FRDCSA Hacks

Andrew Dougherty

FRDCSA Hacks
by Andrew Dougherty

Table of Contents

1. Introduction....1

Introduction....1

2. Random Hacks....2

Packaging Software....2
Automatic Classification of Software Using Folksonomy....2
QUAC (Question Answering Component)....2
(Nonsensical) Self-Directed Goal Setting....3
Dream Analysis....3
CLEAR....3
Spidering Sourceforge and Packaging its Software....3
Deidentifying Personal Information....4
Recording Emacs Sessions....4
Code Monkey....4
Manually Classifying UniLang Entries....4
Finding Words That Fit Syntactic and Semantic Constraints....4
Comprehensive Software Ontology....5
Comprehensive Software Ontology, More....5
Book Scanner....5
Bus Route Planning....5
Meal Planning....5
Food Ontology....5
FreeKBS....5
Generate a Movie or a Book....6
The Event Log and Recording User Activities....6
APIs for Many Systems....6
Course Calendar Integrated with AI Planning....7
Defining Words in Text For Learners....9
....9
An Exercise Manager....10
Find dependencies....10
System to Find Jobs....10
Automatic Processing of Email....10
Automatic Acronym Generation....10
Automatic Acronym Generation....10
System Recommender....10
Releasing and Packaging the FRDCSA....10
Goal Interest Ability System For Group Meetings....11
Formalization of Natural Language Into Logic Using Cyc NL....11
Auto Information Extraction....11
Subsystems Listing....11
A Functionality Classifier....12
Multiagent System with Automatic Message Routing....12
A Knowledge Based Chess Analysis/Playing Program....13
Language Translation and Semantic Markup Web Proxy....13
War On Words Web Proxy....13

Circular Dependencies 13
Emacs Matrix Mode 13
Apartment Finder 13

3. Irish Heritage Computer Systems 14

.... 14
Formalization of the Irish Annals 14
Irish TTS 14
Save Tara Myspace Mailings 14
More Comprehensive Irish/English Freedict 14

4. Incomplete Hacks 15

Awesome NLP Library 15
Inventory Management 15

5. Wishlist Hacks 16

Nuance PaperPort Equivalent 16
IRC Topic Detection and Tracking 16
Boycott Manager 16
A programming language based on Natural Language/Perl 16
Speech Recognition 16
Planning, Scheduling and Execution Systems 16
Terminology Management 16
Linux Alternatives/Equivalents/Replacements, Software Substitution Database 17
Intelligent Software Review 17

6. Non-FRDCSA Hacks 18

Accessing your bank acount with GnuCash 18
Conglomerating Events Calendars with Geek Event Aggregator 18

Chapter 1. Introduction

Introduction

First I am going to apologize because I doubt this talk will be anything like it could have been. I was very fatigued this week and lay in bed quite often. First I will show you some of the important systems of the FRDCSA, then show some interesting hacks I was able to find looking around.

Chapter 2. Random Hacks

Packaging Software

~400 rough quality Debian packages have been built semi-automatically using the packager system. There is a related system which is packaging all of CPAN for Debian. Eventually, we'd like to package Sourceforge and Freshmeat, along with many systems that are not in any metasite.

Automatic Classification of Software Using Folksonomy

Per John Quigley's suggestion, Folksonomy takes a tag system, that is a bunch of tags, and learns a relationship between project descriptions and the tags, so that you can tag arbitrary projects based on just their description. So far, it just uses the Debian debtags tag system, but it is trivial to implement for Sourceforge as well as Freshmeat tags. The dataset is the datamart from the Flossmole project, plus the output of an 'apt-cache search -f .'

```
./query-tool.pl tagsearch .
./query-tool.pl search "works-with::image:vector"
```

The program needs to be cleaned up and given to the Debtags people. The models take a long time to learn and use the AI::Categorizer::Learner::SVM perl module.

QUAC (Question Answering Component)

QUAC currently answers only factoid questions, i.e. questions that expect a "who, what, where, when" type answer, because the only QA backend is Aranea. When accessing google, it is really quite slow so for this presentation I've just recorded the output of a query. Aranea uses its own interface to Google instead of the standard CPAN ones. I should adapt it but can't because I don't have an old google api license key. If anyone can present me with one that would be nice!

```
./quac --mods Aranea -s Glimpse -q "Who died?"

andrewdo@al:/var/lib/myfrdcsa/codebases/internal/quac$ quac -q "Who was the King of Ireland?" -s Glimpse --mods Aranea
Initializing sources...
Initializing QUAC/Source/Glimpse.pm...
Starting ModManager...
Starting at entry 0,  attempting to grab 100
Searching sources...
Searching Glimpse...
Searching source: Glimpse
Got 72 hits.
QUACSources reports 72 results to this query
Sucessfully grabbed 5.
In total, 5 have been retrieved.
```

```
1 Q0 DOC 1 3 ARANEA ENGLAND
[3.0, ENGLAND]
```

(Nonsensical) Self-Directed Goal Setting

Wouldn't it be nice if the software itself was able to develop its own goals. Well the opportunity afforded itself to write a very simple but nonsensical system for doing this. A trigram model is built over any source text of existing goals and then sentences are randomly generated and checked with the link parser grammar.

```
Later to make the fweb service better, we can implement PIC.
Is it worth searching for some of the items.
Keep us up to order DSL.
Think holistically of all the relevant mathematical concepts in a program.
Be sure to add parts of systems, for use in representing them.
This entire model of programming is going to be one!
Try to learn the various parts of systems, for use in tracking time.
Review it at the same study component for both Evangelist and Study.
Integrate reverse stimulation (the effect of being worked on.
Add current buffer detection to the backup disk.
Then, we can pass a few days, but now they are autoclassified.
Keep up to date on perl regular expressions.
Tie up your hands when you have to restart everything.
Is it worth searching for some of the art stuff.
Generix should have a daily summary of what we need.
Come up with a simple perltk user interface for classification to be in.
Currently my strategy seems to be determined before mission critical usage.
```

Dream Analysis

In Manager, Sbagen can be used to help record dreams. There is software to predict when you are going to go to sleep, and wake up, etc.

CLEAR

Clear can read books, with pause, resume, forward to sentence, etc. It can also quiz you on the readings.

Spidering Sourceforge and Packaging its Software

This system scrapes over sourceforge and downloads /var/lib/myfrdcsa/codebases/minor/game-time

Deidentifying Personal Information

This program checks for passwords and other personal information in text files to allow you to release them.

Recording Emacs Sessions

Using ttyrec and the dribble file, record emacs sessions and post-process them.

Code Monkey

Converting a perl script into a module with all the bells and whistles /var/lib/myfrdcsa/codebases/internal/code-monkey/scripts/convert-script-to-module.pl HilariousOutput.txt Segmenting Camelcase (in manager) ./normal-form/not-spark/best-breakdown-camelcase.pl ?Generating mnemonics? generate-mnemonics.pl

Manually Classifying UniLang Entries

./Corpus/UI.pm Before the catastrophe in which I lost over 15000 todo list entries, there were a substantial number of entries. In order for the system to learn where these entries should be routed at entry time, it needs training data. So this program helped with the labelling. It has really been superseded by the corresponding Critic software.

Finding Words That Fit Syntactic and Semantic Constraints

```
bard -c basic --reg "^lo"
$VAR1 = {
'Rhyme' => undef,
'Regex' => '^lo',
'MaxSize' => undef,
'SeedWords' => [
'basic'
],
'Relation' => '==',
'Syllables' => '0',
```

```
'Subtone' => undef
};
locate, location, lowborn (vs. noble), lowly, low, low gear, lowermost, lodging, love child, localisation, localization, locating, lower-ranking, lot, low (vs. high), low-down, low-pitched, low-spirited, low-toned, loan, lovable (vs. hateful), loveable,   C-c C-c
```

Comprehensive Software Ontology

CSO, -a asset

Comprehensive Software Ontology, More

A massive list of Debian package repositories (from apt-get.org). /var/lib/myfrdcsa/codebases/internal/cso/scripts/sources.list

Book Scanner

The book scanner was a nifty hack that took two days of programming. It never possessed a camera that could be remote controlled using gphoto2. Tests on picture quality revealed that a scanner is preferable to a camera and so when this is rebuilt, it should use a scanner. There are a couple of people interested in this project. http://frdcsa.onshore.net/frdcsa/internal/digilib/bookscanner.mov

Bus Route Planning

At one point in time there were not the bus route planners that there are today. This motivated the creation of one.

Meal Planning

./pgourmet -l

Food Ontology

http://frdcsa.org/~andrewdo/WebWiki/FoodOntology.html

FreeKBS

KBS Store is a useful module for doing relational querying without tables in SQL. in other words you can assert, unassert and query existence and models for relations in SQL of arbitrary arity. A simple context mechanism is provided, although it should probably be expanded to include genlmt type capabilities. It uses MySQL as the storage backend, as such it has capabilities for generating subselects for the user, which may wind up become a full blown query language. It is written in order to allow us to have a quick and convenient persistancy to a knowledgebase. It does not currently do inferencing, although the point is to do inferencing later, if the technology proves sufficiently strong and efficient.

Generate a Movie or a Book

The Event Log and Recording User Activities

Event log records many things, and mainly among them are the files that you are visiting in Emacs.

```
23854 display::/var/lib/myfrdcsa/codebases/internal/evangelist/scripts/correct-terminology.pl::9465::0
23851 opened::/var/lib/myfrdcsa/codebases/internal/evangelist/scripts/correct-terminology.pl::9465::0
23852 closed::/var/lib/myfrdcsa/codebases/internal/evangelist/scripts/correct-terminology.pl::9465::0
23853 conceal::/var/lib/myfrdcsa/codebases/internal/evangelist/scripts/correct-terminology.pl::9465::0
23855 opened::/var/lib/myfrdcsa/codebases/internal/evangelist/scripts/add-capability.pl::9465::0
23860 display::/var/lib/myfrdcsa/codebases/internal/evangelist/scripts/add-capability.pl::9465::0
23856 closed::/var/lib/myfrdcsa/codebases/internal/evangelist/scripts/add-capability.pl::9465::0
23857 conceal::/var/lib/myfrdcsa/codebases/internal/evangelist/scripts/add-capability.pl::9465::0
23859 display::/var/lib/myfrdcsa/codebases/internal/evangelist/scripts/dvds/dvdauthor.xml::9465::0
23858 opened::/var/lib/myfrdcsa/codebases/internal/evangelist/scripts/dvds/dvdauthor.xml::9465::0
23862 conceal::/var/lib/myfrdcsa/codebases/internal/evangelist/scripts/dvds/dvdauthor.xml::9465::0
23861 closed::/var/lib/myfrdcsa/codebases/internal/evangelist/scripts/dvds/dvdauthor.xml::9465::0
23863 opened::/var/lib/myfrdcsa/codebases/internal/evangelist/scripts/fork.pl::9465::0
23864 display::/var/lib/myfrdcsa/codebases/internal/evangelist/scripts/fork.pl::9465::0
23865 closed::/var/lib/myfrdcsa/codebases/internal/evangelist/scripts/fork.pl::9465::0
23868 conceal::/var/lib/myfrdcsa/codebases/internal/evangelist/scripts/fork.pl::9465::0
```

APIs for Many Systems

There are many systems for which I've written wrappers.

```
  Apertium.pm    Docbook    ExtractAbbrev.pm  Glimpse.pm  LinkGrammar.pm  MinorThird.pm
OpenCyc.pm  Spark   TextMine.pm  WWW
  CELT.pm        Docbook.pm  Festival.pm      GnuPlot  LinkParser.pm  MontyLingua.pm  Rhyme.pm    Spark.pm
Wiki
  ConceptNet.pm  Enju.pm    Gate.pm      GnuPlot.pm  MinorThird  OpenCyc  RWhois.pm   Template.pm  Wiki.pm
```

Course Calendar Integrated with AI Planning

```
Day: M
0000 - 1320                     Seminar in IT Policy   WEH 4601
0000 - 1320                   COS Reading & Research   WEH 4601
0900 - 1020 Comparative Methods of Software Development   NSH 3002
0900 - 1020         Methods: Deciding What to Design   NSH 3002
1030 - 1150           Software Development Studio II   NSH 3002
1030 - 1120                       Mathematical Games    BH 231A
1030 - 1120                               Set Theory    BH 231A
1030 - 1120 Advanced Topics in Discrete Mathematics:    BH 231A
1030 - 1150  Communication for Software Engineers II   NSH 3002
1030 - 1220         Interface and Interaction Design   HBH 2224
1030 - 1150 Advanced Artificial Intelligence Concepts   WEH 5409
1030 - 1150 Advanced Topics in Programming Languages  WEH 4615A
1130 - 1220                           Levy Processes    BH 231A
1200 - 1320 Software Engineering for Information Technology   NSH 3002
1230 - 1320                                Algebra I     MM 103
1330 - 1450                      Machine Translation   WEH 4601
1330 - 1450       Semantics of Programming Languages   WEH 7500
1330 - 1450                         Machine Learning   WEH 7500
1500 - 1650           Software Development Studio II   NSH 1305
1500 - 1620            Privacy and Anonymity in Data   WEH 8220
1500 - 1620          computer-mediated communication   WEH 4601
1500 - 1620                  COS Seminar (Computation   WEH 8220
1530 - 1650                           Levy Processes    OSC 201
1530 - 1650          Advanced Stochastic Calculus II    OSC 201
1530 - 1650 Advanced Topics in Financial Mathematics:    OSC 201
1530 - 1650 Software Engineering for Information Technology    DH 2105
1530 - 1650 Language Technologies for Computer Assisted Language Learning    BH 237B
1730 - 1850       Architectures for Software Systems   NSH 1305
1730 - 1850       Architectures for Software Systems   NSH 1305
1730 - 1850             Software Engineering Seminar   NSH 1305

Day: T
0900 - 1020   Engineering Software Intensive Systems   NSH 1305
0900 - 1020              Software Process Definition   NSH 1305
0930 - 1120         Interface and Interaction Design   HBH 2224
1030 - 1150          Advanced Stochastic Calculus II    DH 4303
1030 - 1150           Analysis of Software Artifacts   NSH 1305
1030 - 1150                   Calculus of Variations    DH 4303
1030 - 1150           Analysis of Software Artifacts   NSH 1305
1030 - 1150       Semantics of Programming Languages  WEH 4615A
1030 - 1150 Advanced Topics in Programming Languages  WEH 4615A
1030 - 1150       Architectures for Software Systems   NSH 1305
1200 - 1320          computer-mediated communication  WEH 4615A
1200 - 1320                    Information Retrieval   NSH 3002
1330 - 1450                       MATHEMATICAL GAMES  WEH 4615A
```

```
1330 - 1450                       Mathematical Games  WEH 4615A
1330 - 1520           physical computing - wearables    DH C316
1330 - 1450                         Machine Learning   NSH 1305
1330 - 1450 Advanced Artificial Intelligence Concepts   NSH 1305
1330 - 1450                 Advanced Topics in Logic    PH A19A
1330 - 1450               Mathematical Logic Seminar  WEH 4615A
1330 - 1450                           Lambda Calculus    PH A19A
1330 - 1450                               Set Theory    PH A19A
1500 - 1620         Interface and Interaction Design  WEH 4615A
1500 - 1620           physical computing - wearables  WEH 4615A
1500 - 1620            Privacy and Anonymity in Data WEH 5419AB
1500 - 1620                     Seminar in IT Policy WEH 5419AB
1500 - 1620                       MATHEMATICAL GAMES   WEH 4601
1800 - 1950          Security for Software Engineers   NSH 3002
1800 - 1950    Seminar in Software Development Tools   NSH 3002
1800 - 1950    Seminar in Software Development Tools   NSH 3002

Day: W
0000 - 1320                     Seminar in IT Policy   WEH 4601
0000 - 1320                   COS Reading & Research   WEH 4601
0900 - 1020 Comparative Methods of Software Development   NSH 3002
0900 - 1020         Methods: Deciding What to Design   NSH 3002
1030 - 1150           Software Development Studio II   NSH 3002
1030 - 1120                       Mathematical Games    BH 231A
1030 - 1120                               Set Theory    BH 231A
1030 - 1120 Advanced Topics in Discrete Mathematics:    BH 231A
1030 - 1150  Communication for Software Engineers II   NSH 3002
1030 - 1220         Interface and Interaction Design   HBH 2224
1030 - 1150 Advanced Artificial Intelligence Concepts   WEH 5409
1030 - 1150 Advanced Topics in Programming Languages  WEH 4615A
1130 - 1220                           Levy Processes    BH 231A
1200 - 1320 Software Engineering for Information Technology   NSH 3002
1230 - 1320                                Algebra I     MM 103
1330 - 1450                      Machine Translation   WEH 4601
1330 - 1450       Semantics of Programming Languages   WEH 7500
1330 - 1450                         Machine Learning   WEH 7500
1500 - 1620          computer-mediated communication   WEH 4601
1500 - 1620                 COS Seminar (Computation   NSH 1507
1500 - 1620             Software Engineering Seminar   NSH 1507
1530 - 1650                           Levy Processes    OSC 201
1530 - 1650          Advanced Stochastic Calculus II    OSC 201
1530 - 1650 Advanced Topics in Financial Mathematics:    OSC 201
1530 - 1650 Software Engineering for Information Technology    DH 2105
1530 - 1650 Language Technologies for Computer Assisted Language Learning    BH 237B
1730 - 1850       Architectures for Software Systems   NSH 1305
1730 - 1850       Architectures for Software Systems   NSH 1305
1730 - 1850             Software Engineering Seminar   NSH 1305

Day: R
0900 - 1020   Engineering Software Intensive Systems   NSH 1305
0900 - 1020              Software Process Definition   NSH 1305
0930 - 1120         Interface and Interaction Design   HBH 2224
1030 - 1150          Advanced Stochastic Calculus II    DH 4303
```

```
1030 - 1150           Analysis of Software Artifacts   NSH 1305
1030 - 1150                   Calculus of Variations    DH 4303
1030 - 1150           Analysis of Software Artifacts   NSH 1305
1030 - 1150       Semantics of Programming Languages  WEH 4615A
1030 - 1150 Advanced Topics in Programming Languages  WEH 4615A
1030 - 1150       Architectures for Software Systems   NSH 1305
1200 - 1320               Mathematical Logic Seminar   HBH 1004
1200 - 1320             Advanced Topics in Analysis:   HBH 1004
1200 - 1320         computer-mediated communication  WEH 4615A
1200 - 1320                    Information Retrieval   NSH 3002
1200 - 1320                   Calculus of Variations   HBH 1004
1330 - 1450                       MATHEMATICAL GAMES  WEH 4615A
1330 - 1450                       Mathematical Games  WEH 4615A
1330 - 1520           physical computing - wearables    DH C316
1330 - 1450                         Machine Learning   NSH 1305
1330 - 1450 Advanced Artificial Intelligence Concepts   NSH 1305
1330 - 1450                Advanced Topics in Logic    PH A19A
1330 - 1450               Mathematical Logic Seminar  WEH 4615A
1330 - 1450                           Lambda Calculus    PH A19A
1330 - 1450                                Set Theory    PH A19A
1500 - 1620         Interface and Interaction Design  WEH 4615A
1500 - 1620           physical computing - wearables  WEH 4615A
1500 - 1750   Engineering Software Intensive Systems   WEH 5409
1500 - 1620            Privacy and Anonymity in Data WEH 5419AB
1500 - 1750 Risk Management for Software Intensive Projects   WEH 5409
1500 - 1620                     Seminar in IT Policy WEH 5419AB
1500 - 1620                       MATHEMATICAL GAMES   WEH 4601
1730 - 1950            Managing Software Development   NSH 1305
1730 - 1950           Analysis of Software Artifacts   NSH 1305
1800 - 1950              Software Process Definition  WEH 4615A

Day: F
0930 - 1120    Seminar in Software Development Tools   NSH 3002
0930 - 1120              Seminar in Software Process   NSH 3002
1000 - 1120  Modeling Dynamic Software Architectures   WEH 8220
1030 - 1120                       Mathematical Games    BH 231A
1030 - 1120                                Set Theory    BH 231A
1030 - 1120 Advanced Topics in Discrete Mathematics:    BH 231A
1030 - 1150 Advanced Artificial Intelligence Concepts   WEH 5409
1130 - 1220                           Levy Processes    BH 231A
1230 - 1320                                Algebra I     MM 103
1530 - 1650 Software Engineering for Information Technology    DH 2105
```

Defining Words in Text For Learners

/var/lib/myfrdcsa/codebases/internal/all/scripts/define-previously-unseen-uses-of-words.pl An example run of a system that defines previously unseens wordnet senses after WSD. The goal is to define words /var/lib/myfrdcsa/codebases/internal/all/data/example

A TTS Wrapper that classifies languages and reads using appropriate TTS system and language pair.

An Exercise Manager

/var/lib/myfrdcsa/codebases/internal/akahige/Akahige/Exercise.pm

Find dependencies

System to Find Jobs

Automatic Processing of Email

/var/lib/myfrdcsa/codebases/internal/antispam-console/ASConsole/CorporaManager.pm A system that supports monitoring of an IMAP account among other things. /var/lib/myfrdcsa/codebases/internal/antispam-console/ASConsole/Filter/Setanta.pm A filter that with approval executes common tasks based on incoming mail. ./asconsole --corpora IMAP --filters Setanta (then send a test email)

Automatic Acronym Generation

Automatic Acronym Generation

I am working on a poetry generation system, called Bard, which I would eventually like to enable generation of Irish poems according to traditional bardic poetry styles. It should also handle other languages. http://frdcsa.onshore.net/frdcsa/internal/bard Bard was instrumental in generating the following poem: http://frdcsa.org/~andrewdo/writings/ErrToTheThrown.html It occurred to me just now that Gramadoir could be used to do the checking, the Irish wordnet for synset, etc. So a very good start. I still would like to find nice libre bidirectional Irish / English dictionaries for learning, such as O'Donnell's Foclair Bhearla Gaeilge, if that's available.

System Recommender

http://frdcsa.org/~andrewdo/WebWiki/SystemRecommender.html

Releasing and Packaging the FRDCSA

Goal Interest Ability System For Group Meetings

Generates emails to people reminding them about the meeting and which projects they are working on.

Formalization of Natural Language Into Logic Using Cyc NL

```
Formalize, Calculate the amount of energy of driving car versus using computer.
Calculate#v#1 the amount#n#3 of energy#n#1 of driving#n#2 car#n#1 versus use#v#1 computer#n#1

Calculate#v#1
the
amount#n#3
of
energy#n#1 N10712950 #$EnergyQuantity
of
driving#n#2
car#n#1 N02853224 #$Automobile
versus
use#v#1 V01123102 #$UsingAnObject
computer#n#1 N02971359 #$Computer
```

Much work remains on this project but it is actually easy work. You need to put in an entity recognizer which recognizes noun phrases to their objects, etc. Then you need a semantic search which uses supposition to understand the text, as detailed in thinker/notes.

Auto Information Extraction

Two algorithms have been implemented, one for website info extraction and another for arbitrary extraction. AIE, MDR

Subsystems Listing

/var/lib/myfrdcsa/codebases/internal/architect/data/systems/old-data/systems A list of systems extracted from writings. (not all writings used.)

```
abandoned work on a particular system
ability for other systems
acronym quizzing system
```

```
actual entities in the terminology management system
actual system
adequate system
advanced system
again the system
agenda system
agenda with this system
agent management system
ai system
al with syncronized hds system
analogical reasoning on existing systems
annotate instances of the metamax emacs system
appropriate system
archaic system
argument composition system
assignments system
audience dialog system
audience stuff system
autocorrection in documents of system
automated packaging system
automated surveillance system
automatic extraction of the relative merits of software systems
backing up the system
backup my system
backup of this system
basic document management systems
basic nutrition analysis system
basic sanctus system
basic system
basis for some future system
belief system
better backup system
better system
big application for a mate evaluator system
bot system
building systems
calendar.pm for event system
capabilities management system
capabilities management systems
capabilities of the various systems
capabilities of various communication systems
```

A Functionality Classifier

/var/lib/myfrdcsa/codebases/internal/architect/scripts/classify-capability.pl The functionality classifer.

Multiagent System with Automatic Message Routing

A Knowledge Based Chess Analysis/Playing Program

Unfortunately traditional chess algorithms like Negamax fail to take advantage of proof theoretic results that show provably superior play can be had with theorem proving over more sophisticated logics. I wrote a prolog program to encode positional chess knowledge into the search, a small step in the right direction of a knowledge based Chess (and Go) playing program.
/var/lib/myfrdcsa/codebases/internal/ender/data/chess/chess.pro

Language Translation and Semantic Markup Web Proxy

War On Words Web Proxy

This system takes normal newstext or any text for that matter and rewrites it to accomodate differing viewpoints.

Circular Dependencies

/var/lib/myfrdcsa/codebases/internal/boss/circular-dependencies.xml

Emacs Matrix Mode

This is a working system for displaying incoming UniLang messages. I thought it might stir up some interest in the project. (problemspace)

Apartment Finder

/var/lib/myfrdcsa/codebases/internal/apartment-finder/results.html

Chapter 3. Irish Heritage Computer Systems

In general please forgive the special interest in Irish. As I am interested in Irish culture and history, since the psychosis I use that interest to propel projects of that can be applied to many other areas as well.

Formalization of the Irish Annals

This hack seeks to codify all the ancient Irish annals into a knowledge based system suitable for asking all kinds of queries about the histories of particular people, groups, places, etc, in Ireland. Obviously it could be applied elsewhere. A related effort is to construct what I call the "Tara Knowledge Archives" which seeks to index texts relating to this ancient capital of Ireland and augment them with all kinds of semantic markup, like found in the AlleyOop tool. Very little has been done save to run the Free Logic Form parser against them to create logic forms. Additionally, they can be queried using the QUAC question answering system. The annals themselves are mostly from the CELT project. They should be stored properly in DigiLib using the Greenstone digital library or something similar.
/var/lib/myfrdcsa/codebases/minor/irish-annals/results.txt

Irish TTS

all -s data/irish-sample -l GA

Save Tara Myspace Mailings

Nifty little program to send messages to people on myspace who are friends of certain groups.
/home/andrewdo/save-tara

More Comprehensive Irish/English Freedict

Chapter 4. Incomplete Hacks

Awesome NLP Library

Combining as many different NLP libraries together PerlLib::NLP; # the most complete known NLP library # given an object what can we do with it or transform it to # aspects of functions my $c = [Text => { "" }, QuestionAnswering => { "", }, Extraction => { "Named Entities", "N-Grams", "Rhymes", "", }, Phrase => { "Dictionary Lookup", "Encyclopedia Lookup", "Wordnet Lookup", "Rhyming", "Connotations", "Associations", }, Question => { },]; my $capabilities = { "Question Answering" => { "TextMine" => [qw(QuestionP ClassifyQuestionType AnswerQuestion)], }, "NamedEntityExtraction" => {}, "Topic Detection and Tracking" => {}, "LanguageModelling" => {}, "AffectiveClassification" => { "Conceptnet" => [], } "English Predicate Parser" => { }, "Formalization" => {}, }

Inventory Management

Show Eric's 3d inventory module. /var/lib/myfrdcsa/codebases/internal/shops/shops

Chapter 5. Wishlist Hacks

Nuance PaperPort Equivalent

IRC Topic Detection and Tracking

Boycott Manager

We need to develop software that provides users with the ability to determine the pro's and con's of purchasing from a particular vendor or company, based on knowledge about the company's political dealings. The largest problem with this is contamination and hacking of it. But basically, software should maintain knowledge about specific issues, and make deductions based on what you tell it your position is, helping you choose a company to purchase from. Functionality belongs in Broker. P.S. You may also be interested in a project of mine to create a system for diseminating intelligence to consumers about which companies to boycott. The idea is that various sources can put out boycott demands (signed with their key), along with their reason for the boycott, and then the application can track which issues they agree with, in order to effectively boycott certain companies. While it may be a nice tool, it is somewhat subject to attack and disinformation, but I think it can improve the current situation nonetheless.

A programming language based on Natural Language/Perl

Speech Recognition

We need to get usable large vocabulary speaker independent recognition packaged.

Planning, Scheduling and Execution Systems

There are several systems here designed to improve the planning scheduling and execution abilities of the system, such as PSE, Score, etc.

Terminology Management

Linux Alternatives/Equivalents/Replacements, Software Substitution Database

(http://www.linuxalt.com/ http://www.osalt.com/)

Intelligent Software Review

Write a system to model what you know about software. And then present you novel information.

Chapter 6. Non-FRDCSA Hacks

Accessing your bank acount with GnuCash

Conglomerating Events Calendars with Geek Event Aggregator