MetaCoDe

A GATE PLUGIN FOR TAGGING MEDICAL CORPORA IN FRENCH WITH CONTROLED TERMINOLOGIES

Thierry Delbecque
thierry.delbecque@semantic-valley.org
Pierre Zweigenbaum
pz@limsi.fr
Motivations:

To provide a UMLS mapping tool that can process large corpora in French:

- Operating at an high enough speed
- Able to operate 'on the fly', for example for processing stream of data from Internet
- Using limited computing resources
- Saving resources for others algorithms such as Machine Learning
- Built in an open platform and easily integrated in bigger applications
- Scalable and easily maintainable
Principles:

• To implement the software as a GATE java plugin, in order to ensure inter-operability of the tagger with other software components (through the java API, or through the GATE standard representation of annotated texts)

• To code linguistic preprocessing (tokenization, noun phrases extractions, etc) with GATE processing resources (JAPE Grammars, ...)

• To code the core extraction algorithm in C++, to speedup the computation time

• In memory loading of strictly necessary knowledge resources as hash tables: no file or data base accesses during the tagging process

• To rely only on the UMLS Metathesaurus strings to handle variant generation (no specific variant generation algorithm
Architecture

Processing Resources

- POS extraction (Tree Tagger, Gazeeter, JAPE)
- NP Extraction (JAPE)
- Core Tagger (C++ native library)

Language resources

- Initial Corpus
- Annotations
- Annotations
- MetaCoDe Annotations

Visualization resources

- MetaCoDe Browser

UMLS Data Bases

SoDAD INSIGHT
Data Mining & Information Systems
Extracted In-Memory Resources

Ad hoc resources can be extracted according to the processed corpora and loaded in memory:

- **WDSUI**: mapping from individual words to SUI strings;
- **CUISUI**: mapping from each SUI to its corresponding CUI;
- **SUILENGTH**: lengths (token number) of each SUI;
- **CUISTY**: semantic types of each CUI.

Specific vocabularies can be selected.
Mapping Engine Algorithm

**Step 1:** Noun Phrase Extraction (GATE PR). Based on POS extraction and JAPE grammar => each noun phrase is latter represented as a bag of words \( \{w_1, ..., w_n\} \)

**Step 2:** For each noun phrase, selection of candidates SUI's based on the bag of words => \( C_1 = \{sui_1, ..., sui_p\} \)

**Step 3:** Build a lattice on \( C_1 \); pruning of the lattice to keep the SUI's with wider coverage of the noun phrase without being too specific => \( C_2 \)

**Last Step:** for each element of \( C_2 \), retrieves its corresponding CUI and semantic types from UMLS data base
Mapping Engine Algorithm
SUIs selection for a given Noun Phrase

(a) : Build the lattice of all SUIs
That contains at least one word of the NP

(b) Pruning.
Keeps only those SUI S such that
|S| <= |S ∩ NP| + tolerance

Noun Phrase
Eg “occult nodal metastase”

SUI1
(S0443268 = “nodal”)

SUI2
(S0421910 = “metastasis”)

SUI

SUI

SUI

SUI

SUI

=> C₁ = {SUI1, SUI2}
Evaluation

Motivation: to evaluate the price to pay for not computing variants generation as is currently done in Metamap;

Principle: to run MetaMap on an English corpus, and to use MetaMap as the gold standard; compare the output of both tools on a sample of 30 independent abstracts issued from Medline;
Evaluation: metrics

MATCH: a common decision has been issued by both tools;

AMBIGUOUS: given a Noun Phrase, MetaCoDe produced issued incorrect decisions along with correct ones;

BAD: given a Noun Phrase or a part of a Noun Phrase, MetaMap took a decision, according to which MetaCoDe produced only bad output;

MISS: no decision was output by MetaCoDe when MetaMap has been producing a mapping.
Evaluation Results

- $P = | \text{MATCH} | / ( | \text{MATCH} | + | \text{BAD} | ); \text{ (pseudo-precision)}$
- $R = | \text{MATCH} | / ( | \text{MATCH} | + | \text{BAD} | + | \text{MISS} | ); \text{ (pseudo-recall)}$
- $F = 2 \times P \times R / (P + R)$

<table>
<thead>
<tr>
<th>MATCH</th>
<th>AMB</th>
<th>BAD</th>
<th>MISS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1300</td>
<td>397</td>
<td>93</td>
<td>307</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>P</th>
<th>R</th>
<th>F</th>
<th>Amb.Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.93</td>
<td>0.76</td>
<td>0.83</td>
<td>0.31</td>
</tr>
</tbody>
</table>
Running MetaCoDe

Processing Resources Chain
Conclusions

• The good precision rate is due to the rather conservative algorithm of MetaCoDe

• Relying to the Metathesaurus only for variants generation leads to 0.76 “recall” rate but could be improved by allowing some verbal forms to be tagged as well

• The core tagging process is fast enough to process big corpora or to add extra algorithms for specific applications*, though GATE platform slows down the process significantly

• Young project, still a lot to be done on JAPE grammars and on the core tagger (not industrial today)

• The code is open source and available under GPL license

* 26 minutes spent for processing 7260 abstracts from MEDLINE (2160613 words), when called outside of GATE on a Pentium 4 biprocessor, 3Ghz, 1.5Go, Linux.
Rindflesch TC and Aronson AR
Semantic Processing in Information Retrieval.
Proc Annu Symp Comput Appl Med Care, Charles Safran (ed.), 1993

Bodenreider 0, Nelson SJ, Hole WT and Chang FH
Beyond synonymy: exploiting the UMLS semantics in mapping vocabularies.
Proc AMIA Symp. 1998

Aronson AR, Mork JG, Gay CW, Humphreys SM and Rogers WJ
The NLM Indexing Initiative's Medical text Indexer.
MedInfo. 2004

Humphreys BL, Lindberg DA, Schoolman HM and Barnett GO
The Unified Medical Language System: An informatic research collaboration.
J Am Med Inform Assoc, 1998;5(1)

Aronson AR
Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program.
Proc AMIA Symp. 2001

Lindberg DA, Humphreys BL and McCray AT
The Unified Medical Language System.
Methods Inf med, 1993;32(2)

Delbecque T and Zweigenbaum P
MetaCoDe: a lightweight UMLS mapping tool.
Actes 11th Conference on Artificial Intelligence in Medicine Europe, Heidelberg, 2007