrases: [’has been visiting’]
     verb_phrases_tagged: [’has/VBZ been/VBN visiting/VBG’]

                    None
                          [  [’say’, ’Source’, ’that Khalid Sattar’],
                             [’speak’, ’who’, ’about presence’, ’of U.S.’],
                             [’visit’, ’’, ’house’, ’on Dhubat
                              Street’, ’in Adhamiya’, ’near al-Assaf
                              mosque several time’, ’during past week’]]

GENERATED SUMMARY:
Source said that Khalid Sattar.  Who spoke about presence of U.S..
Visited house on Dhubat Street in Adhamiya near al-Assaf mosque
several time during past week.
-- monty took 0.43 seconds. --

 >

(NX Source/NN NX) (VX does/VBZ not/RB know/VB VX) (NX who/WP NX) (NX
lives/NNS NX) there/RB ./.

 SENTENCE #1 DIGEST:

             adj_phrases: []

      adj_phrases_tagged: []

               modifiers: [’not’, ’there’]

        modifiers_tagged: [’not/RB’, ’there/RB’]

            noun_phrases: [’Source’, ’who’, ’lives’]

     noun_phrases_tagged: [’Source/NN’, ’who/WP’, ’lives/NNS’]

parameterized_predicates: [  [  [’not know’, [’negation’]],
                                [’Source’, []],
                                [’who’, []],
                                [’life’, [’plural’]]]]
            prep_phrases: []

     prep_phrases_tagged: []

     verb_arg_structures: [  ['does/VBZ not/RB know/VB', 'Source/NN',
                               ['who/WP', 'lives/NNS']]]

verb_arg_structures_concise: ['("not know" "Source" "who" "life")']

            verb_phrases: ['does not know']

     verb_phrases_tagged: ['does/VBZ not/RB know/VB']

                    None
                          [  ['not know', 'Source', 'who', 'life']]

GENERATED SUMMARY:
Source did not know who life.
-- monty took 0.05 seconds. --

 >

(NX 02/CD NX) :/: (NX 12/CD NX) :/: (NX 07/CD NX) -/: (NX There/EX NX)
(VX have/VBP been/VBN VX) (NX a/DT large/JJ number/NN NX) of/IN "/"
(NX hits/NNS NX) "/" for/IN (NX the/DT web/NN site/NN NX) of/IN (NX
Yarmuk/NNP Hospital/NNP NX) in/IN (NX the/DT Yarmuk/NNP district/NN
NX) of/IN (NX Baghdad/NNP NX) ./.

 SENTENCE #1 DIGEST:

             adj_phrases: []

      adj_phrases_tagged: []

               modifiers: ['large']

        modifiers_tagged: ['large/JJ']

            noun_phrases: ['02', '12', '07', 'There', 'large number',
                           'hits', 'web site', 'Yarmuk Hospital',
                           'Yarmuk district', 'Baghdad']

     noun_phrases_tagged: ['02/CD', '12/CD', '07/CD', 'There/EX',
                           'large/JJ number/NN', 'hits/NNS', 'web/NN
                           site/NN', 'Yarmuk/NNP Hospital/NNP',
                           'Yarmuk/NNP district/NN', 'Baghdad/NNP']

parameterized_predicates: [  [  ['be', []],
                                ['There', []],
                                ['large number', ['determiner=a']]]]

            prep_phrases: ['for the web site', 'of Yarmuk Hospital',
                           ’in the Yarmuk district’, ’of Baghdad’]

     prep_phrases_tagged: [’for/IN the/DT web/NN site/NN’, ’of/IN
                           Yarmuk/NNP Hospital/NNP’, ’in/IN the/DT
                           Yarmuk/NNP district/NN’, ’of/IN
                           Baghdad/NNP’]

     verb_arg_structures: [  [’have/VBP been/VBN’, ’There/EX’,
                               [’large/JJ number/NN’]]]

verb_arg_structures_concise: [’("be" "There" "large number")’]

            verb_phrases: [’have been’]

     verb_phrases_tagged: [’have/VBP been/VBN’]

                    None
                          [  [’be’, ’There’, ’large number’]]

GENERATED SUMMARY:
There was large number.
-- monty took 0.17 seconds. --

 >

(NX 02/CD NX) :/: (NX 24/CD NX) :/: (NX 07/CD NX) -/: (NX Source/NN
NX) (VX says/VBZ VX) that/IN (NX Khalid/NNP Sattar/NNP NX) (VX is/VBZ
applying/VBG VX) for/IN (NX a/DT job/NN NX) as/IN (NX an/DT
ambulance/NN driver/NN NX) at/IN (NX Yarmuk/NNP Hospital/NNP NX) in/IN
(NX Baghdad/NNP NX) ./.

 SENTENCE #1 DIGEST:

             adj_phrases: []

      adj_phrases_tagged: []

               modifiers: []

        modifiers_tagged: []

            noun_phrases: [’02’, ’24’, ’07’, ’Source’, ’Khalid
                           Sattar’, ’job’, ’ambulance driver’, ’Yarmuk
                           Hospital’, ’Baghdad’]

     noun_phrases_tagged: [’02/CD’, ’24/CD’, ’07/CD’, ’Source/NN’,
                           ’Khalid/NNP Sattar/NNP’, ’job/NN’,
                           ’ambulance/NN driver/NN’, ’Yarmuk/NNP
                           Hospital/NNP’, ’Baghdad/NNP’]

parameterized_predicates: [  [  [’say’, [’past_tense’]],
                                [’Source’, []],
                                [’that Khalid Sattar’, [’prep=that’]]],
                             [  ['apply', ['past_tense', 'passive_voice']],
                                ['Khalid Sattar', []],
                                ['for job', ['prep=for', 'determiner=a']],
                                ['as ambulance driver',
                                  ['prep=as', 'determiner=an']],
                                ['at Yarmuk Hospital', ['prep=at']],
                                ['in Baghdad', ['prep=in']]]]

             prep_phrases: ['that Khalid Sattar', 'for a job', 'as an
                            ambulance driver', 'at Yarmuk Hospital',
                            'in Baghdad']

      prep_phrases_tagged: ['that/IN Khalid/NNP Sattar/NNP', 'for/IN
                            a/DT job/NN', 'as/IN an/DT ambulance/NN
                            driver/NN', 'at/IN Yarmuk/NNP
                            Hospital/NNP', 'in/IN Baghdad/NNP']

      verb_arg_structures: [  ['says/VBZ', 'Source/NN',
                                ['that/IN Khalid/NNP Sattar/NNP']],
                              ['is/VBZ applying/VBG', 'Khalid/NNP Sattar/NNP',
                                ['for/IN job/NN', 'as/IN ambulance/NN
                                 driver/NN', 'at/IN Yarmuk/NNP Hospital/NNP',
                                 'in/IN Baghdad/NNP']]]

verb_arg_structures_concise: ['("say" "Source" "that Khalid Sattar")',
                               '("apply" "Khalid Sattar" "for job" "as
                               ambulance driver" "at Yarmuk
                               Hospital" "in Baghdad")']

             verb_phrases: ['says', 'is applying']

      verb_phrases_tagged: ['says/VBZ', 'is/VBZ applying/VBG']

                     None
                          [  ['say', 'Source', 'that Khalid Sattar'],
                             ['apply', 'Khalid Sattar', 'for job', 'as
                              ambulance driver', 'at Yarmuk Hospital',
                              'in Baghdad']]

GENERATED SUMMARY:
Source said that Khalid Sattar.  Khalid Sattar applied for job as
ambulance driver at Yarmuk Hospital in Baghdad.
-- monty took 0.22 seconds. --

 >

(NX Source/NN NX) (VX said/VBD VX) that/IN (NX Sattar/NNP NX) (VX
had/VBD never/RB shown/VBN VX) any/DT (VX prior/RB interest/VB VX)
in/IN (NX health/NN care/NN or/CC emergency/NN services/NNS NX) ./.

 SENTENCE #1 DIGEST:

              adj_phrases: []
      adj_phrases_tagged: []

               modifiers: [’never’, ’prior’]

        modifiers_tagged: [’never/RB’, ’prior/RB’]

            noun_phrases: [’Source’, ’Sattar’, ’health care or
                           emergency services’]

      noun_phrases_tagged: [’Source/NN’, ’Sattar/NNP’, ’health/NN
                           care/NN or/CC emergency/NN services/NNS’]

parameterized_predicates: [  [  [’say’, [’past_tense’]],
                                [’Source’, []],
                                [’that Sattar’, [’prep=that’]]],
                             [  [’show’, [’past_tense’]],
                                [’Sattar’, []]],
                             [  [’interest’, []],
                                [’’, []],
                                [’in health care or emergency service’,
                                  [’prep=in’, ’plural’]]]]

            prep_phrases: [’that Sattar’, ’in health care or emergency
                           services’]

      prep_phrases_tagged: [’that/IN Sattar/NNP’, ’in/IN health/NN
                           care/NN or/CC emergency/NN services/NNS’]

      verb_arg_structures: [  [’said/VBD’, ’Source/NN’,
                               [’that/IN Sattar/NNP’]],
                             [’had/VBD never/RB shown/VBN’,
                              ’Sattar/NNP’, []],
                             [’prior/RB interest/VB’, ’’,
                               [’in/IN health/NN care/NN or/CC
                                emergency/NN services/NNS’]]]

verb_arg_structures_concise: [’("say" "Source" "that Sattar")’,
                              ’("show" "Sattar" )’, ’("interest" ""
                              "in health care or emergency service")’]


            verb_phrases: [’said’, ’had never shown’, ’prior
                           interest’]

      verb_phrases_tagged: [’said/VBD’, ’had/VBD never/RB shown/VBN’,
                           ’prior/RB interest/VB’]

                    None
                          [  [’say’, ’Source’, ’that Sattar’],
                             [’show’, ’Sattar’],
                             [’interest’, ’’,
                              ’in health care or emergency service’]]
```

```
GENERATED SUMMARY:
Source said that Sattar showed .  Interested in health care or
emergency service.
-- monty took 0.13 seconds. --
```

3.5.2 Performance on POS Tagging

- Recall (the percentage of occurrences that should have been tagged that were tagged): 100%
- Precision (the percentage of occurrences that were tagged correctly): 96.34%
- Overall (P*R*2/P+R): 98.14%

3.5.3 Performance on Lemmatizing

- Recall (the percentage of occurrences that should have been lemmatized that were lemmatized): 82.83%
- Precision (the percentage of occurrences that were lemmatized correctly): 81.99%
- Overall (P*R*2/P+R): 82.41%

3.6 FreeLing

FreeLing is an open source software package licensed under the GNU Lesser General Public License of the Free Software Foundation. The latest version of FreeLing (1.5) has the following capabilities:

- text tokenization
- sentence splitting
- morphological analysis
- named entity detection
- date/number/currency/ratios recognition
- part of speech tagging
- chart-based shallow parsing
- contraction splitting
- improved suffix treatment, retokenization of clitic pronouns
- physical magnitudes detection (speed, weight, temperature, density, etc.)
- named entity classification
- wordNet based sense annotation
- dependency parsing [**?**]

FreeLing was developed at the TALP Research Center of the Universitat Politécnica de Catalunya. It was designed to be used as an external library with a simple main program that can be run from the command line [**?**].

FreeLing was originally developed for linguistic analysis of Spanish and Catalan data. Since then it has been extended to include English, Italian, and Galician. For morphological analysis and POS tagging in Spanish, Catalan, Italian, and Galician, FreeLing contains morphological dictionaries extracted from various corpora in said languages. For morphological analysis and POS tagging in English, FreeLing has a morphological dictionary which was extracted from the WSJ corpus. This FreeLing dictionary contains over 160,000 forms which correspond to 102,000 lemmas. [**?**]

3.6.1 Trial run 1

From the directory where the tool is installed do the following:

- `cd config`
- `analyzer -f en.cfg <SampleData3.txt >SampleData3.mrf`[?]

`analyzer` is the command to start the analysis process. It is unclear in the documentation what “`-f`” refers to. `en.cfg` is the default configuration file for English that gives the `analyzer` the configuration parameters for English. `<SampleData3.txt` is the input file for this trial run and `>SampleData3.mrf` is the output file for this trial run.

The input for this trial run was `SampleData3.txt`. The following is the output from the trial run (FreeLing also uses the Penn Treebank tagset; underlined items indicate incorrect taggings, incorrect lemmatising, or otherwise interesting phenomena):

```
02/03/07 [??:??/??/??:??.??:??]  W
- - Fg
Source source NN
says say VBZ
that that IN
Khalid_Sattar khalid_sattar NP
, , Fc
a 1 Z
young young JJ
, , Fc
unemployed unemployed JJR
Sunni sunni NP
who who WP
has have VBZ
spoken speak VBN
angrily angry RB
about about IN
the the DT
presence presence NN
of of IN
U.S. u.s.  NP
troops troops NNS
in in IN
Iraq iraq NP
, , Fc
has have VBZ
been be VBN
visiting visit VBG
a a DT
house house NN
on on IN
Dhubat_Street dhubat_street NP
in in IN
Adhamiya adhamiya NP
near near IN
the the DT
al-Assaf al-assaf JJ
mosque mosque NN
several several JJR
```

```
times times NNS
during during IN
the the DT
past past JJ
week week NN
. .  Fp

Source source NN
does do VBZ
not not RB
know know VBP
who who WP
lives live VBZ
there there NN
. .  Fp

02/12/07 [??:??/??/??:??.??:??]  W
- - Fg
There there NN
have have VBP
been be VBN
a a IN
large large JJ
number number NN
of of IN
" " Fe
hits hit NNS
" " Fe
for for IN
the the DT
web web NN
site site NN
of of IN
Yarmuk_Hospital yarmuk_hospital NP
in in IN
the the DT
Yarmuk yarmuk NP
district district NN
of of IN
Baghdad baghdad NP
. .  Fp

02/24/07 02/24/07 Z
- - Fg
Source source NN
says say VBZ
that that IN
Khalid_Sattar khalid_sattar NP
is be VBZ
applying apply VBG
for for IN
a a DT
job job NN
```

```
as as IN
an an DT
ambulance ambulance NN
driver driver NN
at at IN
Yarmuk_Hospital yarmuk_hospital NP
in in IN
Baghdad baghdad NP
. .  Fp

Source source NN
said say VBD
that that IN
Sattar sattar NP
had have VBD
never never RB
shown show VBN
any any DT
prior prior NN
interest interest NN
in in IN
health health NN
care care NN
or or CC
emergency emergency NN
services service NNS
. .  Fp

2/27/07 2/27/07 Z
- - Fg
The the DT
principal principal NN
of of IN
Ali_Primary_School ali_primary_school NP
on on IN
Bayaa_Street bayaa_street NP
in in IN
Bayaa bayaa NP
reports report VBZ
a 1 Z
suspicious suspicious JJ
visit visit NN
by by IN
parents parent NNS
of of IN
a a DT
prospective prospective JJR
student student NN
. .  Fp

The the DT
parents parent NNS
were be VBD
```

```
not not RB
interested interest VBN
in in IN
the the DT
school school NN
curriculum curriculum NN
but but CC
instead instead RB
were be VBD
asking ask VBG
specific specific JJ
questions question NNS
about about IN
the the DT
building building NN
. .  Fp

03/02/07 [??:??/??/??:??.??:??]  W
- - Fg
Detainee detainee NN
Mohammed_Janabi mohammed_janabi NP
, , Fc
an an DT
al-Qaeda al-qaeda NNP
in in IN
Iraq iraq NP
member member NN
, , Fc
reports report NNS
that that WDT
suspected suspect VBD
terrorist terrorist JJ
cell cell NN
leader leader NN
Ziyad ziyad NP
al-Obeidi al-obeidi NNP
talked talk VBD
last last JJ
fall fall NN
about about IN
using use VBG
liquid liquid NN
bombs bomb NNS
for for IN
IED ied NP
attacks attack NNS
against against IN
coalition coalition NN
forces forces NNS
. .  Fp

03/14/07 [??:??/??/??:??.??:??]  W
- - Fg
```

```
A a DT
soldier soldier NN
who who WP
speaks speak VBZ
Arabic arabic NP
overheard overhear VBD
a a DT
conversation conversation NN
in in IN
a a DT
restaurant restaurant NN
in in IN
Baghdad baghdad NP
concerning concern VBG
a 1 Z
" " Fe
new new JJ
" " Fe
type type NN
of of IN
weapon weapon NN
. .  Fp

Not not RB
clear clear JJ
what what WP
this this DT
was be VBD
or or CC
who who WP
may may MD
have have VBP
it it NN
. .  Fp

03/15/07 [??:??/??/??:??.??:??]  W
- - Fg
Source source NN
reports report NNS
boasted boast VBD
to to IN
him him PRP
about about IN
being be VBG
in in IN
charge charge NN
of of IN
" " Fe
something something NN
very very JJ
big big JJ
. .  Fp
" " Fe
```

```
03/17/07 [??:??/??/??:??.??:??]  W
- - Fg
Surveillance surveillance NN
of of IN
house house NN
#23 #23 Z
on on IN
Dhubat_Street dhubat_street NP
found find VBD
that that IN
Sufian_Mashhad sufian_mashhad NP
entered enter VBD
at at IN
0700 700 Z
. .  Fp

Two 2 Z
unknown unknown NN
males male NNS
, , Fc
approximately approximately RB
18 18 Z
to to IN
20 20 Z
years years NNS
old old JJ
, , Fc
visited visit VBD
the the DT
house house NN
at at IN
0932 932 Z
. .  Fp

Another another DT
male male NN
, , Fc
identified identify VBN
as as IN
Khalid_Sattar khalid_sattar NP
, , Fc
arrived arrive VBN
at at IN
1035 1035 Z
. .  Fp

One 1 Z
of of IN
the the DT
unidentified unidentified JJR
men men NN
left leave VBN
```

```
at at IN
1600 1600 Z
, , Fc
the the DT
other other JJ
at at IN
1640 1640 Z
. .  Fp

Sattar sattar NP
left leave VBD
at at IN
2130 2130 Z
. .  Fp

03/20/07 03/20/07 Z
- - Fg
E-mail e-mail NN
message message NN
from from IN
an an DT
unknown unknown NN
user user NN
of of IN
an an DT
Internet internet NP
cafe cafe NN
in in IN
Baghdad baghdad NP
to to IN
an an DT
unknown unknown NN
recipient recipient NN
in in IN
an an DT
Internet internet NP
cafe cafe NN
in in IN
Karachi karachi NP
requested request VBD
information information NN
on on IN
" " Fe
that that DT
stuff stuff NN
" " Fe
.  .  Fp

03/22/07 03/22/07 Z
- - Fg
Two 2 Z
unidentified unidentified JJR
males male NNS
```

```
were be VBD
seen see VBN
walking walk VBG
around around IN
the the DT
Ali_Primary_School ali_primary_school NP
in in IN
Bayaa bayaa NP
taking take VBG
pictures pictures NNS
and and CC
writing write VBG
in in IN
a a DT
notepad notepad NN
. .  Fp

They they PRP
ran run VBD
away away JJ
when when NN
approached approach VBN
by by IN
security security NN
guards guard NNS
. .  Fp

03/24/07 03/24/07 Z
- - Fg
A a DT
map map NN
found find VBD
in in IN
a a DT
search search NN
of of IN
the the DT
home home NN
of of IN
Ziyad ziyad NP
al-Obeidi al-obeidi NNP
had have VBD
red red JJ
circles circle NNS
drawn draw VBN
around around IN
the the DT
Ali_Primary_School ali_primary_school NP
in in IN
Bayaa bayaa NP
and and CC
Yarmuk_Hospital yarmuk_hospital NP
in in IN
```

```
Yarmuk yarmuk NP
. .  Fp
```

3.6.2 Performance on POS Tagging

- Recall (the percentage of occurrences that should have been tagged that were tagged): 100%
- Precision (the percentage of occurrences that were tagged correctly): 92.68%
- Overall (P*R*2/P+R): 96.20%

3.6.3 Performance on Lemmatizing

- Recall (the percentage of occurrences that should have been lemmatized that were lemmatized): 100%
- Precision (the percentage of occurrences that were lemmatized correctly): 96.12%
- Overall: 98.02%

3.6.4 Trial run 2

This trial run demonstrates the contraction splitting capabilities of FreeLing. To execute the trial run do the following:

- `cd config` from the top FreeLing directory
- `analyzer -f en.cfg <SampleData.txt >SampleData.mrf`

The following is the input file for this trial run:

```
A village northwest of Baghdad .
A tall man got into a car .
The man drove off to the west .
The man isn’t my brother .
He’s the brother-in-law of my sister’s husband .
Can’t you do any better ?
Why weren’t you here on time ?
```

The following is the output from the trial run:

```
The the DT
man man NN
isn’t isn’t RB
my my PP$
brother brother NN
. .  Fp


He he NN
’s ’s POS
the the DT
brother-in-law brother-in-law NN
of of IN
my my PP$
sister sister NN
’s ’s POS
```

```
husband husband NN
. .  Fp


Can’t can’t NP
you you PRP
do do VBP
any any DT
better better JJR
? ?  Fit


Why why NN
weren’t weren’t RB
you you PRP
here here JJ
on on IN
time time NN
? ?  Fit
```

The contraction splitter did not perform well on this trial run. It did not split the negated contractions on the auxiliary or modal verbs. It did split the contractions on “He’s” and “sister’s” but it incorrectly tagged the “’s” on “He’s” as a possessive suffix. Only one out of the five contractions was split and then tagged correctly.

3.7 The Joy Lemmatizer

The Joy Lemmatizer (a.k.a. the Joy Morphological Analyzer) is part of SNaLPS (the SNePS Natural Language Processing System) [**?**]. It was first developed in 1976 by Darrel Joy. The lemmatizer works by first searching for the input word in a lexicon which contains only the root forms of words (along with their part of speech information) and exceptional wordforms (e.g., “went” and “ran”). If the lexicon contains the input word then the lemmatizer assumes that the word is already in its root form and is, therefore, outputted with the part of speech information.

If the input word is not found in the lexicon, then the lemmatizer removes whatever suffixes may be attached to the word[3] and searches the lexicon again. At this point, the root form should be found in the lexicon unless prefixes are present on the word.

If the word does have prefixes, the lemmatizer replaces the suffixes, removes the prefixes, and searches the lexicon again. If the word is still not found, then the lemmatizer removes the suffixes once again and looks up the resulting word in the lexicon. At this point the root form should be found in the lexicon. The following example from [**?**] shows what wordforms are looked up in the lexicon, and in what order, for a word with both prefixes and suffixes,

```
UNDOUBTEDLY
UNDOUBTEDL
UNDOUBTED
UNDOUBTE
UNDOUBT
  DOUBTEDLY
  DOUBTEDL
  DOUBTED
  DOUBTE
  DOUBT (lookup succeeds)
```

[3] For words whose root form must change to accommodate certain affixes (e.g., “-y” → “-i” when adding “-es” in “tries” and other similar words) the change is returned to the original form (according to the morphological rules of English, which the Joy Lemmatizer is programmed to be able to handle)

When the word is found the root form is output with the part, or parts, of speech of the original word form.

3.7.1 Trial run

To run the lemmatizer, load the program at the Common Lisp prompt, and evaluate the Lisp form

```
(parseFile <input file> <output file> :grammar <grammar file>
                                      :lexicon <lexicon file>)
```

The input words for this trial run were taken from `SampleData2.txt`. The following is the output from the trial run, slightly edited for readability:

```
"bought"
(buy v)

"beginning"
(begin v)
(beginning adj)

"went"
(go v)

"going"
(go v)
(going adj)

"teams"
(team n)
(team v)

"wives"
(wife n)
(wife v)

"flies"
(fly n)
(fly v)

"now"
(now adj)

"are"
(be v)
(be aux)

"ones"
(ones adv)

"negotiation"
(negotiation n)

"types"
(type n)
(type v)
```

```
"happiness"
(happiness n)
```

3.7.2 Performance

Recall, precision, and overall performance for the Joy Lemmatizer are given below:

- recall (the percentage of occurences that should have been lemmatized that were lemmatized): 100%
- precision (the percentage of occurrences that were lemmatized correctly): 98.05%
- overall (2*P*R/P+R): 92.3%

4 Tools that automatically process text

4.1 AeroText (Lockheed Martin)

AeroText(TM) is an information extraction tool used to deal with data overload. AeroText(TM) is not free software. This tool has the following capabilities:

- location normalization to GIS
- multilingual extraction
- BlockFinderTM Product—patented table processing
- named entity recognition
- entity association
- entity co-reference resolution
- grammatical phrase recognition
- event extraction
- topic categorization
- temporal reasoning [**?**]

It is not clear from the website whether AeroText(TM) can do stemming and tense/aspect resolution.

4.2 Global Name Recognition (IBM)

The Global Name Recognition software suite from IBM is capable of recognizing and scoring names in dozens of languages. This software suite allows you to “search, recognize, and manage multicultural names, screen potential threats, and perform background checks across multiple geographies and cultures” [**?**]. It does not provide tagging functions or stemming and tense/aspect analysis.

4.3 IdentiFinder (BBN/Verizon)

IdentiFinder is a proprietary software package capable of finding the following in text or in speech that has been transcribed: people, places, companies, numbers, dates, amounts, and other categories that must be defined by the human user [**?**]. This software package does not do stemming or tense/aspect resolution.

4.4 Intelligent Miner for Text (IBM)

Intelligent Miner for Text is a proprietary software with a 60 day free trial available. It is capable of the following: "…discover in which language a document is written, and it can extract names, multiword terms, abbreviations, and other vocabulary such as dates, figures, and amounts. It extracts patterns, organizes documents by subject, finds predominant themes, and searches for relevant documents" [**?**]. This software is not capable of stemming or tense/aspect resolution.

4.5 NetOwl (SRA)

NetOwl is another proprietary software package used for text extraction and text mining. NetOwl is capable of extracting the following: people, organizations, places, artifacts, phone, social security numbers, dates, and addresses [**?**]. NetOwl is also capable of distinguishing between entities and events (i.e. nouns/names and verbs) but does not do stemming or tense/aspect resolution (at least, it was unclear from the website whether these were possible).

4.6 Thing Finder (Inxight)

Thing Finder is a proprietary software package used for entity extraction. It is capable of extracting people, names, dates, companies, company names, and "other things" [**?**]. Thing Finder does not, however, have the functionality to stem verbs and analyze their tense/aspect.

5 Human trained tools

5.1 Alembic Natural Language Processing System

Alembic is a natural language processing system used for manual annotation of linguistic data and for automatic tagging of the following linguistic phenomena:

- sentences
- part of speech
- date
- place names
- time [**?**]

Alembic can also do tokenization and named entity tagging in English and a host of other languages [**?**]. Alembic's extraction tasks include the following entities:

- named entities
- title and position
- sentence "chunks" (noun/verb groups)
- MUC6-style coreference
- template elements
- template relations
- scenario templates

Alembic does not stem verbs or analyze their tense and aspect (at least automatically).

5.2 Annotate

Annotate is a semi-automatic NLP tool used for annotation of corpus data. This tool is capable of tagging the following linguistic phenomena:

- part of speech
- morphology
- phrase category
- grammatical function
- syntactic category
- basic tokenization commands (for splitting or merging words, moving sentence boundaries, etc.) [**?**, **?**]

The type and number of the labels used are defined by the user. Although Annotate can label syntactic category it cannot stem verbs and analyze their tense and aspect. This software is free for academic use. Academics wishing to access this software for research purposes are required to complete and sign a license agreement which must be sent to Universität des Saarlands in Germany.

6 Tools for manual annotation of texts

6.1 ACE

The ACE tool is open source software available through the ACE (Automatic Content Extraction) project [**?**]. The ACE tool is a manual annotation package which relies on "color-coded underlining to display layers of annotation on spans of text" [**?**]. As a manual annotation tool, the ACE tool does not support stemming capabilities or tense/aspect resolution.

6.2 Callisto

Callisto is an open source software tool that is used for hand annotation of linguistic data. Its modular design "allow[s] for unique tag-set definitions and domain dependent interfaces" [**?**]. As a manual annotation tool, Callisto does not provide stemming and tense/aspect resolution.

6.3 Ellogon

Ellogon is an open source software package licensed under the GNU/GPL license [**?**]. Like GATE (below), Ellogon was developed under the TIPSTER data model allowing it to provide the following infrastructure:

- Managing, storing and exchanging textual data as well as the associated linguistic information.
- Creating, embedding and managing linguistic processing components.
- Facilitating communication among different linguistic components by defining a suitable programming interface (API).
- Visualising textual data and associated linguistic information. [**?**]

Ellogon is a referential/annotation-based platform (i.e. it stores the text separately from the linguistic information and has references back to the original text) [**?**], thus avoiding the problem of direct text manipulation.

6.4 GATE

GATE is an open source software tool with a graphical development environment [**?**]. GATE can be used to tag parts of speech, dialog/speech act, utterance boundaries, and to link entities within the text [**?**]. As a manual annotation tool it does not include stemming capabilities or tense/aspect resolution.

7 Online Dictionaries

7.1 WordNet

WordNet is an online lexical database of English. Nouns, adjectives, verbs, and adverbs are grouped into sets of synonyms (*synsets*)[4]. WordNet contains more than 118,000 word forms (*f*) and more than 90,000 word senses (*s*) (where each sense consists of all of the synonyms in the database for that sense) which combine to form more than 166,000 (*f*, *s*) pairs. Inflectional morphology is ignored in WordNet, meaning that if given the word "flew" at the command line, WordNet will return the information it has for "fly". Derivational and compound morphology for each form is entered into the database. For example, all of the derivations of "interpret" (e.g., "interpreter", "interpretation", etc.) have distinct word forms. [**?**]

WordNet contains the following sematic relations for each database entry:

- *Synonymy* is WordNet's basic relation, because WordNet uses sets of synonyms (*synsets*) to represent word senses. Synonymy (*syn* same, *onyma* name) is a symmetric relation between word forms.
- *Antonymy* (opposing-name) is also a symmetric semantic relation between word forms, especially important in organizing the meanings of adjectives and adverbs.
- *Hyponymy* (sub-name) and its inverse, *hypernymy* (super-name), are transitive relations between synsets. Because there is usually only one hypernym, this semantic relation organizes the meanings of nouns into a hierarchical structure.
- *Meronymy* (part-name) and its inverse, *holonymy* (whole-name), are complex sematic relations. WordNet deistinguishes *component* parts, *substantive* parts, and *member* parts.
- *Troponymy* (manner-name) is for verbs what hyponymy is for nouns, although the resulting hierarchies are much shallower.
- *Entailment* relations between verbs are also coded in WordNet. [**?**]

Each semantic relation is represented in WordNet by "pointers" between the word forms or the synsets [**?**]. A "pointer" is essentially an index number in the database corresponding to synset membership. For example, *vituperate* and *revile* have the following pointer values:

```
vituperate%2:32:00::  00838910 1 0
revile%2:32:00::  00838910 1 1
```

It is unclear from the documentation what each number/symbol in the above examples corresponds to. It would appear, however, that *vituperate* and *revile* have the same index numbers corresponding to their pointer values.

7.1.1 Trial run

In order to run the program at the command prompt do the following:

- `cd` into the top level subdirectory where the program is installed
- `wn [word] [search option]`

If one only includes the word to search for with no search option(s) WordNet will output the search options available for that word. One can then rerun the program as before with the desired search option(s) included. The following is the initial output for the word *beginning* (with no search option(s)):

```
Information available for noun beginning
  -antsn   Antonyms
  -hypen   Hypernyms
```

[4] These are the open-class words of English. The closed-class words of English (e.g., prepositions, pronouns, and determiners) are not included in WordNet

```
   -hypon, -treen Hyponyms & Hyponym Tree
   -synsn    Synonyms (ordered by estimated frequency)
   -derin    Derived Forms
   -famln    Familiarity & Polysemy Count
   -coorn    Coordinate Terms (sisters)
   -hholn    Hierarchical Holonyms
   -grepn    List of Compound Words
   -over   Overview of Senses

No information available for verb beginning

Information available for verb begin
   -antsv    Antonyms
   -hypev    Hypernyms
   -hypov, -treev Hyponyms & Hyponym Tree
   -synsv    Synonyms (ordered by estimated frequency)
   -causv    Cause to
   -deriv    Derived Forms
   -famlv    Familiarity & Polysemy Count
   -framv    Verb Frames
   -coorv    Coordinate Terms (sisters)
   -simsv    Synonyms (grouped by similarity of meaning)
   -grepv    List of Compound Words
   -over   Overview of Senses

Information available for adj beginning
   -antsa    Antonyms
   -synsa    Synonyms (ordered by estimated frequency)
   -famla    Familiarity & Polysemy Count
   -grepa    List of Compound Words
   -over   Overview of Senses

No information available for adv beginning
```

A SampleData2.txt

```
bought
beginning
went
going
teams
wives
flies
now
are
ones
negotiation
types
happiness
```

B SampleData3.txt

The following are the contents of `SampleData3.txt` used in the demonstration runs (the text has been tokenized and formatted with one sentence per line):

```
02/03/07 - Source says that Khalid Sattar , a young , unemployed Sunni who has spoken
angrily about the presence of U.S. troops in Iraq , has been visiting a house on
Dhubat Street in Adhamiya near the al-Assaf mosque several times during the past
week .

Source does not know who lives there .

02/12/07 - There have been a large number of " hits " for the web site of Yarmuk
Hospital in the Yarmuk district of Baghdad .

02/24/07 - Source says that Khalid Sattar is applying for a job as an ambulance
driver at Yarmuk Hospital in Baghdad .

Source said that Sattar had never shown any prior interest in health care or emergency
services .

2/27/07 - The principal of Ali Primary School on Bayaa Street in Bayaa reports a
suspicious visit by parents of a prospective student .

The parents were not interested in the school curriculum but instead were asking
specific questions about the building .

03/02/07 - Detainee Mohammed Janabi , an al-Qaeda in Iraq member , reports that
suspected terrorist cell leader Ziyad al-Obeidi talked last fall about using liquid
bombs for IED attacks against coalition forces .

03/14/07 - A soldier who speaks Arabic overheard a conversation in a restaurant
in Baghdad concerning a " new " type of weapon .

Not clear what this was or who may have it .

03/15/07 - Source reports boasted to him about being in charge of " something very
big .  "

03/17/07 - Surveillance of house #23 on Dhubat Street found that Sufian Mashhad
entered at 0700 .  Two unknown males , approximately 18 to 20 years old , visited
the house at 0932 .

Another male , identified as Khalid Sattar , arrived at 1035 .

One of the unidentified men left at 1600 , the other at 1640 .

Sattar left at 2130 .

03/20/07 - E-mail message from an unknown user of an Internet cafe in Baghdad to
an unknown recipient in an Internet cafe in Karachi requested information on " that
stuff " .
```

03/22/07 - Two unidentified males were seen walking around the Ali Primary School in Bayaa taking pictures and writing in a notepad .

They ran away when approached by security guards .

03/24/07 - A map found in a search of the home of Ziyad al-Obeidi had red circles drawn around the Ali Primary School in Bayaa and Yarmuk Hospital in Yarmuk .

C Penn Treebank tagset

The following are the glosses of the tags (in alphabetical order) from the Penn Treebank tagset used in several of the demonstration runs above:

1. CC Coordinating conjunction
2. CD Cardinal number
3. DT Determiner
4. EX Existential *there*
5. FW Foreign word
6. IN Preposition or subordinating conjunction
7. JJ Adjective
8. JJR Adjective, comparative
9. JJS Adjective, superlative
10. LS List item marker
11. MD Modal
12. NN Noun, singular or mass
13. NNS Noun, plural
14. NNP Proper noun, singular
15. NNPS Proper noun, plural
16. PDT Predeterminer
17. POS Possessive ending
18. PRP Personal pronoun
19. PRP$ Possessive pronoun
20. RB Adverb
21. RBR Adverb, comparative
22. RBS Adverb, superlative
23. RP Particle
24. SYM Symbol
25. TO *to*
26. UH Interjection
27. VB Verb, base form
28. VBD Verb, past tense
29. VBG Verb, gerund or present participle
30. VBN Verb, past participle

31. VBP Verb, non-3rd person singular present
32. VBZ Verb, 3rd person singular present
33. WDT Wh-determiner
34. WP Wh-pronoun
35. WP$ Possessive wh-pronoun
36. WRB Wh-adverb

D UCREL CLAWS7 Tagset

1. APPGE possessive pronoun, pre-nominal (e.g. my, your, our)
2. AT article (e.g. the, no)
3. AT1 singular article (e.g. a, an, every)
4. BCL before-clause marker (e.g. in order (that),in order (to))
5. CC coordinating conjunction (e.g. and, or)
6. CCB adversative coordinating conjunction (but)
7. CS subordinating conjunction (e.g. if, because, unless, so, for)
8. CSA as (as conjunction)
9. CSN than (as conjunction)
10. CST that (as conjunction)
11. CSW whether (as conjunction)
12. DA after-determiner or post-determiner capable of pronominal function (e.g. such, former, same)
13. DA1 singular after-determiner (e.g. little, much)
14. DA2 plural after-determiner (e.g. few, several, many)
15. DAR comparative after-determiner (e.g. more, less, fewer)
16. DAT superlative after-determiner (e.g. most, least, fewest)
17. DB before determiner or pre-determiner capable of pronominal function (all, half)
18. DB2 plural before-determiner (both)
19. DD determiner (capable of pronominal function) (e.g any, some)
20. DD1 singular determiner (e.g. this, that, another)
21. DD2 plural determiner (these,those)
22. DDQ wh-determiner (which, what)
23. DDQGE wh-determiner, genitive (whose)
24. DDQV wh-ever determiner, (whichever, whatever)
25. EX existential there
26. FO formula
27. FU unclassified word
28. FW foreign word
29. GE germanic genitive marker - (’ or’s)
30. IF for (as preposition)
31. II general preposition
32. IO of (as preposition)

33. IW with, without (as prepositions)
34. JJ general adjective
35. JJR general comparative adjective (e.g. older, better, stronger)
36. JJT general superlative adjective (e.g. oldest, best, strongest)
37. JK catenative adjective (able in be able to, willing in be willing to)
38. MC cardinal number,neutral for number (two, three..)
39. MC1 singular cardinal number (one)
40. MC2 plural cardinal number (e.g. sixes, sevens)
41. MCGE genitive cardinal number, neutral for number (two’s, 100’s)
42. MCMC hyphenated number (40-50, 1770-1827)
43. MD ordinal number (e.g. first, second, next, last)
44. MF fraction,neutral for number (e.g. quarters, two-thirds)
45. ND1 singular noun of direction (e.g. north, southeast)
46. NN common noun, neutral for number (e.g. sheep, cod, headquarters)
47. NN1 singular common noun (e.g. book, girl)
48. NN2 plural common noun (e.g. books, girls)
49. NNA following noun of title (e.g. M.A.)
50. NNB preceding noun of title (e.g. Mr., Prof.)
51. NNL1 singular locative noun (e.g. Island, Street)
52. NNL2 plural locative noun (e.g. Islands, Streets)
53. NNO numeral noun, neutral for number (e.g. dozen, hundred)
54. NNO2 numeral noun, plural (e.g. hundreds, thousands)
55. NNT1 temporal noun, singular (e.g. day, week, year)
56. NNT2 temporal noun, plural (e.g. days, weeks, years)
57. NNU unit of measurement, neutral for number (e.g. in, cc)
58. NNU1 singular unit of measurement (e.g. inch, centimetre)
59. NNU2 plural unit of measurement (e.g. ins., feet)
60. NP proper noun, neutral for number (e.g. IBM, Andes)
61. NP1 singular proper noun (e.g. London, Jane, Frederick)
62. NP2 plural proper noun (e.g. Browns, Reagans, Koreas)
63. NPD1 singular weekday noun (e.g. Sunday)
64. NPD2 plural weekday noun (e.g. Sundays)
65. NPM1 singular month noun (e.g. October)

66. NPM2 plural month noun (e.g. Octobers)
67. PN indefinite pronoun, neutral for number (none)
68. PN1 indefinite pronoun, singular (e.g. anyone, everything, nobody, one)
69. PNQO objective wh-pronoun (whom)
70. PNQS subjective wh-pronoun (who)
71. PNQV wh-ever pronoun (whoever)
72. PNX1 reflexive indefinite pronoun (oneself)
73. PPGE nominal possessive personal pronoun (e.g. mine, yours)
74. PPH1 3rd person sing. neuter personal pronoun (it)
75. PPHO1 3rd person sing. objective personal pronoun (him, her)
76. PPHO2 3rd person plural objective personal pronoun (them)
77. PPHS1 3rd person sing. subjective personal pronoun (he, she)
78. PPHS2 3rd person plural subjective personal pronoun (they)
79. PPIO1 1st person sing. objective personal pronoun (me)
80. PPIO2 1st person plural objective personal pronoun (us)
81. PPIS1 1st person sing. subjective personal pronoun (I)
82. PPIS2 1st person plural subjective personal pronoun (we)
83. PPX1 singular reflexive personal pronoun (e.g. yourself, itself)
84. PPX2 plural reflexive personal pronoun (e.g. yourselves, themselves)
85. PPY 2nd person personal pronoun (you)
86. RA adverb, after nominal head (e.g. else, galore)
87. REX adverb introducing appositional constructions (namely, e.g.)
88. RG degree adverb (very, so, too)
89. RGQ wh- degree adverb (how)
90. RGQV wh-ever degree adverb (however)
91. RGR comparative degree adverb (more, less)
92. RGT superlative degree adverb (most, least)
93. RL locative adverb (e.g. alongside, forward)
94. RP prep. adverb, particle (e.g about, in)
95. RPK prep. adv., catenative (about in be about to)
96. RR general adverb
97. RRQ wh- general adverb (where, when, why, how)
98. RRQV wh-ever general adverb (wherever, whenever)

99. RRR comparative general adverb (e.g. better, longer)
100. RRT superlative general adverb (e.g. best, longest)
101. RT quasi-nominal adverb of time (e.g. now, tomorrow)
102. TO infinitive marker (to)
103. UH interjection (e.g. oh, yes, um)
104. VB0 be, base form (finite i.e. imperative, subjunctive)
105. VBDR were
106. VBDZ was
107. VBG being
108. VBI be, infinitive (To be or not... It will be ..)
109. VBM am
110. VBN been
111. VBR are
112. VBZ is
113. VD0 do, base form (finite)
114. VDD did
115. VDG doing
116. VDI do, infinitive (I may do... To do...)
117. VDN done
118. VDZ does
119. VH0 have, base form (finite)
120. VHD had (past tense)
121. VHG having
122. VHI have, infinitive
123. VHN had (past participle)
124. VHZ has
125. VM modal auxiliary (can, will, would, etc.)
126. VMK modal catenative (ought, used)
127. VV0 base form of lexical verb (e.g. give, work)
128. VVD past tense of lexical verb (e.g. gave, worked)
129. VVG -ing participle of lexical verb (e.g. giving, working)
130. VVGK -ing participle catenative (going in be going to)
131. VVI infinitive (e.g. to give... It will work...)

132. VVN past participle of lexical verb (e.g. given, worked)

133. VVNK past participle catenative (e.g. bound in be bound to)

134. VVZ -s form of lexical verb (e.g. gives, works)

135. XX not, n’t

136. ZZ1 singular letter of the alphabet (e.g. A,b)

137. ZZ2 plural letter of the alphabet (e.g. A’s, b’s)

References

[1] Alembic Workbench Project, `http://www.mitre.org/tech/alembic-workbench/`.

[2] Annotate, `http://www.coli.uni-saarland.de/projects/sfb378/negra-corpus/annotate.html`.

[3] Automatic Content Extraction Project, `http://projects.ldc.upenn.edu/ace/tools/2005Toolkit.html` (2005).

[4] BBN Technologies, `http://www.bbn.com/Products_and_Services/Unstructured_Data/Identifinder.html`.

[5] Brill, Eric (1994). “Some Advances in Transformation-Based Part of Speech Tagging”, *AAAI-94*.

[6] Brill, Eric, *Transformation-based Part of Speech Tagger*, `RULE_BASED_TAGGER_V1.14` (software download from Eric Brill’s website), `README.TRAINING` (1993).

[7] Frakes, William B. and Fox, Christopher J. (2003). “Strength and Similarity of Affix Removal Stemming Algorithms”, *ACM SIGIR Forum*, volume 37, issue 1, pages 26-30.

[8] FreeLing User Manual 1.5 (2006). `http://garraf.epsevg.upc.es/freeling/index.php?option=com_content&task=view&id=18&Itemid=47`.

[9] GATE—General Architecture for Text Engineering, `http://gate.ac.uk/`.

[10] IBM Global Name Recognition, `http://www-306.ibm.com/software/data/globalname/`.

[11] Inxight, *Thing Finder and Thing Finder Professional: Automatic Entity and Fact Extraction*, `http://www.inxight.com/products/sdks/tf/`.

[12] Joy, Darrel (1976). *Lex: A Morphological Analyzer*.

[13] Jurafsky, Daniel and Martin, James H. (2000). *Speech and Language Processing*. Prentice-Hall, Inc., Upper Saddle River, New Jersey.

[14] Liu, Hugo (2004). MontyLingua: An end-to-end natural language processor with common sense. Available at: `web.media.mit.edu/˜hugo/montylingua`.

[15] Lockheed Martin, `http://www.lockheedmartin.com`.

[16] Miller, George A. (1995). “WordNet: A Lexical Database for English”, *Communications of the ACM*, volume 38, issue 11, pages 39-41.

[17] Mitre, *Callisto*, `http://callisto.mitre.org/`.

[18] Natural Language Software Registry, `http://registry.dfki.de/`.

[19] NetOwl, *Discovery Software from SRA International*, `http://www.netowl.com/products/extractor.html`.

[20] Petasis, Georgios, *Ellogon*, `http://www.ellogon.org/index.php?option=com_content&task=view&id=24&Itemid=56`, (2005).

[21] Porter, M.F. (1980). “An Algorithm for Suffix Stripping”, *Program* 14, 130-137.

[22] Rennhacckamp,Martin, *IBM’s Intelligent Family: Decision making driven by IBM’s Business Intelligence Initiative*, August 1998, `http://www.dbmsmag.com/9808d17.html`.

[23] Santorini, Beatrice (1990). “Part of Speech Tagging Guidelines for the Penn Treebank Project (3rd Revision, 2nd Printing)”, `http://www.cis.upenn.edu/˜treebank/`.

[24] Shapiro, Stuart C. (2006). “Information Extraction for Soft Target Exploitation and Fusion Statement of Work”, unpublished.

[25] Shapiro, Stuart C. and The SNePS Implementation Group (2004). SNePS 2.6.1 User’s Manual, Department of Computer Science and Engineering, University at Buffalo, The State University of New York, Buffalo, NY. `http://www.cse.buffalo.edu/sneps/Manuals/manual261.pdf`.

[26] TALP Research Group, Universitat Politécnica de Catalunya, `http://garraf.epsevg.upc.es/freeling/index.php`.

[27] UCREL (University Centre for Computer Corpus Research on Language), Lancaster University, `http://www.comp.lancs.ac.uk/ucrel/claws/`