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

Cyril Grouin, MSc[1], Asma Ben Abacha, MSc[1,2], Delphine Bernhard, PhD[1], Bruno Cartoni, PhD[1], Louise Deléger, PhD[1], Brigitte Grau, PhD[1,3], Anne-Laure Ligozat, PhD[1,3], Anne-Lyse Minard, MSc[1,2], Sophie Rosset, PhD[1], Pierre Zweigenbaum, PhD[1,4]

[1]LIMSI–CNRS, Orsay, France [2]Université Paris-Sud 11, Orsay [3]ENSIIE, Evry [4]INaLCO, Paris

Abstract *This year's i2b2/VA challenge is dedicated to medical concept extraction as well as the annotation of assertions and relationships of concepts. Several kinds of concepts, assertions, and relations must be processed. In this paper, we present the methods we used, mainly based upon machine-learning systems. The results we obtained on the final ground truth (F-measures up to 0.773 for concepts, 0.931 for assertions, and 0.709 for relations) constitute a basis for further work.*

Introduction

The i2b2/VA 2010 challenge focuses on the extraction of medical concepts and the annotation of assertions made about medical problems and relationships between concepts. This kind of information allows to sum up the organization of the content of medical reports and is in line with the i2b2 2009 challenge which aimed to give an easy access to medication information within medical reports.

In this paper, we present the LIMSI participation in this challenge. After a short reminder of the challenge requirements and of the corpus, we describe the pipelines we developed for each task. We then conduct a short evaluation and discuss the results.

Challenge requirements

The fourth i2b2/VA challenge consists of three tracks: first, the extraction of three types of medical concepts (problems, tests, and treatments); secondly, the annotation of assertions made on medical problems; and finally, the annotation of relations between concepts.

Concepts are of three types and mainly correspond to a set of semantic types from the UMLS: (i) *problems* concern observations made about the patient if thought to be abnormal or caused by a disease, (ii) *treatments* describe all methods used to resolve a medical problem, and (iii) *tests* refer to examinations and procedures done about a medical problem.

Assertion annotations must be provided only for medical problems and consist of six categories: the patient experiences the medical problem (*present*), or does not (*absent*); the patient may have a problem that is uncertain (*possible*), or that occurs only under certain conditions (*conditional*); the patient may develop the problem (*hypothetical*), or the problem is mentioned in relation to someone else (*not associated with patient*).

Relation annotations must describe relationships between: (i) a problem and a treatment where the treatment can improve (*TrIP*), worsen (*TrWP*), cause (*TrCP*) the problem, where it can be administered (*TrAP*) or not (*TrNAP*) for the problem, (ii) a problem and a test where the test reveals (*TeRP*) or allows a physician to investigate (*TeCP*) the problem, and (iii) a problem that indicates another problem (*PIP*).

Corpus description

The corpus includes discharge summaries from three institutions (Partners HealthCare, Beth Israel Deaconess Medical Center, and University of Pittsburgh Medical Center) and progress notes only from UPMC. This year, ground truth annotations were provided to each participant in three dedicated files per report. The training corpus is composed of 349 reports while the ground truth corpus is made up of 477 documents. We split the training corpus into training (241 reports), development (54), and test (54) sub-corpora.

System description

Task 1. Concept extraction

Medical Natural Language Processing techniques are generally of two kinds: (i) machine-learning approaches that are more and more used because they provide a fast path to results, once corpora have been annotated; and (ii) expert-knowledge-based techniques that require much work while providing reliable results. Concept extraction has been studied by defining rules and gazetteers[1] or using linguistic resources obtained from the UMLS[2,3,4]. A few approaches mainly used the structure of the discharge summaries to extract test and treatment concepts[5]. We developed two pipelines for the concept extraction task: the first uses a machine-learning method, the second is mainly based upon MetaMap.

Machine-learning method. We defined the following pipeline for the first two runs we submitted. Each of them first performs limited linguistic analysis, whose output is represented as features which a machine-learning algorithm then uses to make decisions on concept boundaries and types. These features included: (i) n-grams of tokens, (ii) typographic clues (*letter case, alphabetic/date/digit/punctuation category*), and (iii) syntactic and semantic tags defined for each token as follows.

First, we performed morpho-syntactic analysis using Tree Tagger[6]; POS tags and lemmas are thus associated to each token. We then performed a semantic tagging using a specific lexicon of 62,263 adjectives and 320,013 nouns based on the UMLS Specialist Lexicon. 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*).

We extended the semantic tagging with 11 major semantic categories: 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, upper, right, left*), medication, mode of administration, medical object (*cannula, drain, pacemaker, stent*), procedure (*amputation, blood transfusion, dialysis*), and dosage. We created these categories thanks to lists from the UMLS[7], from Sager's work[8,9], and lists we had compiled for i2b2 2009 (medication names).

We built a model over the training corpus using CRF++[10], a machine-learning tool based upon conditional random fields. We applied this model over the test corpus. This pipeline was used for our first submission (run C1).

We tried to refine the output of this model by designing a few post-editing rules. 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).

MetaMap based method. MetaMap[11] is a tool conceived by the NLM to locate medical terms and their corresponding concepts and semantic types from the UMLS metathesaurus and semantic network. However it has some residual problems at the noun-phrase segmentation level and at the recognition of several known drugs, diseases and tests. We enhance MetaMap's output by performing two steps before the execution of MetaMap (run C3): (i) segmentation into noun-phrases with treetagger-chunker and (ii) search of the located terms in pre-compiled lists of medical problems, tests and treatments. We finally filter the obtained results using lists of common errors and stopwords.

Task 2. Assertion annotation

Two approaches to assertion classification are usually distinguished: (i) expert-knowledge-based approaches which consist in listing and detecting indicative phrases or specific syntactic dependencies for a given type of assertion[12,13,14,15,16] and (ii) machine-learning approaches which rely on annotated data to train a supervised classification system[13,16]. Assertion classification is also closely related to hedge classification, which aims at detecting speculation in natural language texts. This task has been addressed with weakly supervised machine-learning[17].

We developed two systems for assertion annotation, the first one using machine-learning techniques, the second one using manually-designed rules.

For the first system, we considered assertion identification as a classification task, with the six assertion types as target classes (run A1). We trained an SVM with the libsvm tool[18] based on binary feature vectors. We automatically selected the optimal parameter values using cross-validation[⋆]. We focused on three types of features: contextual lexical features, trigger-based features and target concept internal features:

- Contextual lexical features consist of token and stemmed token unigrams in a 5-word window to the left and to the right of the target concept. We also experimented with POS unigrams, as well as token bigrams and trigrams, but these did not lead to significant improvements.
- Triggers consist of phrases which are indicative of a given assertion class. These indicative phrases are used, e.g. in NegEx[12] and GenConText. We used the triggers collected for our exten-

[⋆] This step was performed with the easy.py script provided with libsvm.

sion of GenConText, with few additions. These triggers were identified before and after the problem concept, again in a 5-word window. We also identified some concept-internal triggers such as "on exertion" which is indicative of the conditional assertion class when it occurs within an annotated concept.

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

Our study of the development data showed that many problem concepts are coordinated with commas or coordinating conjunctions, e.g. "pleural effusion or pneumothorax". These sequences of coordinated problems 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. The important role of coordination has been highlighted before for event extraction[15]. We therefore pre-processed 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 and right windows start at the end of the sequence. These contexts are shared by all concepts occurring in the same coordinated sequence. We also kept specific features encoding 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", the concepts "pneumothorax", "pleural" and "effusion", as well as the stem "effus" are used as features.

The second system (run A2) was based on an extension of the NegEx[12] algorithm, which 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. Then, we extended the General ConText Java implementation[†] to deal with the categories *conditional*, *hypothetical* and *not associated with the patient*.

Task 3. Relation annotation

We considered relation identification as a classification task, with the 8 relation types (TrIP, TrWP, TrCP, TrAP, TrNAP, TeRP, TeCP and PIP). We used a hybrid approach which combines machine-learning techniques and linguisitic-pattern matching. We trained an SVM with the libsvm tool and constructed linguistic patterns manually. After empirical observations we kept only the patterns of four relations types as the others did not offer satisfying results.

The advantages of such a hybrid approach lie in the fact that some relation types do not have enough annotations to feed the automatic classifiers. In such cases (e.g. The TrWP relation), linguistic patterns provide a substantial enhancement of the obtained results. A second advantage is that linguistic patterns may confirm automatically-induced relations which helps adding confidence to the obtained results.

In our experiments we performed three runs based on supervised learning. For the first one (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. The second (run R2) is supervised learning from simplified texts. Finally, the last one (run R3) is a combination of the first two results. The features of our SVMs are as follows:

- Surface features: order of the candidate concepts, distance between them (i.e. the number of tokens), presence of other concepts, semantic type (from the UMLS) of tokens in a 3-word window to the left and the right of each candidate concepts, type of the concepts (problem, test or treatment) and normalized title of the section.
- Lexical 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[19]), preposition between concepts, headword of concepts. headword is the token after preposition, else it's the last token).
- 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.

Files are preprocessed and normalized. First, we replaced abbreviations by their meanings, for example *h.o.* are converted into *history of* and *p.r.n.* into *as needed*. Then we substituted the person's name (or e.g. **NAME[VVV]), the date (or e.g. **DATE[Jan

[†] http://code.google.com/p/negex/

06 2008]), the person's age and other numbers respectively with <NAME>, <DATE>, <AGE> and <NUM>. Finally files are POS tagged by the TreeTagger.

Syntactic simplification. For the second run, preprocessing of the text consists of a syntactic simplification, which involves deletion of some syntactic phrases between the candidate concepts. The aim of the simplification is to delete useless information for the relation identification process, rather than to obtain grammatically correct sentences. Before the simplification process, concepts are substituted with their types (problem, test or treatment), and each sentence is duplicated for each candidate relation (if there are 3 concepts in a sentence, it is written 3 times). Then texts are analyzed by the Charniak/McClosky self-training parser[20]. The simplification proceeds in two steps. First, if the concept is at the beginning of the noun phrase, all words after the concept in the noun phrase are deleted. Second, if there is a prepositional phrase (PP), an adjectival phrase (ADJP), a phrase with a conjunction (CONJP), a relative pronoun (WHNP) or a coordination conjunction (followed by a noun phrase) between the concepts, it is replaced with its POS tag (<PP>, <ADJP>, etc.).

Evaluation and discussion

Evaluation

We evaluated each task using the ground truth corpora.

Concept extraction. The ground truth corpus is composed of 45,009 concepts to be extracted (18,550 problems, 13,560 treatments, and 12,899 tests).

	Run C1	Run C2	Run C3
All concepts	**0.772**	**0.773**	**0.454**
Problem	0.767	0.769	0.452
Treatment	0.779	0.778	0.509
Test	0.770	0.771	0.363

Table 1: Concepts: F-measure (class exact span).

Run C1 is machine-learning based while run C2 applies correcting rules to the previous output. The low results in run C3 are mainly due to an issue in offset computation in some entries which was not detected before submission.

Assertion annotation. The ground truth corpus is composed of 18,550 assertions on medical problems (13,025 present, 3,609 absent, 883 possible, 717 hypothetical, 171 conditional, and 145 associated with someone else).

	Run A1	Run A2
All assertions	**0.931**	**0.882**
Present	0.956	0.924
Absent	0.939	0.849
Possible	0.622	0.547
Hypothetical	0.876	0.784
Conditional	0.363	0.327
Associated with someone else	0.816	0.704

Table 2: Assertions: F-measure (exact span with matching assertion).

Relation annotation. The ground truth corpus is composed of 9,069 relationships (198 TrIP, 143 TrWP, 444 TrCP, 2,486 TrAP, 191 TrNAP, 1,986 PIP, 3,033 TeRP, and 588 TeCP).

	Run R1	Run R2	Run R3
All relations	**0.706**	**0.669**	**0.709**
TrIP	0.409	0.367	0.435
TrWP	0.264	0.041	0.276
TrCP	0.487	0.406	0.486
TrAP	0.710	0.668	0.715
TrNAP	0.289	0.243	0.333
PIP	0.641	0.612	0.657
TeRP	0.853	0.820	0.846
TeCP	0.417	0.430	0.477

Table 3: Relations: F-measure (exact span).

The three runs are machine-learning based. Run R1 uses rules for four relations, run R2 applies simplification to input sentences, and run R3 is a union of the results of run R1 and run R2.

Discussion

Our approach to concept extraction aimed to determine morpho-syntactic and semantic information for each token and let a state-of-the-art sequence classifier make concept type and boundary decisions. This allowed us to obtain a good basis. Nevertheless, we did not use syntactic chunking which would have determined more precisely the boundaries of concepts, especially for prepositional phrases.

In assertion annotation, since our system is machine-learning based, we achieved better results over well-represented classes (such as "present" and "absent" that totalize 89.7% of all assertions) than small classes such as "conditional" (only 0.9% of the assertions). Nevertheless, we achieved good results over the other small class "associated with someone else" (0.8% of the assertions) due to the use of trigger words.

In relationship 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.

Conclusion

Using machine-learning methods, we achieved F-measures of 0.773 in concept extraction, 0.931 in assertion annotation, and 0.709 in relation annotation. For each task, we obtained better results over well-represented classes, machine-learning approaches being highly representativeness-dependent. Using rules to refine these outputs is a challenging task we started in concept extraction with promising results[‡].

References

1. Mykowiecka A, Marciniak M, and Kupść A. Rule-based information extraction from patients' clinical data. *J Biomed Inform* Oct 2009;42(5):923–36.
2. Friedman C, Shagina L, Lussier Y, and Hripcsak G. Automated encoding of clinical documents based on natural language processing. *J Am Med Inform Assoc* 2004;11(5):392–402.
3. Li Q and Wu YFB. Identifying important concepts from medical documents. *J Biomed Inform* Dec 2006;39(6):668–79.
4. Denecke K. Semantic Structuring of and Information Extraction from Medical Documents Using the UMLS. *Methods Inf Med* 2008;47(5):425–34.
5. Long W. Lessons Extracting Diseases from Discharge Summaries. In: AMIA Annu Symp Proc, 2007:478–82.
6. Schmid H. Probabilistic Part-of-Speech Tagging Using Decision Trees. In: Proceedings of the International Conference on New Methods in Language Processing, 1994:44–9.
7. Lindberg DA, Humphreys BL, and McRay AT. The Unified Medical Language System. *Meth Inform Med* 1993;32(4):281–91.
8. Sager N, Lyman M, Nhàn NT, and Tick LJ. Medical language processing: applications to patient data representation and automatic encoding. *Meth Inform Med* 1995;34(1–2):140–6.
9. Sager N and Nhàn NT. The computability of strings, transformations, and sublanguage. In: Nevin BE and Johnson SM, eds, *The legacy of Zellig Harris – Language and information into the 21st century - volume 2: computability of language and computer applications*, (vol2). John Benjamins Publishing Company, Amsterdam/Philadelphia, 2002:79–120.
10. Kudo T. CRF++: Yet Another CRF toolkit. Software available at http://chasen.org/taku/software/crf++/.
11. Aronson A. Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. In: Proc AMIA Symp, 2001:17–21.
12. Chapman WW, Bridewell W, Hanbury P, Cooper GF, and Buchanan BG. A Simple Algorithm for Identifying Negated Findings and Diseases in Discharge Summaries. *J Biomed Inform* 2001;2001:301–10.
13. Light M, Qiu XY, and Srinivasan P. The Language of Bioscience: Facts, Speculations, and Statements In Between. In: Hirschman L and Pustejovsky J, eds, HLT-NAACL 2004 Workshop: BioLINK 2004, Linking Biological Literature, Ontologies and Databases, Boston, Massachusetts, USA. Association for Computational Linguistics, May 2004:17–24.
14. Chapman W, Dowling J, and Chu D. ConText: An Algorithm for Identifying Contextual Features from Clinical Text. In: Biological, translational, and clinical language processing, Prague, Czech Republic. Association for Computational Linguistics, June 2007:81–8.
15. Kilicoglu H and Bergler S. Syntactic dependency based heuristics for biological event extraction. In: BioNLP'09: Proceedings of the Workshop on BioNLP, Morristown, NJ, USA. Association for Computational Linguistics, 2009:119–27.
16. Uzuner O, Zhang X, and Sibanda T. Machine Learning and Rule-based Approaches to Assertion Classification. *J Am Med Inform Assoc* 2009;16(1):109–15.
17. Szarvas G. Hedge Classification in Biomedical Texts with a Weakly Supervised Selection of Keywords. In: Proc ACL-08: HLT, Columbus, Ohio. Association for Computational Linguistics, June 2008:281–89.
18. Chang CC and Lin CJ. LIBSVM: a library for support vector machines, 2001. Software available at http://www.csie.ntu.edu.tw/˜cjlin/libsvm.
19. Kipper K, Korhonen A, Ryant N, and Palmer M. A large-scale classification of english verbs. *Language Resources and Evaluation Journal* 2008;42(1):21–40.
20. McClosky D and Charniak E. Self-training for biomedical parsing. In: Proceedings of the Association for Computational Linguistics (ACL 2008, short papers), Columbus, Ohio. 2008.

[‡] This work was partially funded by the Akenaton project under grant number ANR-07-TecSan-001, and partly realized as part of the Quæro Programme, funded by OSEO, French State agency for innovation. The 2010 i2b2/VA challenge and the workshop are funded in part by the grant number U54-LM008748 on Informatics for Integrating Biology to the Bedside from National Library of Medicine. This challenge and workshop are also supported by resources and facilities of the VA Salt Lake City Health Care System with funding support from the Consortium for Healthcare Informatics Research (CHIR), VA HSR HIR 08-374 and the VA Informatics and Computing Infrastructure (VINCI), VA HSR HIR 08-204. MedQuist, the largest transcription technology and services vendor, generously co-sponsored the 2010 i2b2/VA challenge meeting at AMIA.