Freetext Matching Algorithm Version 15

Anoop Shah, Clinical Epidemiology Group

May 17, 2013

Contents

1 Introduction

The Freetext Matching Algorithm extracts structured information in the form of Read terms, dates and numerical values from unstructured free text.

The program was developed initially in 2003-2005 by Anoop Shah working for the General Practice Research Database Division of the Medicines and Healthcare products Regulatory Agency. A Microsoft Access 2000 database was used for program development and can be downloaded from the original publication `http://www.biomedcentral.com/1472-6947/12/88/`. The history of development of the program is described in `http://www.biomedcentral.com/content/supplementary/1472-6947-12-88-s1.pdf`. This document describes only the latest version of the program.

Since the original publication, the program has since been updated as follows:

- Use human-readable text file lookups
- Compiled Visual Basic 6.0 instead of an Access database
- Output format similar to Clinical Practice Research Datalink (CPRD) GOLD format of electronic general practice records for research
- Code and lookups stored on public GitHub repositories, so that they are readily available, viewable and changes can be tracked

The program is designed to handle free text from general practice electronic health records held by the CPRD. It may be of use for other types of clinical free text but has not been tested on them.

If you use the Freetext Matching Algorithm in a research study, please cite: Shah AD, Martinez C, Hemingway H. The freetext matching algorithm: a computer program to extract diagnoses and causes of death from unstructured text in electronic health records. BMC Med Inform Decis Mak 2012;12:88 doi: 10.1186/1472-6947-12-88 `http://www.biomedcentral.com/1472-6947/12/88/`

The program is licensed under the GNU General Public License Version 3 (`http://www.gnu.org/licenses/gpl-3.0-standalone.html`).

2 Quick guide to running the program

System requirements: Microsoft Windows or Linux with wine-1.5.26 (it will probably also work with CrossOver on Macintosh), at least 500MB RAM.

Download the file 'fma15gui.exe' from the 'binaries' folder of this repository and lookup tables from the lookups repository (`https://github.com/anoopshah/freetext-matching-algorithm-lookups`). Save all these files in the same folder.

Run fma15gui.exe; it should display a dialog box as shown in Figure 1:

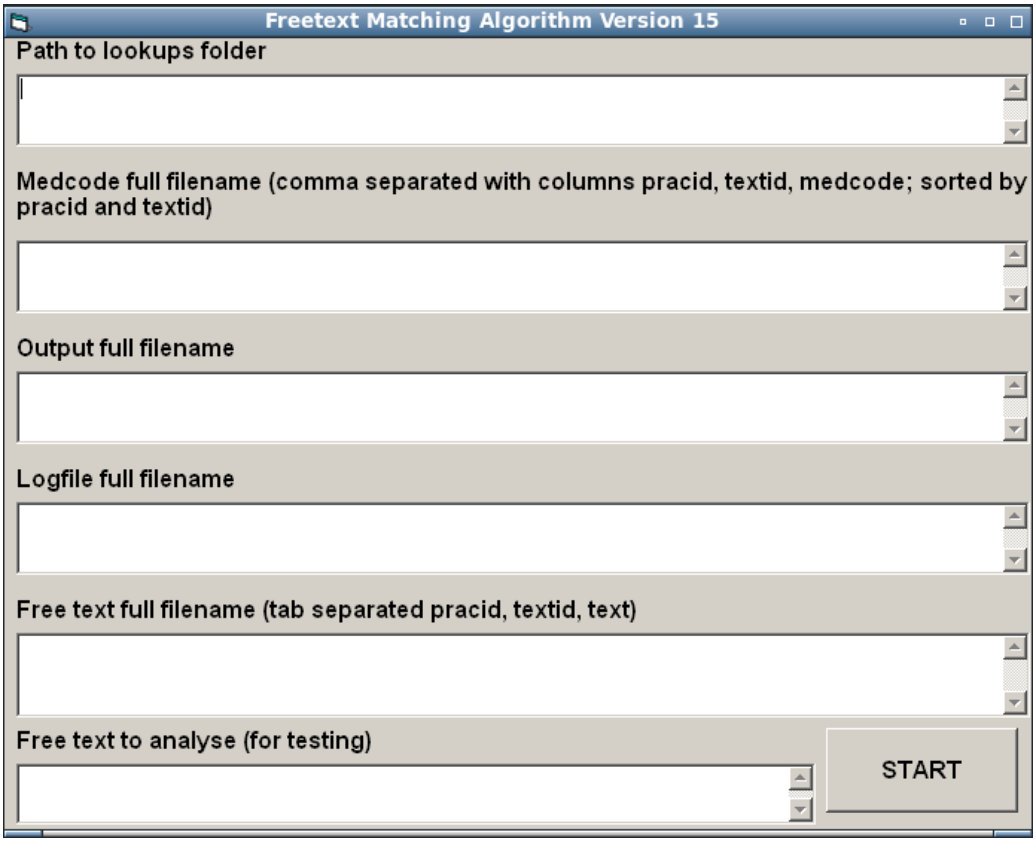

Figure 1: Freetext Matching Algorithm graphical user interface

To run the program, fill in the fields and click **START**. Fields which state 'full filename' require the complete filepath starting with the drive letter.

Lookups folder – folder containing lookup tables. Leave blank if they are in the same folder as the FMA program itself.

Medcode full filename – comma-separated file with the medcode associated with each pracid / textid combination, in one of the following formats:

1. A header row with column names 'pracid', 'textid', 'medcode' in any order
2. No header row, but pracid in the first column, textid in the and medcode in the third column (it is permitted for each textid / pracid combination to have multiple medcodes, and the medcodes do not have to be sorted)

The file must be sorted by pracid and textid and can have up to 200,000 rows. It must have Windows-style line endings (i.e. carriage-return, newline).

Output full filename – filename for the output file, which will be comma separated with column headers: pracid, textid, origmedcode, medcode, enttype, data1, data2, data3, data4. Apart from the header row, all data in this file are numerical.

Logfile full filename – filename for the output log file, which is a brief log file stating the time and date that the program was run, the name of the input file, the number of medcodes and texts used. When analysing a single text in test mode, the intermediate and final results of analysis of a single text are written to the log file, but when analysing texts from a file, no free text or actual results are written to the log file.

Free text full filename – tab-separated file containing pracid, textid, text (in that order, with no header row).

If a single free text is filled in the bottom text box, the medcode file, output file and free text file entries are ignored and only the single text is analysed.

If modifying the lookup tables, ensure that they remain in the same format (see `https://github.com/anoopshah/freetext-matching-algorithm-lookups/blob/master/README.md` for details).

3 Program output

The program processes unstructured free text and outputs it in a structured format (Figure 2).

The output format is a comma separated values file with the following columns, similar to the CPRD GOLD data format, so that it can be analysed alongside existing coded data in CPRD.

medcode new medcode extracted from free text. This can be interpreted in a similar way to medcodes in the original Clinical Practice Research Datalink GOLD data format; a medcode in this column is for a past or present event for this patient.

enttype virtual entity type for information extracted from text

data1 additional information (e.g. laboratory values, family history)

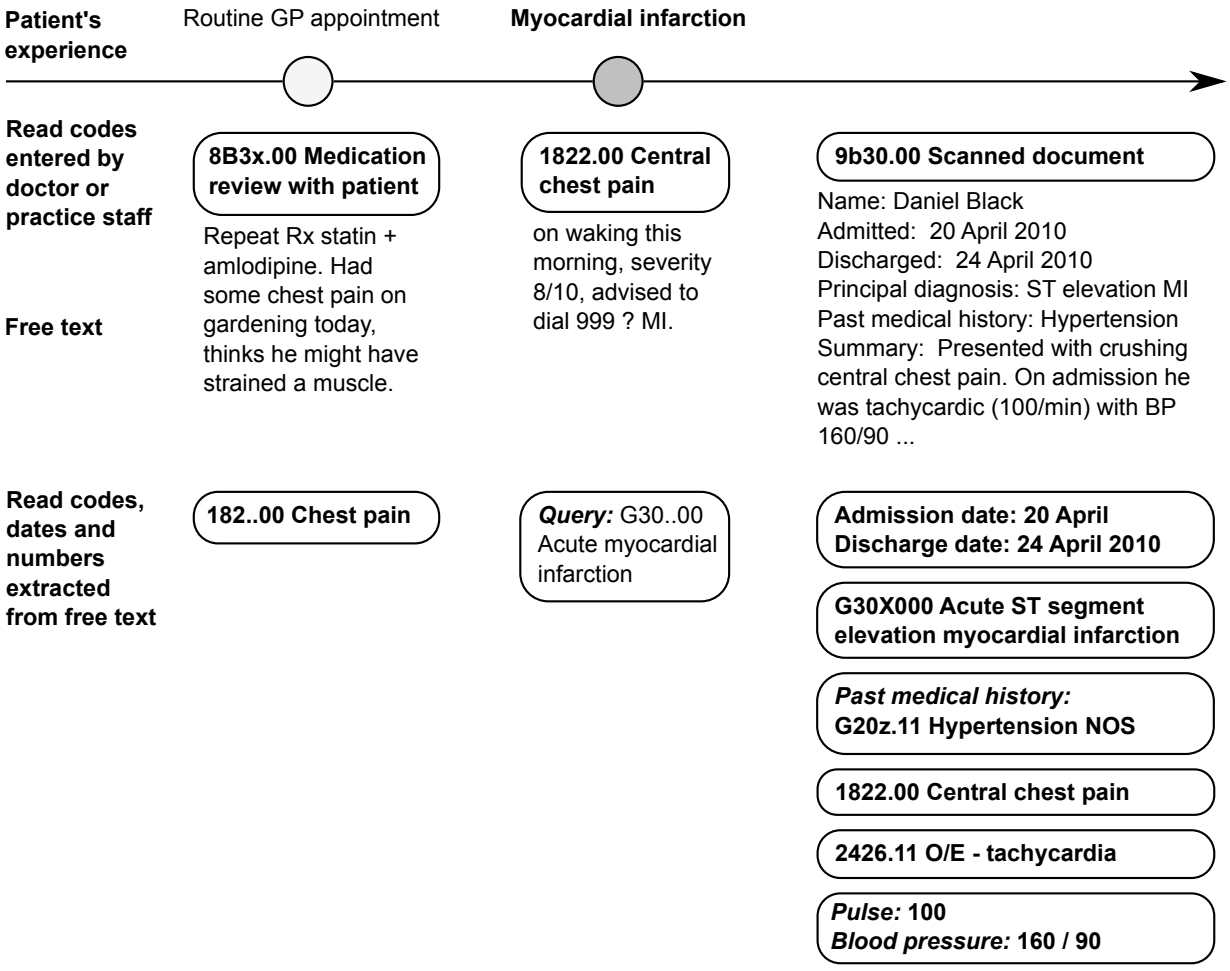

Figure 2: Hypothetical general practice record, showing additional information available in free text and how it can be encoded by the Freetext Matching Algorithm

data2, data3, data4 more additional information

All columns contain numeric data only. Categorical data values are represented by a code from the CPRD lookup tables, and dates are integers with digits representing the year in the format YYYYMMDD. The file **`fma_entity_definitions.txt`** is a tab-separated text file which specifies the meaning of each data element, and is used to interpret the output.

3.1 New entity types for information extracted from free text

enttype	description	category	num fields	data1	data1 lookup	data2	data2 lookup	data3	data3 lookup	data4	comment	medcode	readterm
1311	PF current	Asthma	2	Operator	OPR	Value					units and normal range are not extracted. Operator is always =	11772	Peak flow rate
1203	Thyroid stimulating hormone	Biochemistry (Hormone)	2	Operator	OPR	Value					OPR = 3 (=)	13598	TSH level
1157	B12 levels	Biochemistry (Other)	2	Operator	OPR	Value					OPR = 3 (=)	7926	Serum Vitamin B12
1169	Serum ferritin	Biochemistry (Other)	2	Operator	OPR	Value					OPR = 3 (=)	8491	Serum ferritin
1170	Folate level	Biochemistry (Other)	2	Operator	OPR	Value					OPR = 3 (=)	13748	Serum folate
1198	Thyroxine	Biochemistry (Other)	2	Operator	OPR	Value					OPR = 3 (=)	941	Serum T4 level
1159	Calcium	Biochemistry (Routine)	2	Operator	OPR	Value					OPR = 3 (=)	77	Serum calcium
1163	Serum cholesterol	Biochemistry (Routine)	2	Operator	OPR	Value					OPR = 3 (=)	12	Serum cholesterol
1165	Serum creatinine	Biochemistry (Routine)	2	Operator	OPR	Value					OPR = 3 (=)	5	Serum creatinine
1175	High density lipoprotein	Biochemistry (Routine)	2	Operator	OPR	Value					OPR = 3 (=)	44	Serum HDL cholesterol level
1177	Low density lipoprotein	Biochemistry (Routine)	2	Operator	OPR	Value					OPR = 3 (=)	65	Serum LDL cholesterol level
1197	Triiodothyronine	Biochemistry (Routine)	2	Operator	OPR	Value					OPR = 3 (=)	13791	Serum T3 level
1202	Triglycerides	Biochemistry (Routine)	2	Operator	OPR	Value					OPR = 3 (=)	37	Serum triglycerides
1204	Urea - blood	Biochemistry (Routine)	2	Operator	OPR	Value					OPR = 3 (=)	18587	Urea in sample
1275	HbA1c - diabetic control	Biochemistry (Routine)	2	Operator	OPR	Value					OPR = 3 (=)	14051	Haemoglobin A1c level
1148	Death administration	Death Administration	1	Date of Death	YYYYMMDD						times are not extracted	43009	Date of death
1149	Cause of death	Death Administration	1	Category of death	COD						COD is coded Ia = 1, 1b = 2, 1c = 3, II = 4.	(extracted from free text)	
2001	Cause of death (no category)	Death Administration	0								cause of death, category not specified	(extracted from free text)	
2011	Quantitative test result	Diagnostic Tests	2	Operator	OPR	Value					OPR = 3 (=)	(original associated with text)	
2012	Qualitative test result	Diagnostic Tests	1	Qualifier	TQU						for generic test results, keep the original medcode of the test and use the TQU lookup for qualifier:15 – nil, 9 – normal, 12 – abnormal, 22 – negative, 21 – positive	(original associated with text)	

(continued...)

(continued from previous page: New entity types for information extracted from free text)

enttype	description	category	num fields	data1	data1 lookup	data2	data2 lookup	data3	data3 lookup	data4	comment	medcode	readterm
1001	Blood pressure	Examination Findings	2	Diastolic		Systolic					combine into one entry	1	O/E - blood pressure reading
1131	Pulse (CVS/BP)	Examination Findings	1	Pulse rate								6154	O/E - pulse rate
1152	Albumin	Haematology	2	Operator	OPR	Value					OPR = 3 (=)	31969	Albumin in sample
1173	Haemoglobin	Haematology	2	Operator	OPR	Value					OPR = 3 (=)	4	Haemoglobin estimation
1182	Mean corpuscular volume	Haematology	2	Operator	OPR	Value					OPR = 3 (=)	10	Mean corpuscular volume (MCV)
1189	Platelets	Haematology	2	Operator	OPR	Value					OPR = 3 (=)	7	Platelet count
1207	Total White Blood cell count	Haematology	2	Operator	OPR	Value					OPR = 3 (=)	13818	White cell count
1273	Erythrocyte sedimentation rate	Haematology	2	Operator	OPR	Value					OPR = 3 (=)	46	Erythrocyte sedimentation rate
1323	INR	Haematology	2	Operator	OPR	Value					OPR = 3 (=)	71	International normalised ratio
2000	Red cell distribution width	Haematology	2	Operator	OPR	Value					not available in structured data	64	Red blood cell distribution width
2006	Weeks gestation	Maternity	1	Weeks gestation								55352	Gestational age
2007	Estimated date of delivery	Maternity	1	Estimated date of delivery	YYYYMMDD							8879	Estimated date of delivery
2008	Last menstrual period	Maternity	1	Last menstrual period	YYYYMMDD							6769	Last menstrual period -1st day
1002	Medical History	Medical History	4	Date of event	YYYYMMDD	Duration		Duration units:	SUM	Age in years			(extracted from free text)
1085	Absence of Condition	Medical History	1	Medcode For Condition	Medical Dictionary							72907	Negative
1087	Family History	Medical History	1	Medcode For Condition	Medical Dictionary							17485	[V]Health problems in family
2002	Negative past medical history	Medical History	1	Medcode For Condition	Medical Dictionary						negative past medical history	11435	No relevant past medical hist.
2003	Negative family history	Medical History	1	Medcode For Condition	Medical Dictionary						negative family history	13240	No relevant family history
2004	Suspected condition	Medical History	1	Medcode For Condition	Medical Dictionary							5494	[V]Observation and evaluation for suspected conditions
2005	Current or previous diagnosis	Medical History	4	Date of event	YYYYMMDD	Duration		Duration units	SUM	Age in years			(extracted from free text)
2009	Hospital admission	Medical History	2	Admission date	YYYYMMDD	Discharge date	YYYYMMDD				hospital admission. admit date must be before discharge date	43828	Hospital admission note
2010	Sickness certificate	Medical History	3	Until date	YYYYMMDD	Duration		Duration units	SUM			5761	[V]Issue of medical certificate
2013	Follow up	Medical History	3	Follow-up date	YYYYMMDD	Follow-up interval		Follow-up interval units	SUM			1793	Medical follow-up

The virtual Read terms in Table 1 have been supplied in the **virtualterms.txt** lookup to encode conditions that currently do not have specific Read terms.

medcode	virtual Read term
Coronary anatomy	
500000	Left anterior descending disease
500001	Left main stem disease
500002	Left coronary artery disease
500003	Right coronary artery disease
500004	Posterior descending artery disease
500009	Circumflex coronary artery disease
Left ventricular function	
500005	Good left ventricular function
500006	Mildly impaired left ventricular function
500007	Moderately impaired left ventricular function
500008	Severely impaired left ventricular function

Table 1: Virtual Read terms from the `virtualterms` table

4 Algorithm

The program is designed to extract diagnostic Read terms and several other types of structured data from unstructured free text.

4.1 Analysis modes

The analysis mode can be selected automatically based on the category of Read term associated with the text. This has been allocated in the **nativeterms** table (see subsection 5.7).

D – Death searches for the cause of death and interprets 1a, 1b etc. as death certificate entries. Laboratory results are not extracted.

P – Pregnancy a duration given in weeks is interpreted as gestational age if it is less than 43 weeks.

L – Labtest a numerical value or 'normal', 'abnormal' etc. can be interpreted as the test result

N – Normal / abnormal 'normal', 'abnormal', 'nad', 'positive' etc. can be interpreted as the investigation result, but numerical values cannot.

T – Date only a single date is allowed in the output.

S – Sicknote dates are regarded as medical certificate start or end dates.

M – Medical dates are regarded as medical certificate start or end dates.

4.2 Rationale for design of the system

The UK Clinical Practice Research Datalink (CPRD) contains a large database of primary care records and is an important source of clinical information for epidemiology and drug safety research. It contains details of consultations, diagnoses, test results, prescriptions and referrals. General practitioners (GPs) code important diagnoses using a structured clinical terminology. Currently the 'Read' clinical terminology is used, but OXMIS (Oxford Medical Information System) was used previously, and SNOMED-CT (Systematized Nomenclature of Medicine–Clinical Terms) will be used in the future. Additional information is entered in free text associated with the coded entries.

Our aim was to develop a natural language processing system to extract diagnoses as Read terms from free text in the CPRD, thus allowing researchers to combine information in coded and unstructured data in research studies using primary care data. We chose to develop our system independently rather than adapting an existing system so that we would have access to all the code and would be able to optimise its performance. Eventually it may be possible to use this software for data entry by doctors, enabling clinical information to be coded at the point of care with minimal cost on the doctor's time.

The Freetext Matching Algorithm uses manually entered lookup tables of phrases and synonyms, and simple semantic information from the Read terms themselves (e.g. negation) to identify appropriate Read terms for diagnoses stated in the text. The algorithm was developed by an iterative process. After writing the initial program, we used it to analyse randomly chosen free text entries, and reviewed the output manually. As well as the final structured output, the program can produce a report of the intermediate stages of analysis. We modified the lookup tables and program code based on the results of each test, re-tested the program on the same sample to verify that the errors had been resolved, and then tested it on a new sample of texts.

4.3 Clinical terminology

The algorithm was principally designed to encode diagnoses in the free text to terms in the Read Clinical Terminology, which is the system used for the existing coded entries in UK general practice. Apart from diagnoses, the Read terminology includes codes for other categories of information such as history, examination findings, procedures and test results. Our algorithm was designed to extract some of these entries but the main focus was on diagnoses.

We standardised the wording of Read terms by replacing abbreviations such as 'a/n' (antenatal) with the full word, and removing phrases such as 'NEC' (not elsewhere classified) which would not be found in ordinary clinical text. If some Read terms became identical after this process of standardisation, only one of them was retained. We categorised each word in a Read term as positive, negative, optional or ignorable. For example, for the Read term K510000 'Cystocele without uterine prolapse', the word 'cystocele' would have to have a positive attribute, 'uterine prolapse' would have a negative attribute and 'without' would be ignored. We used these allocations to define which words in a Read term need to be present in the text in order for the term to be matched. For example, in order to match the Read term B723z00 'Benign neoplasm of

bronchus or lung NOS', a phrase would only need to include one of the words 'bronchus' or 'lung'. Short words which would not alter the meaning of the term if omitted (e.g. 'of', 'or' and 'NOS' in the example), and words which influence the true / false status of nearby words but have no other meaning (e.g. 'lack of') were designated as ignorable and did not need to be present in the text.

We wrote a function called 'readscore' to grade the closeness of a match between a text phrase and a Read term, with a minimum threshold for a satisfactory match. The function attempts to map each word or phrase in the text to a word or phrase in the Read term, and vice versa, in order to verify that they have the same meaning. Points are deducted from the readscore for each ignorable or negative word not matched, and for use of synonyms rather than identical words. The algorithm tries a number of possible matches for a text phrase and choose the match with the highest readscore. For example, the text phrase 'Benign neoplasm of lung' would preferentially be matched to the Read term with the same wording, even though it could also match 'Benign neoplasm of bronchus or lung NOS' with a lower readscore. However, for the text phrase 'Benign neoplasm of bronchus', there is no specific term, so the only valid match would be with the Read term 'Benign neoplasm of bronchus or lung NOS'.

4.3.1 Selection of terms

We manually defined a subset of terms which the algorithm was allowed to select. Terms with more than 5 non-ignorable words were excluded as they are too long and complex to match and are infrequently used. This is particularly the case for Read Chapter 'T', which contains over 3000 terms describing specific (and often rare) external causes of injury, e.g. 'T546000 Sucked into jet - occupant of spacecraft injured'.

At the time of release of the Freetext Matching Algorithm Version 15, the lookups contained 81613 native terms of which 28005 were available for mapping, and 10 virtual terms (listed in Table 1).

4.3.2 Alternative forms of terms

The OXMIS dictionary is no longer used by GPs to encode information, but the OXMIS to Read mapping was used to designate some OXMIS terms as alternative forms of Read terms. They are included in the 'alternateterms' lookup table, which contains 10734 alternative forms of terms at the time of this release.

4.3.3 Adding new codes

Our system was designed to enable the easy addition of new codes, which may be useful for coding emerging diseases even before they are recognised in official coding terminologies. The method for doing so is described in subsection 8.2.

4.4 Analysis sequence

Figure 3 gives an overview of the analysis sequence, which is described in more detail in the following sections. We will consider the analysis of the text "START_1.00 Previpus MI in 2003".

Analysing this in test mode produces the following output in the log file:

```
Analysing a single text:
START_1.00 Previpus MI in 2003

INITIAL_SEARCH, ATTRIB.PD_SEARCH2
Attrib phrase (search position 550): 8|had|several|freq|frequent|prev|...
Matches to: previous mi
Attrib phrase (search position 706): 1|[CLIN] 1|* 1|[DATE]*
Matches to: previous mi in
Attrib phrase (search position 706): 1|[CLIN] 1|* 1|[DATE]*
Matches to: mi in 2003
Attrib phrase (search position 707): 1|[CLIN] 1|[DATE]*
Matches to: mi in
Attrib phrase (search position 988): 1|[NUMB_1_4] 1|previous* 1|[CLIN]*
Matches to: 1.00 previous mi
Word : Punctuation : Meaning : Attribute
start : _ : WORD start :
1.00 :  : NUMB 1.00 :
previous :  : CLIN previous : pmh
mi :  : CLIN 4 1 : pmh
in :  : DATE_year 2003 : dateprev
2003 :  : DATE_year 2003 : dateprev

ATTRIB_SEARCH, ANALYSE_PD
List of candidate terms
0 0  [1.00 previous mi]
0 0  [1.00 history of mi ]
0 0  [1.00 previous myocardial infarction ]
0 0  [1.00 previous acute myocardial infarction ]
```

(some output omitted)

```
List of candidate terms
0 0  [previous]
0 0  [history of ]
0 0  [history of ]
0 0  [h / o ]
Total 3 terms
```

```
List of candidate terms
0 0  [mi]
0 0  [myocardial infarction ]
96 14658  acute myocardial infarction
          [acute myocardial infarction]
Total 2 terms
Word : Punctuation : Meaning : Attribute
start : _ : WORD start :
1.00 :  : NUMB 1.00 :
previous :  : CLIN previous : pmh
mi :  : READ 14658 96 : pmh
in :  : DATE_year 2003 : dateprev
2003 :  : DATE_year 2003 : dateprev

PD.COMPRESS, PD.CHECK_COMPRESSED
Word : Punctuation : Meaning : Attribute
 :  : READ 14658 : pmh
 :  : DATE_year 2003 : dateprev

Output:
pracid,textid,medcode,enttype,data1,data2,data3,data4,std_term
0,0,,14658,1002,2003,,,, acute myocardial infarction
```

4.5 Key procedures in the program

4.5.1 Sub do_analysis

The Sub **do_analysis** in module **fma_gold** handles files in CPRD GOLD format, and outputs the structured data in a similar format. It loads text files containing practice ID, text ID and free text, along with a table of practice ID, text ID and associated medcode, to give the context for the interpretation of the text. It calls **main_termref** to perform the actual analysis. It calls **pd_to_fma_gold** to convert the output format into CPRD GOLD type format as described in section 3.

In test mode it can also be called with an actual free text as the argument, in which case the input files are ignored.

If this program is to be used with another data source (e.g. text entered at the keyboard, or another type of electronic health record), this function could be bypassed.

4.5.2 Sub main_termref

This Sub calls **main_analyse** (see below) with the appropriate analysis options based on the termref of the original Read term.

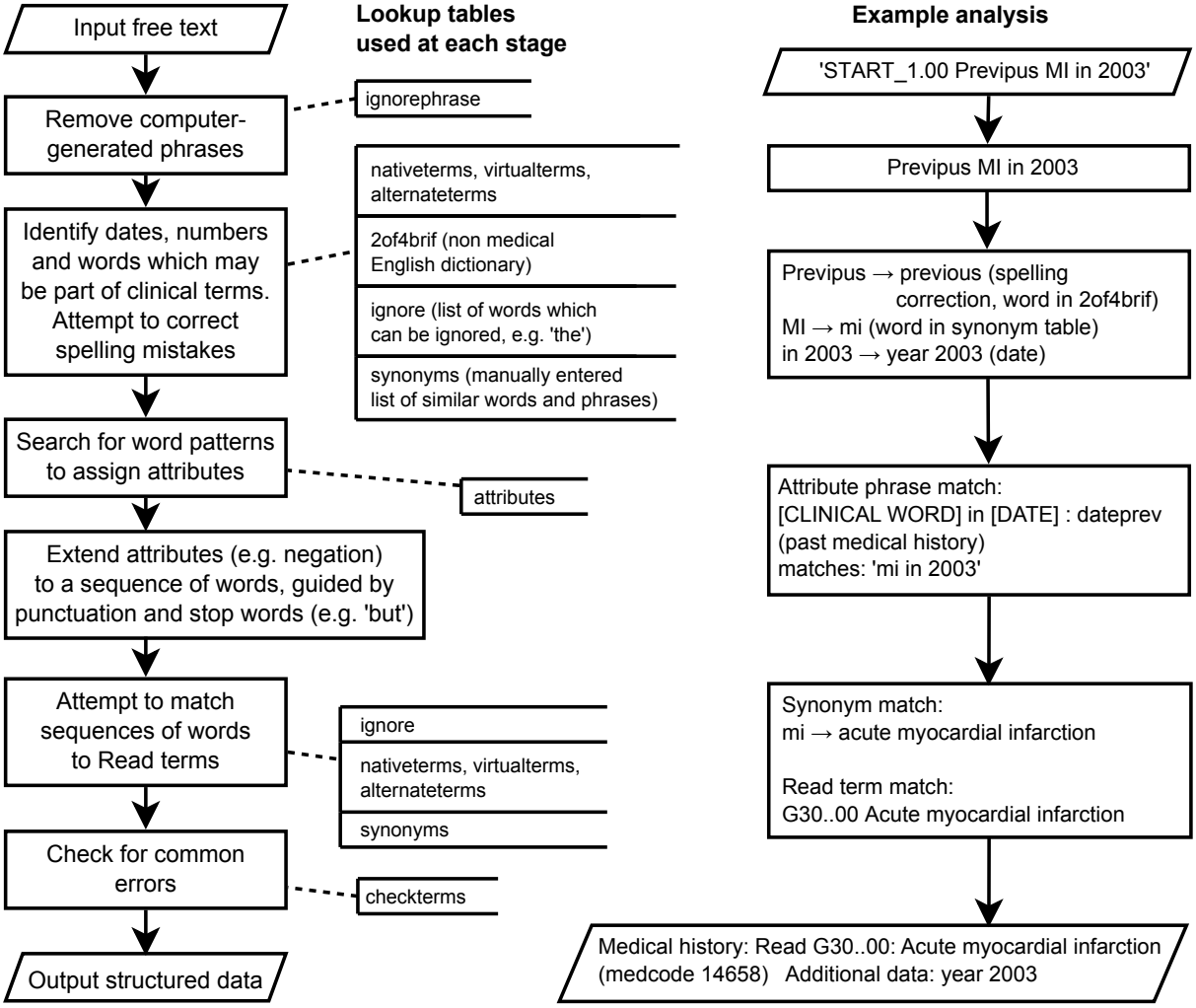

Figure 3: Overview of the steps involved in analysing a text, showing the lookup tables used at each stage and an example analysis

One of the options is **append**, which can be set to TRUE if the free text should be appended to the Read term text (to appear as it would on the doctor's computer). Text is not appended if the Read term type is LABS, DATE or SICKNOTE. The Read term is analysed separately, and the first interpreted value from the main text is removed if it is the same as that from the Read term alone or the existing termref and there is no attribute.

4.5.3 Sub main_analyse

Carries out the analysis of **instring** according to the analysis options. The results are stored in the arrays in module **pd**. If **debug_** is TRUE, the intermediate processes are documented in the global variable **debug_string** (see subsection 7.1).

1. Meaningless computer-generated phrases (e.g. "Hide=N") are removed from the free text by **remove_ignore_phrases**. If the text appears to be from a structured data area, the function **wordlist.initial_process** extracts the useful information and converts it to

a form suitable for further analysis.

2. The words and punctuation are assigned to arrays (module **pd**; by Sub **pd.init_read**.
3. The array of words is searched for dates, numbers, and entries in the synonym, ignore and wordlist tables. A data type is assigned to each word, which is **CLIN** (i.e. may be part of a Read term) for words in the synonym and terms tables. This search is carried out by Sub **initial_search**.
4. The procedure **attrib.pd_search2** uses the **attributes** table to find matching patterns, and attributes are assigned to phrases which match (see subsection 5.3).
5. The attributes are extended to nearby words according to certain rules in the program (e.g. a **negative** attribute is continued until a full stop, colon or the word 'but' is reached).
6. The text is analysed in sequences of up to 5 words with the data type **CLIN**. The words may be adjacent or may have 'ignore' words in between, such as 'the', 'and' etc. This is carried out by Sub **attrib_search**.
7. Call **analyse_pd** to attempt to match sequences of clinical words to Read terms. The program tries to find a Read term which matches the largest possible number of adjacent words, using the synonym table to link alternative phrases with the same meaning. Each word or phrase in the text is mapped to an individual word or phrase in the Read term, so the order of words does not matter. Negative parts of the text must map to negative parts of the Read term.
8. Call **pd.compress** to filter to include only the structured data extracted. The output is condensed into a sequence of converted dates and Read terms with associated attributes. In some cases the output consists solely of an attribute, e.g. if the text consists of "father" then the output is **ATTR family** without a Read term.
9. Call **checkterms.check_all** for a final validity check for a selected set of Read terms, using the **checkterms** table (see subsection 5.4).
10. Call **pd.check_compressed** to check that each structured data element has an appropriate attribute and value for its data type.
11. Convert the output to CPRD GOLD format, using functions in the **fma_gold** module.

Before the final step, the intermediate result is a list of structured data elements each of which has a data type, attribute and value, as shown in Table 2. The final step can be replaced by an alternative output format if desired, by writing another function to call **main_analyse**.

5 Lookup tables

5.1 2of4brif

A list of British English words from `http://wordlist.sourceforge.net/`.

Data type	*Possible attributes*	*Value*	*Description*
READ	*Death mode:* **Deathcause, deathcause1a, deathcause2** etc. *Others:* **Family, negfamily, query** + others	**Termref_uid** of matched Read term	Match to Read diagnostic term (i.e. lettered chapter). Tests, family history, personal history, investigations and administrative terms are currently not used.
DATE_full, DATE_year, DATE_time	*Death mode:* **deathdate, certdate** *Others:* **admitdate, followup, dob, edd, lmp, dateprev, datenext**	Date	Date in various formats.
DURA_yrs_, DURA_mths, DURA_wks_, DURA_days	**duraprev, duranext, followup, age**	Number	Duration in various formats.
LABS	**gest, sysbp, diabp, haemoglobin, mcv, pulse** and others	Numerical value or 'normal', 'abnormal', 'low', 'high'	Laboratory values. 'gest' is gestational age in weeks.
ATTR	**family, negative, query**	none	If the text suggests that the Read Term does not refer to a definite diagnosis for this patient (e.g. "*Death* of mother", "*Pneumonia* possible")

Table 2: Intermediate structured data format

5.2 alternateterms

Exact synonyms for native or virtual Read terms, which assist in the matching but are not returned as output.

Format: Tab-separated text file with no quotes, sorted by medcode

medcode – unique identifier for the term, starting from 600000

stdterm – lower case term to match, with one space before and after the term, one space between each pair of words and no punctuation

attrstring – a string with as many characters as the number of words in stdterm, stating how each word should be interpreted:

T – true

F – false

I – ignorable

O – option

linkto – medcode of the native or virtual Read term that is returned by the algorithm if this term is matched

comment

5.3 attributes

The attribute search takes place *after* detection of dates and durations but *before* analysis of Read terms. However, words which might form part of a Read term are marked with data type **`CLIN`**. Lab results are extracted during the attribute search. Some attributes are further manipulated by the core program.

Each row of the **`attributes`** table contains up to 5 words with punctuation, and the attributes to which they map. For each word it is possible to specify a choice of words or data types which are acceptable for the match.

The *search position* is the order in which the patterns are used; those with higher search positions are used first. Attributes of patterns which are used later (i.e. with lower numbers) can overwrite attributes set by earlier patterns.

Format: Tab-separated text file with no quotes, sorted by order

order – sort order (lowest to highest). Any additional entries must have an appropriate sort order, using decimals as needed so that the order of existing entries does not need to change

w1 – expression specifying which word to match. A number of options can be supplied separated by |. If a data type rather than a specific word is given, it must be in square brackets. See Table 3 for full details.

p1 – one or more characters specifying which punctuation characters (in any order) can be found after the word for it to match. For example, if the pattern is '_:=' then ':', '=', ':=' or no punctuation would be allowed. Special codes:

***** for any punctuation or blank,

_ for no punctuation

w2 ... p5 – alternating word and punctuation, to define a pattern of up to 5 words

a1 ... a5 – context attribute to set for each word in the pattern. The following special codes can also be used:

anon Anonymise; do not analyse this part of the text (e.g. if it may contain the name of a doctor, patient or hospital)

ignore Ignore

normalrange Ignore any following lab values unless they have their own attribute

possibility Ignore any following diagnoses unless they have their own attribute

_ (underscore) Retain current attribute

. (full stop) Set attribute to blank

DATATYPE attribute Set attribute to **`attribute`** and data type to **`DATATYPE`** (e.g. **`LABS inr`**; **`DURA_wks_ followup`**)

death_only – TRUE for attributes which are only relevant to texts associated with death, FALSE otherwise

comment

5.4 checkterms

Phrases which must be present or must not be present in the text for particular medcodes to be valid. This may be used for diagnoses which are commonly stated in the context of screening or immunisation, or for Read terms which are commonly extracted using abbreviations which are only valid in particular contexts. A medcode can only have 'qualifying' or 'dequalifying' phrases; it does not make sense to have both.

Format: Tab-separated text file with no quotes, sorted by medcode

medcode – unique identifier for a Read term

qualify – lower case comma separated word or phrase fragments which are necessary for the Read term to be valid

dequalify – lower case comma separated word or phrase fragments which invalidate the Read term

Example or code	*Meaning*
*	Any word or punctuation
thisword	Specific word: 'thisword'
this|that|[NUMB]	Either 'this', 'that' or a number
2	Specific number: 2
[CLIN]	Any word which might be part of a Read term
[IGNO]	An ignorable word, e.g. 'and', 'as', 'at', 'by'
[NUMB]	Any word with data type 'number', which includes some lab results such as 'normal'
[NUMB_70_230]	Number between 70 and 230
[DATE]	Date in any format
[DATE_full]	Full date
[DATE_year]	Year
[DURA]	Duration in any format
[DURA_wks_]	Durations in weeks
[ATTR attribute]	Any word with specified attribute

Table 3: Attribute patterns

5.5 ignore

Words or phrases which can be omitted without significantly altering the meaning of most Read terms or clinical text phrases.

Format: lower case words or phrases, with one space between words and no punctuation, sorted alphabetically

5.6 ignorephrase

Verbatim semi-structured text phrases which may appear in free text in Vision systems. START_ is used to match the beginning of the text.

Format: words or phrases with punctuation, case sensitive

5.7 nativeterms

Read terms in the Clinical Practice Research Datalink Read dictionary.

Format: Tab-separated text file with no quotes, sorted by medcode

medcode – unique identifier for the term, starting from 1

readcode – 7-character Read code, unique for each term (for reference, not used by program)

term – Read term (for reference, not used by program)

stdterm – standardised lower case version of Read term, with one space before and after the term, one space between each pair of words and no punctuation

attrstring – a string with as many characters as the number of words in stdterm, stating how each word should be interpreted:

- **T** – true
- **F** – false
- **I** – ignorable
- **O** – option

include – TRUE if the program is permitted to map to the term, FALSE otherwise.

type – a single character or blank, denoting the category of Read term in order to trigger the appropriate analysis mode:

- **D** – death; allow detection of cause of death
- **L** – laboratory result
- **M** – Read diagnosis term
- **N** – test result which may be qualitative; the result may be 'normal' or 'abnormal'
- **P** – pregnancy term; numbers such as 28/40 are interpreted as gestational age
- **S** – sick note; duration may be state
- **T** – date or time

comment

5.8 synonyms

Words or phrases which may be matched in Read terms and the original free text.

Format: Tab-separated text file with no quotes, sorted by text, then read, then priority

text – word or phrase in the free text

read – word or phrase in a Read term

priority – a number between 1 and 5:

- **1** loosely associated (read phrase broader than text phrase) e.g. foot (is a part of) = lower limb
- **2** non-standard abbreviation or distorted form; possible one-way match (read phrase broader than text phrase) e.g. rsi = repetitive strain injury
- **3** moderate match e.g. b pne = bronchopneumonia; carcinoma (is a type of) = malignant neoplasm

4 almost exact match e.g. cancer = malignant neoplasm

5 exact match e.g. chronic obstructive pulmonary disease = copd

-100 opposites e.g. left / right

comment

5.9 virtualterms

Terms created to code useful concepts in the output for which there are no current Read terms.

Format: Tab-separated text file with no quotes, sorted by medcode

medcode – unique identifier for the term, starting from 500000

term – term description

stdterm – standardised lower case version of term, with one space before and after the term, one space between each pair of words and no punctuation

attrstring – a string with as many characters as the number of words in stdterm, stating how each word should be interpreted:

T – true

F – false

I – ignorable

O – option

comment

6 Attributes

The general format of the attributes table is in subsection 5.3. This section describes the available attributes associated with different data types.

6.1 Read terms

deathcause1a, deathcause1b, deathcause1c Cause of death – death certificate categories; e.g. '1a) MI b) coronary atheroma' → 'MI' has attribute **`deathcause1a`**; 'coronary atheroma' is **`deathcause1b`**

deathcause1c

deathcause1

deathcause2

deathcause Specifically stated as cause of death; e.g. “Cause of death: bronchopneumonia”

negative Associated clinical term is negative, e.g. “not cancer”

family Clinical term is associated with family member, not patient e.g. “wife has cancer”

negfamily negative family history; e.g. “no family history of stroke”

pmh past medical history e.g. “asthma age 7”

negpmh negative past medical history e.g. “no previous MI”

query Uncertainty about diagnosis (’query’ or ‘rule out’), e.g. “rule out MI”

dueto previous condition was caused by this condition, e.g. “MI due to atherosclerosis”

causing this condition was caused by previous condition e.g. “Septicaemia complicated by renal failure”

6.2 Dates

certdate date when death was certified

admitdate admission or readmission date

dischdate discharge date

deathdate death date

lmp last menstrual period

edd expected date of delivery

datenext date refers to next Read term (e.g. “1990 stroke”)

dateprev date refers to preceding Read term (e.g. “MI in 1982”)

followup follow up date

sicknote any date in a medical certificate entry (may be start or end date of certificate)

6.3 Duration

duranext duration refers to next Read term

duraprev duration refers to preceding Read term

followup follow up time (e.g. “see in 3 months”)

age e.g. “this 40-year-old man”

ageprev age at event relating to previous Read term, e.g. “diagnosed with asthma aged 10”

sicknote e.g. “*MED3* 1 week”

6.4 Lab tests

Laboratory results are not extracted in 'Death' analysis mode.

calcium

cholestesterol

cobalamin Vitamin B12

creatinine

diabp, sysbp Blood pressure. The program recognises a format such as '150/80' without 'blood pressure' stated explicitly, but only if the systolic pressure is higher than diastolic and both are in sensible ranges (`sysbp` 80–230, `diabp` 40–150)

esr Erythrocyte sedimentation rate

fbc Full blood count

ferritin

folate

gest Gestational age (duration in weeks, less than 43. In 'pregnant' mode, the program will interpret any duration in weeks as gestational age, as long as it does not have another attribute, and there no different duration in the text. Fractions are ignored by the function **`strfunc.get_date`** which converts durations into a structured format, e.g. "32/40 + 6" is converted to '32 weeks'.)

glyhb HbA1c

haemoglobin

hdl HDL cholesterol

inr International normalised ratio

ldl LDL cholesterol

mcv Mean cell volume

pefr Peak flow ('predicted' or 'best' peak flow is ignored)

platelets

pulse pulse rate in beats per minute, must be within the range 20 to 300

rdw red cell distribution width (not available in structured data in CPRD, but is quite commonly recorded in the free text)

tetrathyroid Thyroxine, T4

trithyroid T3

tsh Thyroid stimulating hormone

triglycerides

urea

wbc White blood cells, leucocytes (may apply to blood, urine or other fluids, depending on associated Read term)

The function **pd.correct_attr** controls whether an attribute is correct for a particular meaning, except for LABS attributes. It is possible to detect a new LABS data type by adding an entry to the **attributes** table, but this will not be output in GOLD format unless an output statement is added in the function **pd_to_fma_gold** (see subsection 8.3).

7 Testing the algorithm

The test mode of the program is invoked by specifying actual free text rather than a file to analyse. The structured output along with an analysis report is printed to the log file.

7.1 Analysis reports

Module **freetext_core** contains a global variable 'debug_string', which is used to store an analysis report in test mode. Function can add entries to debug_string to document the stages of analysis. Line breaks are inserted by appending ASCII character 13 then character 10.

This example is based on the following free text:

"MI in 2011, chest pain on exertion ever since."

The analysis report contains the following items (some parts omitted to save space):

1. Heading `INITIAL_SEARCH, ATTRIB.PD_SEARCH2`
2. Attribute patterns which match to the text (show attrib phrase 'Matches to:' text phrase). Example:

   ```
   Attrib phrase (search position 706): 1|[CLIN] 1|* 1|[DATE]*
   Matches to: mi in 2011
   Attrib phrase (search position 707): 1|[CLIN] 1|[DATE]*
   Matches to: mi in
   ```

   Utility: shows which attribute patterns were used and the text they recognised. If a context is detected incorrectly, this part of the report shows whether the context was detected in the first place. Modification or addition of patterns to the attribute table might reduce errors seen at this stage.
3. Listing of arrays in module pd, containing words, punctuation, attribute and meaning in separate arrays. This listing is after the words have been given provisional data types, and the attributes have been allocated according to the arrays. Example:

```
Word : Punctuation : Meaning : Attribute
mi :  : CLIN 4 1 :
in :  : DATE_year 2011 : dateprev
2011 : , : DATE_year 2011 : dateprev
chest :  : CLIN 2 1 :
pain :  : CLIN 4 1 :
on :  : IGNO : ignore
exertion :  : CLIN 5 1 :
ever :  : WORD ever :
since : . : WORD since :
```

Utility: 'Word' column shows text after **remove_ignore_phrases** and **initial_process**. 'Meaning' shows initial allocation of data types, particularly dates (function **strfunc.get_date** is called to try to extract a date from every sequence of up to 5 words). 'Attribute' shows the result of **attrib.pd_search2** i.e. after recognition of patterns using the attribute table.

4. Heading ATTRIB_SEARCH, ANALYSE_PD

5. Show each sequence of words of data type 'CLIN' (possibly with 'IGNO' or 'NUMB' words in between) tested i.e. words which might be part of a Read term. Each section headed 'List of candidate terms' is the record of a single call to the function 'bestmatch'. This function is called with a sequence of up to 8 contiguous words from the text, and returns the match with the highest readscore or the first match found with readscore higher than threshold_high. If a Read match is found, the report contains the readscore, medcode, Read term text, and the text to which it matches.

   Example of listing:

```
List of candidate terms
0 0  [mi]
0 0  [myocardial infarction ]
96 14658  acute myocardial infarction
          [acute myocardial infarction]
```

   Utility: Shows which sequences of words were chosen for conversion to Read terms (sub **freetext_core.analyse_pd**). Shows which alternative texts were generated by using the synonym table. Shows the converted text from which the Read term match was made, and the readscore (calculated with reference to the original text). Errors at this stage might be reduced by adding or editing synonym entries.

6. Listing of pd arrays, now containing linked Read termref Uids alongside text. Example:

```
Word : Punctuation : Meaning : Attribute
another :  : CLIN another :
hospital :  : READ 309362 100 :
admission : - : READ 309362 100 :
still :  : READ 309362 100 :
having :  : WORD having :
```

```
⋮
and :  : IGNO : ignore
rev :  : WORD rev : followup
1 :  : DURA_mths 1 : followup
m :  :  : followup
if :  : WORD if : followup
no :  : IGNO : possibility
```

Utility: shows the original text which was linked to a Read term.

7. Heading: `PD.COMPRESS, PD.CHECK_COMPRESSED`
8. Listing of pd arrays, now containing one row per structured data element. Example:

```
Word : Punctuation : Meaning : Attribute
:  : READ 309362 100 :
:  : DURA_mths 1 : followup
```

Utility: useful for checking the effect of the sub **`pd.check_compressed`**, which removes attributes or data elements which do not make sense.

8 Modifying the algorithm

The Freetext Matching Algorithm is easy to adapt to detect new types of information or avoid errors (false positives or false negatives).

8.1 Allowing additional existing Read terms in the output

The **`nativeterms`** table contains all Read terms in GPRD so that it can select an appropriate analysis mode and append the text to the Read term for analysis. However the Freetext Matching Algorithm currently uses only a subset of Read terms for coding, for reasons stated in subsubsection 4.3.1.

Additional terms can be added by changing the entry in the 'include' column of the **`nativeterms`** lookup from FALSE to TRUE.

When including a new Read term in this way, check that the **`stdterm`** and **`attrstring`** are correct. The **`stdterm`** column should contain a lower case version of the Read term excluding words which are not necessary for matching, with no punctuation, one space between words and a space at the beginning and end of the term. The format of **`attrstring`** is described in subsection 5.7; it was provisionally assigned to non-included terms using a semi-automated process and may need manual checking.

In order to improve detection of the new Read term, it may be necessary to add alternate forms of the term to the **`alternateterms`** table (see subsection 5.2). The medcode column must be in

ascending order starting from 6000000, and each term should contain the medcode of the linked preferred term in the **linkto** column.

8.2 Adding new Read terms

New 'virtual' Read terms can be added to the **virtualterms** table, sorted by medcode, with medcode in the range 500000–599999. It may be necessary to add alternate forms of this term or synonyms.

8.3 Adding new laboratory results or clinical measurements

Numerical results can be detected by patterns in the **attributes** table with the data type **LABS** and a specified attribute. A new data type can be detected by adding an appropriate entry to the **attributes** table. The first column **order** should contain a number representing the sort order (rows with lower numbers are used first, and their results can be over-written by rows with higher numbers).

For example, to detect sodium results, a possible entry could be:

```
order  w1         p1   w2             p2...a1  a2      ...
398.5  na|sodium _:=- [NUMB_90_180] * ...    LABS na...
```

(tab separated; not all columns shown)

The attributes table must be sorted by **order**, ascending.

In order to produce output in GOLD format, an entity type also needs to be created for the new data type. A line needs to be added to the Sub **pd_to_fma_gold** after **Select Case pd.Attr(i)**, with a **Case** statement for the new attribute and a call to **addOutputRow**.

8.4 Correcting errors in interpretation

If an error is detected in interpretation, the first step is to run the Freetext Matching Algorithm in test mode to obtain an analysis report (see subsection 7.1). If the result is a false positive, it may be because some Read terms or words are ambiguous, and it might be prevented by adding an entry to the **checkterms** table to disqualify detected Read terms in some circumstances, or require additional confirmatory phrases in the text. It might also be because of deficiencies in the main program logic, but this is more difficult to alter and requires a deeper understanding of the program.

If the error is a false negative, recall might be improved by adding additional alternate terms or synonyms for the text.

8.4.1 Adding entries to the checkterms table

Insert a tab-separated entry in the appropriate place so that it is sorted by medcode. The columns are: medcode, qualify, dequalify. Only one of qualify and dequalify should be specified, and it should consist of a comma separated list of word fragments or phrases, e.g. the Read term 19418 (A705400 'Hepatitis non A non B') is invalid if the text is referring to testing or immunisation.

```
medcode  qualify  dequalify
19418             vacc,immuni,travel,boost,antibody,antigen
```

8.4.2 Adding synonyms

Add a row to the synonym table in the format described in subsection 5.8, optionally with a comment in the last column.

For merging with the existing syonyms table it is recommended that the table is sorted by so that the **`diff`** program can easily find which lines have been changed (and Git can track these changes), but for the algorithm itself it does not matter if the table is sorted.