Das Handbuch zu FreeHAL, ein Forschungsprojekt zur Entwicklung einer künstlichen Intelligenz.
Entwickelt und programmiert von Tobias Schulz.

Die Geschichte von FreeHAL

Seine Anfänge nahm FreeHAL als Jeliza im Juni 2006 als Tobias Schulz damit begann die ersten Codezeilen (damals noch in Java) zu schreiben.
2007 wurde die Programmiersprache auf Python und C/C++ umgestellt, doch mit Python konnten nicht alle Features von FreeHAL umgesetzt werden.
Deshalb entschied sich Tobias Schulz 2008 dazu, Perl und C++ zu verwenden. Die Datenmengen wurden größer und mit normalen Textdateien nicht mehr zu bewältigen. Aus diesem Grunde verwendet FreeHAL nun SQLite um die Datenmengen verarbeiten zu können.
Oft gesellten sich freiwillige Helfer zum Projekt, jedoch immer nur von kurzer Dauer.
Tobias Schulz glaubte jedoch an sein Projekt und arbeitete unbeirrt an seiner Idee weiter. Ihm ist es auch zu verdanken, das Freehal seit seiner Geburt eine rasante Entwicklung erlebt hat, deren Ende nicht abzusehen ist.

Seit Februar 2008 hat Tobias Schulz dauerhafte Unterstützung bekommen, die das Projekt mit Ideen, Testen und der Weiterentwicklung des Wissens von FreeHAL unterstützt.

Sollte jemand beim Lesen dieser Dokumentation mehr gefallen an FreeHAL finden und an der Entwicklung mithelfen wollen, würde sich Tobias Schulz freuen, sie im Team begrüßen zu dürfen.
Die entsprechenden Kontaktdaten sind auf der FreeHALseite http://freehal.org zu finden.

Es sollte beim Arbeiten mit FreeHAL nie vergessen werden, das es sich bei dieser Software um keine fertige, fehlerfrei Anwendung handelt, sondern um ein Forschungsprojekt . Aus diesem Grund wird es immer wieder zu kleinen Fehlern kommen.

Inhaltsverzeichnis

Die Arbeitsweise von FreeHAL

Das Computerprogramm FreeHAL ist das am weitesten entwickelte Dialogprogramm, das lernfähig und als freie Software (open source) verfügbar ist.

Es benutzt semantische Netze und arbeitet mit Mustererkennung, Stemmern, Wortartdatenbanken und Hidden Markov Modellen, um in Gesprächen ein möglichst menschliches Verhalten zu imitieren.
Im Gegensatz zu den meisten freien und kommerziellen Chatbots lernt FreeHAL jedoch selbstständig hinzu. Durch das Kommunizieren (per Tastatur) erweitert das Programm seine Wissensdatenbank.
Es unterstützt die Sprachen Deutsch und Englisch, wobei bisher nur für deutsch eine umfangreiche Datenbank - als semantisches Netz - vorliegt.

Der Antwortprozess

Der Benutzer sieht bei FreeHAL zumeist nur die GUI, die grafische Benutzeroberfläche. Im Hintergrund startet FreeHAL aber zwei Prozesse, einer davon ist die GUI, und im anderen findet die Verarbeitung der Benutzereingaben statt. Alle Eingaben und alle Antworten von FreeHAL werden über eine interne TCP/IP-Verbindung verschickt.

Gibt der Benutzer eine Aussage oder Frage ein, wird dieser Text also an den Verarbeitungsprozess mit dem Namen "runner.exe" gesendet. Dort finden nun verschiedene Verarbeitungsstufen statt:

Einfache Mustererkennung

Zuerst wird versucht, bestimmte umgangssprachliche Formulierungen durch äquivalente standardsprachliche Ausdrücke zu ersetzen, um FreeHAL die Verarbeitung zu erleichtern. Auch werden hier die in der Datenbank eingetragenen Tippfehler verarbeitet.

Der Wortart-Tagger

Danach wird jedem Wort in der Eingabe Wortart und Genus (grammatisches Geschlecht) zugeordnet. Dies geschieht einerseits mit Hilfe von bereits definierten Wörtern in den Wortart-Dateien und andererseits - für die noch nicht definierten Wörter - mit dem Wortart-Tagger. Dieser Programmteil ordnet mit statistischen Berechnungen den noch unbekannten Wörtern eine Wortart zu, die er von bereits bekanntem Material abzuleiten versucht. Im Zweifelsfall wird der User um eine Wortart-Eingabe gefragt.

Der Parser

Der Parser ist dafür zuständig, den in natürlicher Sprache geschriebenen Text mit den Ergebnissen des Taggers in seine Bestandteile - Subjekt, Prädikat, Objekt, adverbiale Bestimmungen, ... - aufzutrennen.

Nach dem Parsen werden, soweit möglich, den Personal- und Possessivpronomen ihre entsprechenden Nomen zugeordnet. Als letzter Schritt während dem Parsen versucht FreeHAL, aufgrund von Erfahrungswerten und vorgegebenen Eigenschaften abzuleiten, welche Art von Antwort genau gemeint ist. Das kann eine Bestätigung, eine Sachantwort, eine Aufzählung, eine Begrüßung, eine Beleidigung oder ähnliches sein.

Die Datenbank

Mit Hilfe der Ergebnisse des Parsers wird nun in der Datenbank nach Fakten gesucht, aus denen sich Antworten ableiten lassen. Die Datenbank liegt in Form eines semantischen Netzes vor. Dies ist im Abschnitt "Semantisches Netz" genauer beschrieben.

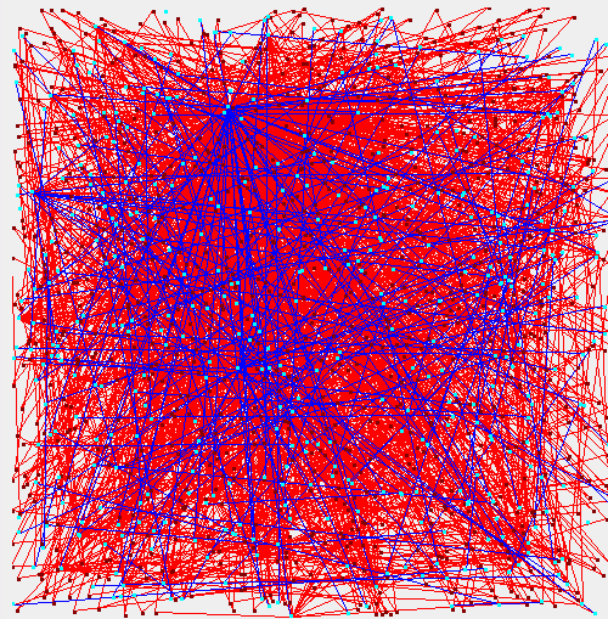
Bewertungssystem

Wenn ein- oder mehrere Fakten, die sich als Antwort eignen, gefunden wurden, wird mit dem Bewertungssystem eine Antwort ausgewählt. Wenn keine Antwort gefunden wurde, wird mit einer Standardantwort geantwortet.

Formulierung der Antwort

Die Formulierungsfunktion formuliert aus dem Fakt einen deutschen Satz. Hier kommt es beispielsweise darauf an, Groß- und Kleinschreibung zu beachten, Artikel zu ergänzen und Verben zu ordnen.

Semantisches Netz - einfache Darstellung eines ganzen Netzes



Bei diesem Screenshot sieht man ein relativ aktuelles semantisches Netz, wobei rote Verbindungen Verben und blaue Orts- und Zeitangaben entsprechen. Bei aktuellen Revisionen sind beide Linientypen zu der Farbe blau zusammengefasst.

Installation

Installation unter Windows

Unter Windows müssen sie das Zip-Archiv herunterladen, in ein beliebiges Verzeichnis entpacken und die FreeHAL-QT.exe ausführen.
Unterschiede in den Versionen:
- Stabile Version: Sie müssen nur in der GUI " Use Database engine... Disk " auswählen damit Freehal startet.
- SVN Version: Diese Versionen stehen mit Database oder ohne Database als Download zur Verfügung. Wenn sie die Version ohne Database auswählen erstellt FreeHAL nach Start der FreeHAL -QT.exe eine Datenbank. Dieser Vorgang wird einige Zeit in Anspruch nehmen.

Installation unter Linux

Zum Kompilieren benötigen sie folgende Softwarepakete:
- gcc und g++
- CMake
- Subversion Client
- libqt4-dev - Qt >= 4.4 + Development Pakete
- libasio-dev (Boost.Asio), mindestens 1.2
- boinc-dev (BOINC development files)
- libperl-dev (Perl 5.10 Development files).

Installieren unter Ubuntu/Debian:
sudo apt-get install build-essential g++ cmake subversion libqt4-dev libasio-dev boinc-dev libperl-dev
Herunterladen des Quellcodes:
svn co http://freehal.org/code/hal2009
Ins Quellcodeverzeichnis wechseln:
cd hal2009
Das Makefile mit CMake generieren:
./configure
FreeHAL kompilieren:
make hal2009-server
Zum kompilieren der GUI in das Verzeichnis der GUI wechseln:
cd gui-qt
Das Makefile mit qmake generieren:
qmake
Die GUI kompilieren:
make
Die ausführbare Datei für die GUI befindet sich nun im Ordner "gui-qt/bin/" und heißt "freehal". Nun sollte sie, wenn man beide Komponenten nicht getrennt starten will, zum FreeHAL-Kern kopiert werden, um von dort ausgestartet zu werden:
cp bin/freehal ..
Nun ins Quellcodeverzeichnis zurückwechseln:cd ..
FreeHAL starten:
./freehal

Programmupdate und Datenbankneuaufbau

Programmupdate

Bei einem Programmupdate ist darauf zu achten, das sie vorher den "lang_de" Ordner sichern. In diesem befinden sich alle Daten die sie beim Arbeiten mit FreeHAL erstellt haben.

Nach dem Sichern die gleichen Schritte wie bei der Installation ausführen und danach den "lang_de" Ordner wieder zurückkopieren.

Sie können auch alternativ nur die neuen Dateien aus dem heruntergeladenen Zipordner in das Hauptverzeichnis von FreeHAL entpacken und damit die alten Dateien überschreiben.

Datenbankneuaufbau

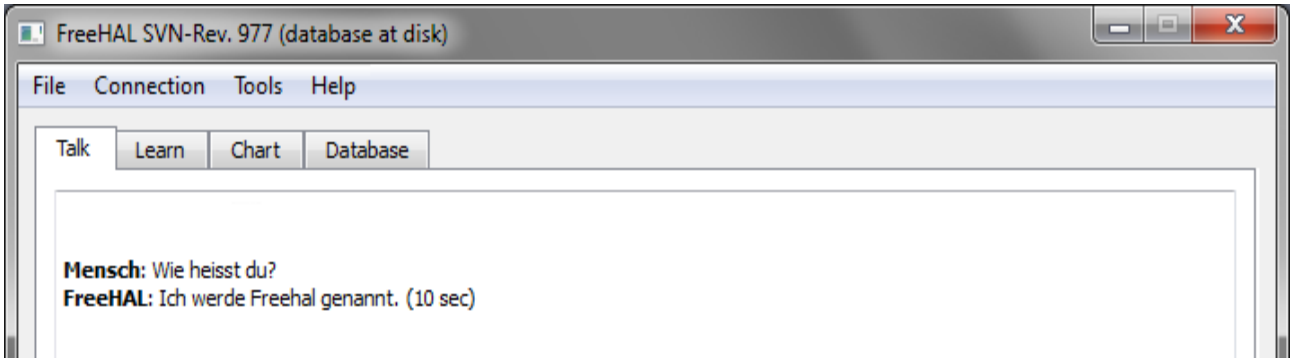
Von Zeit zu Zeit ist es sinnvoll und auch manchmal notwendig ,(wenn die Datenbankstruktur geändert wurde, oder sehr viel in den .pro Dateien editiert wurde) diese neu zu erstellen.

- Alle Dateien die sich im "saved" Ordner befinden bitte löschen.
- Die "database.db" im "lang_de" Ordner löschen.
- Die "FreeHAL-QT.exe" ausführen.

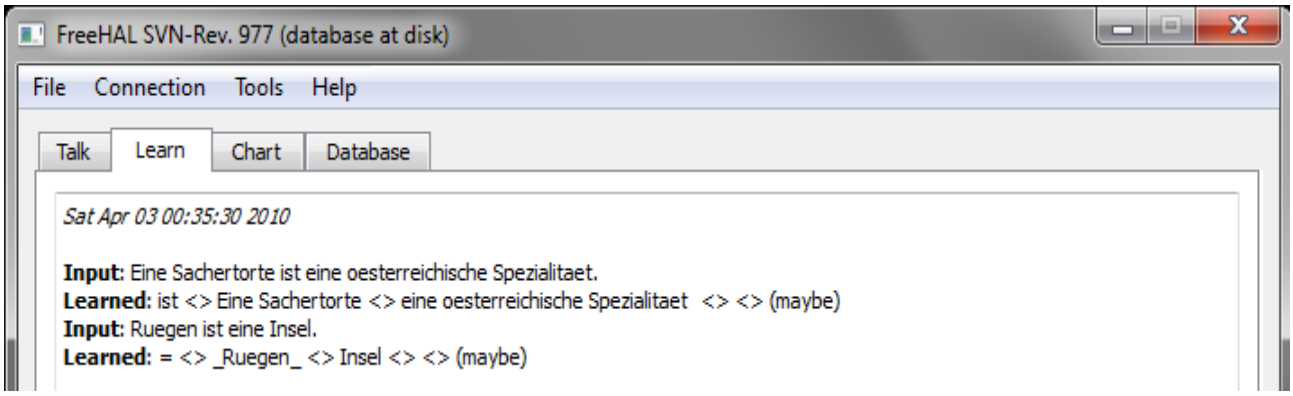
Mit diesen Schritten wird die Datenbank neu aufgebaut. Bitte denken sie daran, dass dies einige Zeit in Anspruch nimmt.

In künftigen Versionen von FreeHAL wird es mehrere Datenbanken geben um die Performance von FreeHAL durch die größer werdenden Datenmengen zu gewährleisten.

Die Benutzeroberfläche

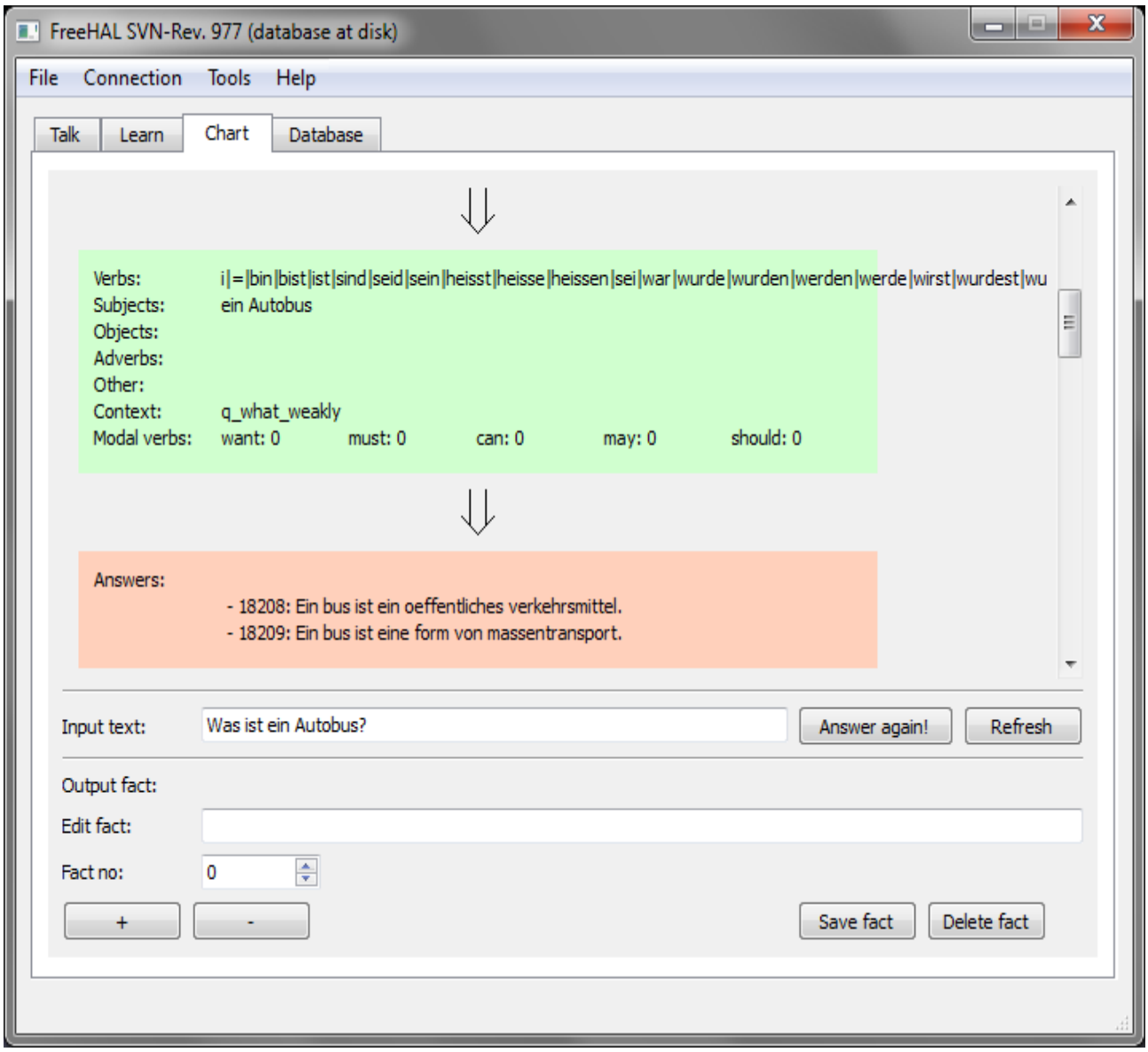


Im Talkregister finden die normalen Unterhaltungen mir FreeHAL statt. Hier ist es nicht möglich FreeHAL Wissen zu vermitteln. Die Zeitangabe in der Klammer gibt Auskunft darüber, wie lange die Verarbeitungszeit der Eingabe des Users beträgt.



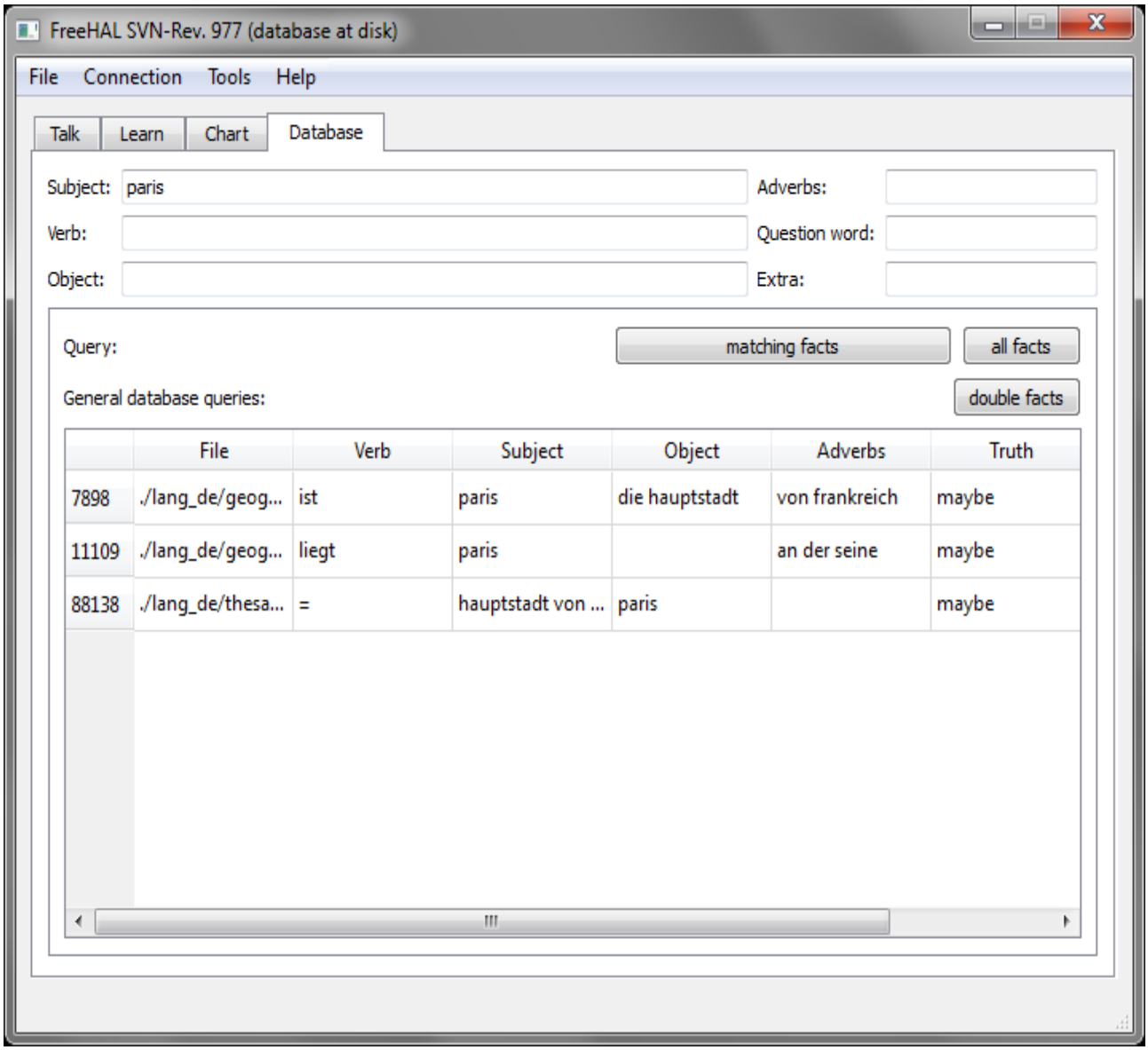
Im Learnregister wird, wie der Name schon sagt, FreeHAL Wissen vermittelt. Im Gegensatz zum Talkregister wird keine Antwort ausgegeben, sondern der Eintrag in die Datenbank angezeigt. Damit ist es möglich zu sehen, ob FreeHAL die Eingabe korrekt verarbeitet hat, oder ob ein Fehler vorliegt. Die Erklärung zum „maybe“ in der Klammer wird in einem späteren Kapitel dieser Dokumentation beschrieben.

Weitere Möglichkeiten FreeHAL schneller und effizienter Wissen zu vermitteln werden im Kapitel „Dateitypen „ beschrieben.



Im Chartregister wird angezeigt, nach welchen Begriffen gesucht wird, und welche möglichen Antworten gefunden werden.
Sollte FreeHAL für die Frage eine passende Antwort in der deutschen Wikipedia finden, bleibt das Antwortfeld leer.
Des weiteren besteht in diesem Register die Möglichkeit Einträge zu editieren oder zu löschen.

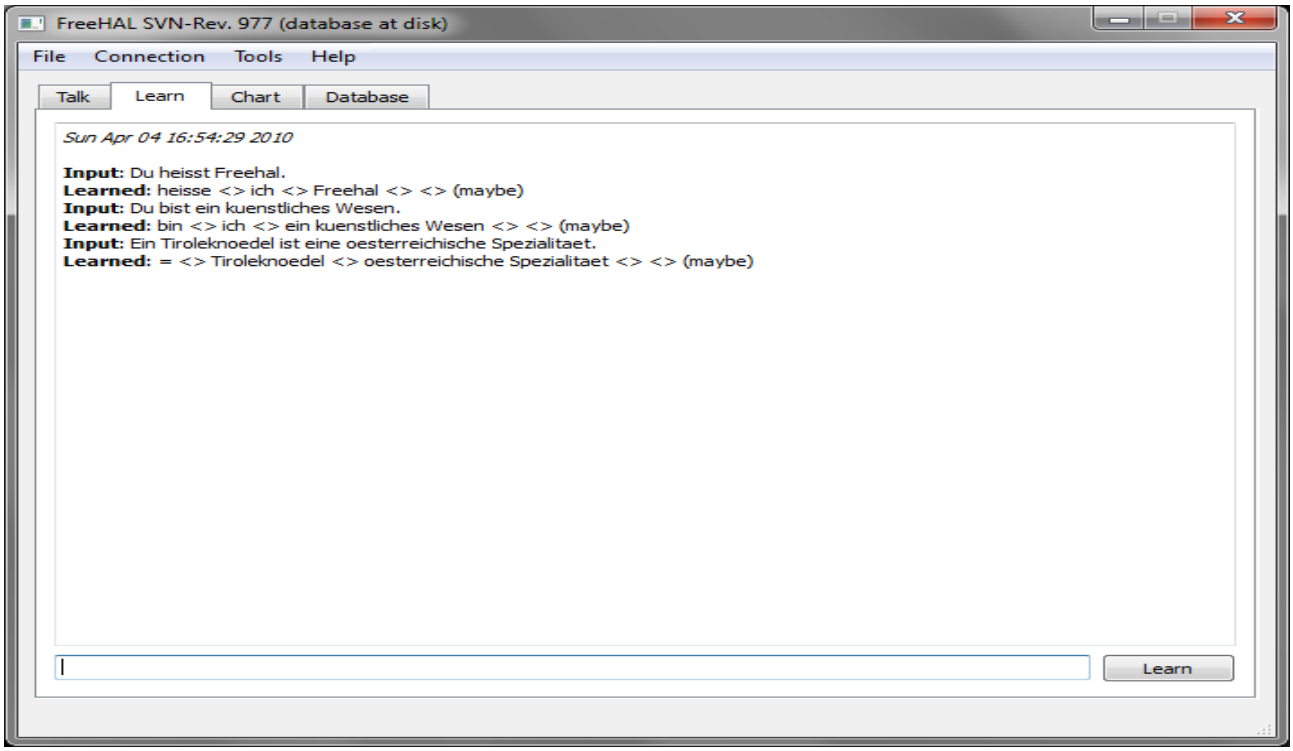
Mit der Plus und Minus Taste wird die Antwort, die FreeHAL ausgegeben hat bewertet, das heißt, es ist möglich Antworten zu favorisieren oder abzuwerten. Diese ist dann hilfreich, wenn Free-HAL mehrere mögliche Antworten findet, aber nur eine wirklich zu 100% zutrifft.



Im Database Register ist es möglich nach einzelnen, oder mehreren Begriffen zu suchen. Dies ist hilfreich bei falschen Antworten auf eine Eingabe, oder auch danach zu suchen , warum FreeHAL eine bestimmte Antwort ausgibt..
Mit der rechten Maustaste kann man auch Einträge löschen . Eine Editierfunktion wie im Chartregister steht in Kürze zur Verfügung.

Lernen mit FreeHAL

Es gibt mehrere Möglichkeiten FreeHAL Wissen zu vermitteln. Der einfachste Weg ist im Lernmodus Daten und Fakten ein zu geben.



Alle Daten die im Lernmodus eingegeben werden speichert FreeHAL in der Datei „facts.pro“. Das oben genannte Beispiel steht dann wie folgt in dieser Datei.

```
heisse <> ich <> FreeHAL <> <> (maybe)
bin <> ich <> ein kuenstliches Wesen <> <> (maybe)
 = <> Tirolerknoedel <> oesterreichische Spezialitaet <> <> (maybe)
```

Diese Methode ist jedoch für große Datenmengen nicht geeignet, da diese Art der Eingabe sehr zeitraubend ist. Eine Weitere Möglichkeit die schneller und effizienter geht ist das Schreiben einer „xxx.prot Datei. Mit einem Texteditor wird eine Datei mit Eingaben für FreeHAL erstellt und als xxxx.prot Datei im „lang_de“ Ordner unter einem passenden Namen z. B. „freehal.prot“ abgespeichert. Der Inhalt dieser Datei könnte zum Beispiel so aussehen:

```
Du heisst FreeHAL.
Du bist eine künstliche Intelligenz.
Du wohnst in einem Computer.
```

FreeHAL wandelt dann diese Datei in eine“ freehal.pro“ Datei um. Die „freehal.prot“ Datei wird danach nicht mehr benötigt und kann aus dem Ordner in einen anderen verschoben werden. „xxx.prot“ sollten nicht gelöscht werden, da jemand vielleicht später sie noch erweitern will. Im Download von FreeHAL sind jedoch nur „xxx.pro“ Dateien enthalten.

Eine weitere interessante Möglichkeit ist, eine sogenannte ps-xxx.prot zu schreiben. Ich nehme hier als Beispiel an, dass ich FreeHAL viele Küchengeräte beibringen will. Dafür erstelle ich wieder im „lang_de“ Ordner eine ps-Kuechengeraet.prot Datei.

```
Abkuehlgitter
Abtropfsieb
Abziehstein usw.
```

Wenn FreeHAL die Datei in eine „ps-Kuechengeraet.pro umgewandelt hat, schaut der Inhalt so aus:

```
= <> abkuehlgitter <> kuechengeraet <> <> (maybe)
= <> abtropfsieb <> kuechengeraet <> <> (maybe)
= <> abziehstein <> kuechengeraet <> <> (maybe)
```

ACHTUNG: Beim erstellen einer „xxx.prot“ Datei ist darauf zu achten, dass keine Umlaute und kein „ß“ verwendet werden, diese würden dann in der „xxx.pro“ Datei stehen.

FreeHAL besitzt einen kompletten Thesaurus, trotzdem kann es notwendig sein, FreeHAL mitzuteilen das ein Wort oder eine Sache ein und das selbe sind. Diese wird so gemacht:

```
„Ewig leben“ = „unsterblich sein“
```

Der Eintrag in der Datenbank sieht so aus:

```
= <> _ewig_leben_ <> _unsterblich_sein_ <> <> (true)
```

Ebenso ist es für FreeHAL interessant Beziehungen zwischen gewissen Dingen zu lernen. Hierzu ein einfaches Beispiel:

```
Hunger haben (reasonof) etwas essen
```

Der Eintrag in der Datenbank sieht so aus:

```
reasonof <> _hunger_haben_ <> _etwas_essen_ <> <> (maybe)
```

Daraus ergibt sich folgende Antwortmöglichkeit für FreeHAL:

```
Mensch: Ich habe Hunger.
FreeHAL: Du solltest etwas essen.
```

Werden diese Eingaben im Lernmodus getätigt, speichert FreeHAL diese direkt in die „Beziehungen.pro“ Datei ab.

FreeHAL will aus wissen ob etwas positiv oder negativ ist. Dafür gibt es (good) und (bad). Bei Eingabe über den Lernmodus werden diese Fakten in der „ps-good.pro“ und in der „ps-bad.pro“ direkt abgespeichert.

```
Freude (good)
Krieg (bad)
```

Dies ergibt folgende Einträge in der entsprechenden Datenbank:

```
= <> _freude_ <> _(good)_ <> <> (maybe)

= <> krieg <> _(bad)_ <> <> (maybe)
```

Für ein zukünftiges Feature gibt es noch (true) , (false) und (maybe). Standartmäßig setzt Free-HAL alle Eingaben in den „xxx.pro“ Dateien auf den Status (maybe). Sollte bei der Eingabe etwas anderes gewünscht werden so muss man dies FreeHAL wissen lassen.

```
„Kühlschrank“ = „Kochgelegenheit“ (false)
„Sonne“ = „Wärme“ (true)
„Flüssigkeit“ = „Alkohol“ (maybe)
```

Die Dateitypen von FreeHAL

Momentan gibt es im "lang_de" Ordner 3 verschiedene Dateitypen:

- ...prot
- ...pro
- ps-xx.prot oder ps-xx.pro

In den .pro-Dateien werden die zerlegten Sätze (Adjektive, Nomen, Verben, usw.) gespeichert und von dort aus in die Datenbank eingelesen. Die Standard-.pro-Datei heißt "facts.pro", in der alle Eingaben über die GUI gespeichert werden.

Die .prot Datein sind reine Textdateien und werden von FreeHAL, wenn sie sich im Ordner lang_de befinden, in .pro-Dateien umgewandelt. Um große Datenmengen zu verarbeiten, ist es daher besser und schneller, die Eingaben in einer Textdatei mit der Endung .prot zu speichern. FreeHAL schaut beim Starten im "lang_de" Ordner nach, ob sich eine .prot Datei darin befindet, und wandelt diese dann automatisch in eine .pro Datei, mit gleichem Namen wie die .prot Datei, um. Bleibt die .prot-Datei im lang_de-Ordner, wird diese beim nächsten Start nicht noch einmal umgewandelt, da bereits eine .pro Datei mit gleichem Namen existiert.

Das t bei "prot" steht für Template (engl. Vorlage, Muster).

Mit den ps-xx.prot und ps-xx.pro werden FreeHAL reine Fakten vermittelt.
In einer ps-xx.prot ist es nicht mehr notwendig ein Fakt komplett zu beschreiben.

Es muss nicht:

```
"Eine Amsel ist ein Vogel"
"Ein Star ist ein Vogel"
```

geschrieben werden, sondern es genügt der Datei den Namen "ps-Vogel.prot" zu geben und sie müssen danach nur mehr eingeben:

```
 "Amsel"
"Star"
```

FreeHAL erzeugt in einer "ps-Vogel.pro" folgende Einträge:

```
= <> amsel <> vogel <> <> 50
= <> star <> vogel <> <> 50
```

Sämtliche "ps_xxx.xx Dateien“ werden im Unterordner "ps_dateien" gespeichert.

Wortarten, Dateien und Definitionen

FreeHAL verwendet zur Verwaltung der Wortarten vier Dateien:

- ***word_types.brain***

Diese Datei ist eine Sicherungsdatei und wird von FreeHAL nicht verwendet. Sie dient dem Entwicklerteam alle bisherigen Wortarten zu verwalten und ev. Fehler aus früheren Bugs auszubessern. Im Auslieferungszustand ist die "word_types.brain" mit allen Wortarten die FreeHAL bekannt sind befüllt.

- ***word_types.memory***

Die ist die Arbeitsdatei von FreeHAL, in diese werden alle neuen Wörter, die FreeHAL aus der word_types.brain bezieht oder die vom User abgefragt werden , gespeichert.

- ***word_types.self-taggered***

Alle Wörter die FreeHAL selbst taggert werden in dieser Datei gespeichert.

- ***protocol.memory***

Alle Wortarten die FreeHAL in die word_types.memory oder in die word_types.self-taggerd schreibt sind in dieser Datei zu finden. Diese Datei dient zur Kontrolle und dem Entwicklerteam zur Überprüfung, ob der Tagger einwandfrei arbeitet.

Folgende Wortarten werden von FreeHAL definiert:

Nomen: n
Artikel: weiblich = f , männlich = m , sächlich = s
Verben: vi oder vt
Adverbien: adv
Adjektive: adj
Fragewort: fw
Präposition: prep
Pronomen: pron
Allgemeiner Ausruf: inter

Das Bewertungssystem

Da viele Sätze je nach Frage von FreeHAL's Logik anders interpretiert werden, als vom User gewünscht, besteht die Möglichkeit, gewisse Antworten zu favorisieren.

Ein schönes Beispiel ist die Frage "**Was bist du?**". Ein User erwartet die Antwort "Ich bin eine KI.". Doch FreeHAL hat viele Sätze in der Datenbank, die mit "Ich bin..." beginnen, so zum Beispiel "Ich bin nicht muede." oder "Ich bin nicht hungrig."

Für FreeHAL würden alle Sätze auf die Frage "Was bist du?" logisch erscheinen. Jeder User hat nun die Möglichkeit mit dem Bewertungssystem FreeHAL zur richtigen Antwort zu lenken:

```
Mensch: Was bist du?
FreeHAL: Ich bin muede.
Mensch: -
FreeHAL: Bewertung erfolgreich
```

Der User hat mit dem "-" FreeHAL nun mitgeteilt, dass diese Antwort nicht die beste ist.

```
Mensch: Was bist du?
Freehal:Ich bin eine KI.
Mensch: +
FreeHAL: Bewertung erfolgreich
```

Mit dieser Antwort ist er User einverstanden.

Bitte nicht vergessen, das es auch im Chartregister möglich ist, FreeHAL's Antworten zu bewerten. Der große Vorteil im Chartregister besteht darin, das FreeHAL dort alle möglichen Antworten ausgibt und die Möglichkeit besteht zu sehen , ob es überhaupt eine gute Antwort für die Frage gibt.

Schlussfolgerungen

Dieses Feature lässt sich am besten anhand eines Beispiels erklären.

Die Voraussetzung ist folgender Datenbankeintrag:

A ist b gross, wenn A einen Durchmesser von B hat

Datenbankeintrag:

ist <> a <> b <> gross <> hat <> a <> einen durchmesser <> von b <> wenn <> <> 50

Dadurch ist folgendes möglich:

Mensch: Wie gross ist Io?

FreeHAL: Io hat einen Durchmesser von 3630 Kilometer. ,)

Rekursiv ist die Schlussfolgerung aus möglich:

Mensch: Wenn A B gross ist, ist A B lang

FreeHAL: Tja ! Hmm, Wie denkst Du ueber einen tasteninstrument?

Mensch: Wie lang ist Io?

FreeHAL: Io hat einen Durchmesser von 3630 Kilometer.

Die Erläuterung dazu:

Die Buchstaben A bis H dienen als Platzhalter (Variablen).

Erweiterte Schlussfolgerungen

Es ist möglich, Aktionen entweder als logische Abfolge oder als Gleichzeitigkeit anzugeben. Dazu werden einfach Nomen-Verb- bzw. Adjektiv-Verb-Paare in Anführungszeichen gesetzt und durch "->" oder das Gleichheitszeichen getrennt. FreeHAL kann diese Beziehungen in verschiedenen Situationen anwenden.

Ursache und Folge:

Mensch: „Durst haben“ -> „etwas trinken“
FreeHAL: (..)
Mensch: Ich habe Durst.
FreeHAL: Wieso trinkst du nichts.

Gleichzeitigkeit:

Mensch: „Durst haben“ -> „etwas trinken“
FreeHAL: (…)
Mensch: Ich habe Durst.
FreeHAL: Wie trinkst du nichts.

Gleiche Bedeutung:

Mensch: Wie gross ist Io?
FreeHAL: Der Jupitermond Io hat einen Durchmesser von 3630 Kilometern.
Mensch: „ gross ein“ = „lang sein“
FreeHAL: Hmmm.
Mensch: Wie lang ist Io=
FreeHAL: Der Jupitermond Io hat einen Durchmesser von 3630 Kilometern.

Komplexe Verbindungen wie die zwischen der Größe und dem Durchmesser ("A ist B groß, wenn A einen Durchmesser hat von B") müssen aber weiterhin durch Variablen ausgedrückt werden. Hier liegt es an dem "Durchmesser *von* *B*". Dazu ist eine Nomen/Verb bzw. Adjektiv/Verb-Beziehung zu allgemein.

FreeHAL kann dem Benutzer auch Vorschriften machen:

Mensch: Ich habe Durst.
FreeHAL: Du solltest etwas trinken.
Mensch: Ich habe Durst.
FreeHAL: Warum trinkst du nichts?

Allerdings kann auch das passieren:

Mensch: Ich habe Angst.
FreeHAL: Du solltest dich fürchten.

Smalltalkfähigkeit

Diese Fähigkeit lässt sich am besten mit Beispielen veranschaulichen.

Mensch: A ist langweilig !=> lies doch ein Buch

Datenbankeintrag:

ist <> a <> nothing <> langweilig <> f=> <> lies <> doch ein buch <> <> 50

Dadurch wird folgendes möglich:

Mensch: Mir ist langweilig.
FreeHAL: >Lies doch ein Buch.

Hier noch ein Beispiel:

Mensch: Dir ist auch langweilig !=> Das ist ganz normal

Datenbankeintrag:

Mensch: Ist dir auch langweilig?
FreeHAL: Das ist ganz normal.

Erläuterungen :

Das Symbol "!=>", das wie eine Konjunktion gehandhabt wird, steht vor einer vom User geforderten Antwort, die ein Aussagesatz sein soll. "?=>" steht vor einem Fragesatz, ".=>" gibt es nicht.

Die Buchstaben A bis H dienen Als Platzhalter (Variablen).

Angabe und Ursache einer Aktion

FreeHAL versteht nur eine Untermenge der deutschen Sprache. Selten erwartet es sogar statt eines normalen Satzes eine spezielle Syntax, da es damit besser klarkommt als mit einem komplexen Satz.

So ist das auch, wenn man die Folge einer bestimmten Handlung angeben will. FreeHAL versteht das oftmals nur mithilfe der folgenden Schreibweise:

```
"Durst haben" -> "etwas trinken"

"muede sein" -> "schlafen gehen"
```

Das Symbol "->" wandelt FreeHAL in das fiktive Prädikat "reasonof" um. Diese Eingabe erkennt FreeHAL's semantisches Netz also als:

```
_muede_sein_ reasonof _schlafen_gehen_
```

Es ist auch möglich, statt der ersten Variante gleich die zweite anzugeben.
Bitte beachten: Als letztes Wort innerhalb der Anführungszeichen muss immer ein Verb stehen!
Wenn FreeHAL in dieser Form dieses Wissen erhalten hat, ist es sofort in der Lage, es zu benutzen:

```
Mensch: Ich bin müde.
FreeHAL: Du solltest schlafen gehen.

Mensch: Ich bin müde.
FreeHAL: Warum gehst du nicht schlafen?
```

Zwei gleichzeitig geschehende Aktionen angeben

FreeHAL versteht nur eine Untermenge der deutschen Sprache. Selten erwartet es sogar statt eines normalen Satzes eine spezielle Syntax, da es damit besser klarkommt als mit einem komplexen Satz.

So ist das auch, wenn man FreeHAL sagen will, dass zwei Aktionen entweder das selbe bedeuten, oder immer gleichzeitig geschehen. Dazu verwendet man folgende Schreibweise:

```
Mensch: „Angst haben“  = „sich fürchten“
```

Bitte beachten: Als letztes Wort innerhalb der Anfuehrungszeichen muss immer ein Verb stehen!

Wenn FreeHAL in dieser Form dieses Wissen erhalten hat, ist es sofort in der Lage, es zu benutzen

```
Mensch: Fürchtest du dich?
FreeHAL: Nein, ich habe keine Angst vor dir.
Mensch: Ich fürchte mich.
FreeHAL: Warum hast du denn Angst?
```

Synonyme und Antonyme

FreeHAL versteht eine zusätzliche Syntax um Synonyme und allgemein zusammenhängende Wörter zu lernen.

Mit dem Kommando "learn:" kann man FreeHAL schnell und einfach zusammenhängende Wörter erklären, zum Beispiel:

Eingabe: learn: gemacht @ getan @ getrieben

Eingabe: learn: gemacht, getan, getrieben

Folgende Syntax ermöglichst es, Unterbegriffe und Eigenschaften anzugeben:

Eingabe: properties: Haus: Gebäude @ Wohnort @ Arbeitsplatz @ aus Beton gebaut @ aus Ziegeln gebaut @ aus Holz gebaut

Eingabe: properties: Haus: Gebäude, Wohnort, Arbeitsplatz, aus Beton gebaut, aus Ziegeln gebaut, aus Holz gebaut

Daraus lernt FreeHAL, dass ein Haus ein Gebäude ist, Wohnort oder Arbeitsplatz sein kann und aus Beton, Ziegeln oder Holz gebaut sein kann. Natürlich kann man diese Informationen auch einzeln eingeben, also: "Ein Haus ist ein Gebäude. Ein Haus kann aus Beton gebaut sein. Ein Haus kann aus Ziegeln gebaut sein. ...." Die neue Syntax ist aber bei vielen Informationen praktischer und zweckmäßiger.

Die Systemvariablen

FreeHal besitzt eine kleine Sammlung von Systemvariablen die hier kurz erklärt werden.

$$month$$ - Monat (Zahl)

$$mday$$ - Tag des Monats (Zahl)

$$year$$ - Jahr (4-stellig)

$$yday$$ - Tag im Jahr

$$time$$ - Zeit: Stunde:Minute:Sekunde

$$wday$$ - Wochentag

$$randomname$$ - Zufallsname, aus der Datenbank entnommen

Die Systemkommandos

Zur Zeit stehen folgende Systemkommandos zur Verfügung, die über die GUI eingegeben werden können.:

- /GEN LIST
FreeHAL generiert für jede .pro Datei eine Textdatei so wie FreeHAL die einzelnen Sätze ausgeben würde

- /GEN LIST dateiname.pro
Mit diesem Kommando wird nur aus dieser bestimmten Datei eine Textdatei erstellt.

- /DEL FACT
Die letzte Eingabe in die Datenbank wird aus dieser und aus der facts.pro gelöscht.

- ENGLISCH
Die Sprache von FreeHAL wird auf Englisch umgestellt.

- DEUTSCH
Die Sprache wird auf Deutsch umgestellt.

Entscheidungshilfen

true, false und maybe;

Diese 3 Wörter wird FreeHAL für die Entscheidungsfindung heranziehen. Standartmässig werden alle Fakten auf den Status **maybe** (vielleicht) gesetzt.

```
Eingabe: Ein killer toetet Menschen.
Fakt: toetet <> ein killer <> menschen <> <> (maybe)
```

Will man nur FreeHAL mitteilen ob ein Fakt definitiv falsch oder richtig ist kann man dies mit **true** oder **false** definieren.

```
Eingabe: Der Himmel ist blau (true)
Fakt:ist <> der himmel <> blau <> <> (true)
oder
Eingabe: Ein Wal ist ein Vogel. (false)
Fakt: = <> ein wal <> vogel <> <> (false)
```

Good und bad:

Mit **good** und **bad** wird FreeHAL mitgeteilt ob ein Wort etwas positives oder etwas negatives bedeutet.

```
Eingabe: Diebstahl (bad)
Fakt: = <> _diebstahl <> _(bad)_ <> <> (maybe)
oder
Eingabe: Freundschaft (good)
Fakt:= <> _freundschaft_ <> _(good)_ <> <> (maybe)
```

Im Gegensatz zu anderen Eingaben werden diese nicht in der facts.pro sonder in der **ps-good.pro und ps-bad.pro** direkt abgespeichert.

Opposite:

Mit **opposite** besteht die Möglichkeit FreeHAL Gegenteile beizubringen.

```
Eingabe: gut (opposite) böse.
```

Diese Fakten werden direkt in die **opposite.pro** abgespeichert.

Systemanforderungen

Nach umfangreichen Tests empfiehlt das Team mindestens einen Pentium 4 mit 2 Gigabyte Ram, um akzeptable Antwortzeiten zu bekommen.
Der Vergleich zwischen einem Pentium 4 und einem Quadcoreprozessor mit 4 Gigabyte Ram zeigt Differenzen in der Antwortzeit von 6 bis 10 Sekunden.
Je schneller und besser die Hardware ist umso schneller wird auch FreeHal antworten.

Softwaremäßig wurde und wird FreeHal auf seine Geschwindigkeit permanent überwacht.