CARAMBA
Concept, Assertion, and Relation Annotation using Machine-learning Based Approaches

Cyril Grouin MSc, Asma Ben Abacha MSc, Delphine Bernhard PhD, Bruno Cartoni PhD, Louise Deléger PhD, Brigitte Grau PhD, Anne-Laure Ligozat PhD, Anne-Lyse Minard MSc, Sophie Rosset PhD, Pierre Zweigenbaum PhD

LIMSI–CNRS, Orsay, France

November 12th, 2010 – i2b2/VA 2010 Workshop

Our system: CARAMBA

System description

- Creating **different systems** per task.
- Using **machine-learning** tools (CRF, SVM).
- Defining **post-editing rules** to refine results.

1. Introduction
2. Task 1. Concept extraction
3. Task 2. Assertion annotation
4. Task 3. Relation annotation
5. Conclusion

Task 1. Concept extraction

Medical NLP techniques

- **Machine-learning approaches:**
  → provide a fast path to results once corpora have been annotated.
- **Expert-knowledge-based techniques:**
  → time consuming but reliable results.

Concept extraction

- To define **rules** and **gazetteers**.
- To use **linguistic resources** obtained from the **UMLS**.
- To use discharge summaries **structure**.

Task 1. Concept extraction

System description (run C1 and C2)

- Limited linguistic analysis.
- Output represented as features:
  - **n-grams** of token.
  - **typographic clues** *(letter case, alphabetical/date/digit/ punctuation category)*.
  - **syntactic** and **semantic tags**.
- A machine-learning tool uses these features to make decisions on concept boundaries and types.

Task 1. Concept extraction

1. Linguistic analysis (run C1 and C2)

- **Morpho-syntactic analysis** (Tree Tagger)
  $\rightarrow$ POS tags and lemmas are thus associated to each token.
- **Semantic tagging** (specific lexicon of 62,263 adjectives and 320,013 nouns based on the UMLS Specialist Lexicon)
  $\rightarrow$ These lists specify the category of adjectives *(relational and qualitative adjectives)* and nouns *(proper name, countable and uncountable nouns)*, and their position in a sentence *(attributive, post-nominal, or predicative)*.

Task 1. Concept extraction

1. Linguistic analysis (run C1 and C2)

- Extension of the **semantic tagging** with 11 major semantic categories (from the UMLS, from Sager's work, and lists of medication names we had compiled for i2b2 2009):
  - anatomy,
  - laboratory analysis *(blood wbc, creatinine, hematocrit)* and examination *(angiography, biopsy, scan, x-ray)*,
  - pre- and post-mark of examination *(follow-up..., physical..., repeat..., ...culture, ...evaluation, ...levels)*,
  - general localization *(lower, up- per, right, left)*,
  - medication,
  - mode of administration,
  - medical object *(cannula, drain, pacemaker, stent)*,
  - procedure *(amputation, blood transfusion, dialysis)*,
  - dosage.

Task 1. Concept extraction

2. Machine-learning (run C1 and C2)

- **Training of a model** over the training corpus using CRF++ *(machine-learning tool based upon conditional random fields)*.
- **Application of this model** over the test corpus.
  → This pipeline was used for our first submission (run C1).

Task 1. Concept extraction

3. Post-editing rules (run C2)

- **Design of a few post-editing rules** to refine the output of this model.
  - A token with "medication" as feature is tagged as a treatment concept if not already detected.
  - We also tried to correct potentially misclassified medical concepts by selecting the most frequently assigned tag in cases where different concepts tags had been assigned.

→ This pipeline was used in our second submission (run C2).

Task 1. Concept extraction

MetaMap-based method (run C3)

- **MetaMap** localizes medical terms and their corresponding concepts and semantic types from the UMLS metathesaurus and semantic network.
- Some **residual problems:**
  - at the noun-phrase segmentation level
  - at the recognition of several known drugs, diseases and tests.
- **Enhancement of MetaMap's output** by performing two steps before the execution of MetaMap:
  - segmentation into noun-phrases with treetagger-chunker
  - search of the located terms in pre-compiled lists of medical problems, tests and treatments.
- **Final filtering** with lists of common errors and stopwords.

Task 1. Concept extraction

	Recall	Precision	F-measure
Run 1	0.723	0.825	0.772
Run 2	0.726	0.826	0.773
Run 3	0.420	0.495	0.454

Best run: #2

	Recall	Precision	F-measure
Problem	0.742	0.799	0.769
Treatment	0.723	0.843	0.778
Test	0.705	0.851	0.771

1. Introduction
2. Task 1. Concept extraction
3. Task 2. Assertion annotation
4. Task 3. Relation annotation
5. Conclusion

Task 2. Assertion annotation

System description (run A1 and A2)

We developed two systems for assertion annotation:

- the first one using **machine-learning techniques** (run A1) → assertion identification considered as a classification task, with the six assertion types as target classes.
- the second one using **manually-designed rules** (run A2).

Task 2. Assertion annotation

Machine-learning techniques (run A1)

- We trained an **SVM** with the libsvm tool based on **binary feature vectors**.
- Automatic selection of the **optimal parameter values using cross-validation**.
- Three types of features:
  - contextual lexical features
  - trigger-based features
  - target concept internal features

Task 2. Assertion annotation

Machine-learning techniques (run A1)

- **Contextual lexical features:**
  - token and stemmed token unigrams in a 5-word window to the left and to the right of the target concept,
- **Trigger-based features:**
  - phrases which are indicative of a given assertion class (triggers collected for our extension of GenConText, with few additions),
  - triggers before and after the problem concept, again in a 5-word window,
  - concept-internal triggers such as "on exertion" (indicative of the conditional assertion class when it occurs within an annotated concept).

Task 2. Assertion annotation

Machine-learning techniques (run A1)

- **Target concept internal features:**
  - problem tokens,
  - stemmed problem tokens,
  - and the presence of the "non" negative prefix in one of the problem words.

Task 2. Assertion annotation

Machine-learning techniques (run A1)

- **Many problem concepts are coordinated**: *"pleural effusion or pneumothorax"*.
  → These sequences might lead to obtaining reduced left and/or right context, containing mostly other coordinated problems.
  → In this case, important cues for a specific assertion type may fall outside the scope of the contextual window.
- **Pre-processing of the data** to identify coordinated problems and redefine the offsets for left and right token windows:
  - left windows end at the beginning of a list of coordinated problems,
  - right windows start at the end of the sequence.

  → These contexts are shared by all concepts occurring in the same coordinated sequence.

Task 2. Assertion annotation

Machine-learning techniques (run A1)

- **Specific features:** encode all coordinated problem words and **stems** occurring in the same sequence as the target concept. → For instance, given the concept sequence "*pleural effusion or pneumothorax*", for the concept "*pneumothorax*", following features are used: "*pleural*" and "*effusion*", as well as the stem "*effus*".

Task 2. Assertion annotation

Manually-designed techniques (run A2)

- The second system was based on an **extension of the NegEx algorithm**. It locates trigger terms indicating a negation or a probability and determines if the concepts fall within the scope of these triggers.
- The corpus was also pre-processed in order to cope with coordinations and to tag each concept with its type.
- We extended the General ConText Java implementation to deal with the categories *conditional*, *hypothetical* and *not associated with the patient*.

Task 2. Assertion annotation

	Recall	Precision	F-measure
Run 1	0.931	0.931	0.931
Run 2	0.882	0.882	0.882

Best run: #1

	Recall	Precision	F-measure
Present	0.970	0.942	0.956
Absent	0.947	0.931	0.939
Possible	0.538	0.738	0.622
Hypothetical	0.830	0.928	0.876
Conditional	0.240	0.745	0.363
Associated w/ se	0.779	0.856	0.816

Task 3. Relation annotation

System description (run R1, R2 and R3)

- Relation identification as a **classification task**.
- We used a **hybrid approach**: combines machine-learning techniques and linguisitic-pattern matching.
- We trained an SVM with the libsvm tool and constructed linguistic patterns manually.

Task 3. Relation annotation

System description (run R1, R2 and R3)

- Run R1: before the prediction of relation types with libsvm, we used patterns to identify 4 relations: TrIP, TrWP, TrNAP, and TeCP for which there are few examples in the training set.
- Run R2: supervised learning from simplified texts.
- Run R3: combination of results of run R1 and R2.

Task 3. Relation annotation

System description (run R1, R2 and R3)

- **Patterns:** After empirical observations we kept only the patterns of four relations types (TrIP, TrWP, TrNAP, TeCP) as the others did not offer satisfying results.
- **TrIP:** _PROB_{0,75 char} ((is|are|was|were))?ruled out (by|with){0,75 char}_TX_
- **TeCP:** _TE_{0,45 char} (in|for) the diagnosis (of)?{0,45 char} _PROB_

	Recall	**Precision**	**# examples**	**# patterns**
TrIP	0.35	0.45	74	43
TrWP	0.16	0.79	39	27
TrNAP	0.16	0.65	71	25
TeCP	0.08	0.60	196	56

Table: Precision and recall of patterns matching.

Task 3. Relation annotation

System description (run R1, R2 and R3)

- **Surface features:**
  - order of the candidate concepts,
  - distance between them (i.e. the number of tokens),
  - presence of other concepts,
  - type of the concepts (problem, test or treatment)
  - normalized title of the section.

Task 3. Relation annotation

System description (run R1, R2 and R3)

- **Lexical-semantic features:**
  - tokens and stemmed tokens in candidate concepts,
  - left and right trigrams (of stemmed tokens) of the two concepts,
  - stemmed tokens between them,
  - verbs in 3-word window before and after each concept and between them,
  - Levin's class of the verbs (coming from VerbNet),
  - semantic type (from the UMLS) of tokens in a 3-word window to the left and the right of each candidate concepts,
  - preposition between concepts,
  - headword of concepts (headword is the token after preposition, else it's the last token).

Task 3. Relation annotation

System description (run R1, R2 and R3)

- **Syntactic features:**
  - part-of-speech in a 3-word window to the left and the right of the candidate concepts,
  - presence of a preposition,
  - presence of a coordination conjunction between concepts.
  - punctuation sign.

Task 3. Relation annotation

Preprocessing (run R1, R2 and R3)

- **Files are preprocessed and normalized**.
  - we replaced abbreviations by their meanings:
    - *h.o.* $\rightarrow$ *history of*
    - *p.r.n.* $\rightarrow$ *as needed*
  - we substituted the person's name (**NAME[VVV]), the date (**DATE[Jan 06 2008]), the person's age and other numbers respectively with <NAME>, <DATE>, <AGE> and <NUM>.
  - Finally files are POS tagged by the TreeTagger.

Task 3. Relation annotation

Preprocessing (run R2)

- **Concepts substitution:** concepts are substituted with their types (problem, test or treatment), and each sentence is duplicated for each candidate relation.
- **Syntactic simplification:** deletion of some syntactic phrases between the candidate concepts.
    - If the concept is at the beginning of the noun phrase, all words after the concept in the noun phrase are deleted.
    - If there is a PP, an ADJP, a CONJP, a WHNP or a CC (followed by a noun phrase) between the concepts, it is replaced with its POS tag (<PP>, <ADJP>, etc.).
- Texts are analyzed by the Charniak/McClosky self-training parser.

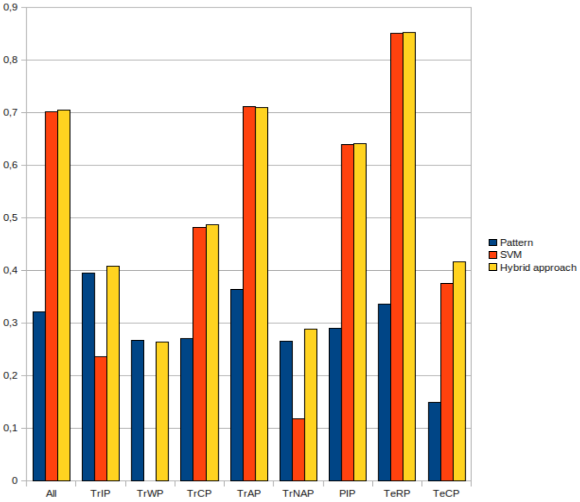
Task 3. Relation annotation

	Recall	Precision	F-measure
Run 1	0.634	0.797	0.706
Run 2	0.626	0.718	0.669
Run 3	0.708	0.711	0.709

Best run: #3

	Recall	Precision	F-measure
TrIP	0.414	0.458	0.435
TrWP	0.168	0.774	0.276
TrCP	0.435	0.550	0.486
TrAP	0.760	0.676	0.715
TrNAP	0.251	0.495	0.333
PIP	0.645	0.670	0.657
TeRP	0.881	0.813	0.846
TeCP	0.391	0.612	0.477

Task 3. Evaluation



1. Introduction
2. Task 1. Concept extraction
3. Task 2. Assertion annotation
4. Task 3. Relation annotation
5. Conclusion

Discussion

Task 1. Concept extraction

- **Our approach:**
  - to determine morpho-syntactic and semantic information for each token.
  - to let a state-of-the-art sequence classifier make concept type and boundary decisions.

  → good basis.

- **Further work:** to use syntactic chunking.

  → to determine more precisely the boundaries of concepts, especially for prepositional phrases.

Discussion

Task 2. Assertion annotation

- **Machine-learning based** system.
- We achieved better results over **well-represented classes** than small classes
  - present and absent totalize 89.7% of all assertions
  - vs. conditional only 0.9% of the assertions.
- The use of **trigger words** allows us to achieved good results over the other small class associated with someone else (0.8% of the assertions).

Discussion

Task 3. Relation annotation

- We obtained better results for the **affirmative classes** (TeRP, TrAP, and PIP) than negative ones (TrNAP, TrWP).
- The algorithm performed better training for relationship with **more examples**.

Thank you!