UCM at CLEF eHealth 2013 Shared Task1

Lucía Hervás[1], Víctor Martínez[1], Irene Sánchez[1], and Alberto Díaz[1]

NIL Group
University Complutense of Madrid
C/Profesor García Santesmases, Madrid, 28040, Spain,
lhervasmartin@gmail.com
victormartinezsimon@gmail.com
irene.sanchzmartinz@gmail.com
albertodiaz@fdi.ucm.es,
http://nil.fdi.ucm.es/

Abstract. We are developing a system that analyze medical reports and extract a SNOMED-CT based concept representation. The more interesting characteristic of our system is not only that it can detect the concepts. It also takes into account if they appear in an affirmative, negative or speculative context. The system also separates the concept representation according to the structure of the document.
Our system takes these steps: automatic orthographic correction, acronyms and abbreviation detection, negation and speculation phrase detection and medical concepts detection.
For participating in Task 1 we have adapted our system in order to obtain the mentions that belong to the Disorders UMLS semantic group. The approach is based on using MetaMap to detect the concepts and the spans. Our aim was to identify what was the best way to use MetaMap in our system to solve the Task 1.

Keywords: Natural Language Processing, medical report, spellcheck, acronym expansion, negated phrase, speculated phrase, concept detection, Metamap, UMLS

1 Introduction

With the advent of computers and especially the Internet, the information explosion produced has grown by leaps and bounds.We are in a time where progress allow us to generate, store and distribute lots of information. But all this advances have disadvantages too.

By this moment lots of information are generated, and differentiate the correct information from the wrong information requires a big amount of money and time.

With the aim of reduce this two problems, the computer engineers, the software developers and the medicine experts, are developing different strategies to extract information from different places and increase the medical knowledge, to improve researches and in the last term, to have a better live.

With this aim in mind, we decide to develop a tool to help medical experts in their job.

During our research, we discover the CLEF tasks [12] for biomedical documents which is very close of what we are developing so we decide to participate to increase our knowing and the performance of our system.

We develop software to solve the first task that consist in discover where are the different medical concepts in a medical report and catalogue them, using for this the UMLS CUIs.

We are developing a system that analyze medical records and extract a SNOMED-CT based concept representation. Before the analysis we have other phases: a language corrector and an acronyms expander. The more interesting characteristic of our system is that not only detect the concepts, it also take into account if they appear in an affirmative, negative or speculative context. The system also separates the concept representation according to the structure of the document, that is, there is a different representation for each section of the document.

For participating in Task 1 we have adapted our system in order to obtain the mentions that belong to the Disorders UMLS semantic group. The approach is based on using MetaMap to detect the concepts and the spans. Our aim was to identify what was the best way to use MetaMap in our system to solve the Task1.

We have submitted runs with no external annotations, two for task 1a and two for task 1b. The difference between the runs is only the DB used. We used the 2012AA USAbase strict model for the first run and the 2011AA USAbase strict model for the second run. Our best results for task 1a show 0.504 F1 score with strict evaluation, and 0.660 F1 score with relaxed evaluation. Our best results for task 1b show 0.362 Accuracy with strict evaluation and 0.870 Accuracy with relaxed evaluation.

2 State of the art

The aim of this part of the text it's explain the different tools and approximations that actually exists in the world of natural language processing and which of them we use in our project and are used in the software developed for Clef task.

The different tools analyzed were:

- Ontologies
- MetaMap
- Concept detection and disambiguation
- Orthographic correction
- Negation detection
- Speculation detection

2.1 Ontologies

One definition of what it is an ontology is this: "An ontology defines the basic terms and relations comprising the vocabulary of a subject area, as well as the

rules for combining these terms and relations define extensions to the vocabulary"[7].

In medical and biomedical reports, there are two important ontologies: SNOMED-CT and UMLS

- SNOMED-CT:
  SNOMED-CT or Systematized Nomenclature of Medicine Clinical Terms is a big ontology of medical concepts, written in different idioms and by this moment it's used in more than fifty different countries. Since 2007, the SNOMED-CT license belongs to the International Health Terminology Standards Development Organization (IHTSDO) which distributed and maintains the Ontology.
  SNOMED-CT it's compound by this elements:
  - Concepts: They represent different medical terms. All of them, have a unique identifier.
  - Descriptors: A concept could have more than one different descriptor, which are synonyms between them.
  - Relationships: These are use to connect different concepts because they have any kind of relationship.
- UMLS:
  UMLS or Unified Medical Language System is a file and software set which have a big amount of biomedical terms. It was developed by the National Library of Medicine (NLM). It's compound by 4 kind of elements.
  - Metathesaurus:
    It's a big database with lots of biomedical concepts. It takes as sources some databases, for example: MeSH, RxNorm and SNOMED-CT.
    All elements in the Metathesaurus gets an identification number to represent them.
  - Semantic type:
    The semantic type it's used to classify the biomedical vocabulary and it also contains the different relationships between all the concepts.
  - Specialized lexicon.
    A specialized lexicon its incorporated into MetaMap distribution, which have more than 200.000 different terms and common English words. This terms are use to improve the different lexical tools included in the distribution.
  - Different lexical tools like:
    * A Lexical Variant Generator (LVG).
    * A word index generator (WordInd).
    * A normalized strings generator (Norm)

2.2 MetaMap

The MetaMap program has been used extensively for a wide array of BioNLP studies, such as automatic indexing of biomedical literature and concept based text summarization.

MetaMap finds Metathesaurus concepts by performing a shallow syntactic analysis of the input text, producing a set of noun phrases. The noun phrases are then used to generate sets of variants which are consequently looked up from the Metathesaurus concepts. Matching concepts are evaluated against the original text and the strength of the mappings are calculated. The candidates are finally combined and the final scores are computed, where the highest score of a complete mapping represents MetaMap's interpretation of the text.

The MetaMap program is provided to be run both locally and remotely. We ran the current version of MetaMap, MetaMap2012 remotely via the batch mode facility.

The parameter set that influences the performance of MetaMap included; using a Relaxed model, selecting the NLM2102AB Metathesaurus version, including all derivational variants, enabling unique acronym/abbreviation variants only, allowing large N, preferring multiple concepts and using word sense disambiguation.

Relaxed Model is a model provided by MetaMap in addition to the Strict model. MetaMap offers the Strict model as a default setting in which all types of filterings are applied, however, we applied the Relaxed model in which only Manual and Lexical filterings are used. While the Strict model is most appropriate for experiments that require the highest accuracy, it only covers only 53% of the Metathesaurus strings. As we consider high coverage of concepts an important factor, we thus applied the relaxed model which consists of up to 83% of Metathesaurus strings.

The versions of Metathesaurus, Base, USAbase and NLM, provided with MetaMap are different in their Metathesaurus coverage and the license type required for using vocabulary sources. The NLM version which is offered at no cost for research purposes and covers all of the provided Metathesaurus was used in our work.

Variants, such as inflectional and derivational variants, are computed by MetaMap to account for the textual variation in the text. With this setting, many types of variants are generated recursively, and only acronyms and abbreviations are restricted to the unique ones. In addition, the candidates also include words that can be prepositions, conjunctions or determiners if they occur often enough in Metathesaurus.

Prefer multiple concepts causes MetaMap to score the mappings with more concepts higher than those with fewer concepts. This option is useful for discovering higher-order relationships among concepts found in the text and as such is assumed to be helpful.

Word sense disambiguation attempts to solve lexical ambiguities by identifying the correct meaning of a word based on its context. By using this option in MetaMap, the program attempts to solve the ambiguities among equally scoring concepts by choosing the concept(s) based on semantic type.

2.3 Concept detection and disambiguation.

When we are detecting a concept, we can't only search any word in the text into a database, because is high probably that this same concept has more than one entrance into the database. To do our work, we start searching different ways to detect and disambiguate concept, and we decide to work with MetaMap software.

As we have just said, MetaMap, includes concept disambiguation, with a very good result ratio.

They use an algorithm written on C++, that after a few time, it was written into Java and Perl. This algorithm is based in the Hidden Markov Model and was modify to include contextual and grammatical information to improve the system accuracy.

The developers also using a training set, learn which words can go with another words based on their grammatical category [11].

With all these improvements, they made an algorithm that with a 1000 different phrases, they have a 97,43% precision with 582 phrases correctly marked and 261 phrases with only one error.

2.4 Orthographic correction.

When a doctor it's writing a medical report, he or she may have a misspelling because he or she is writing very fast and taking notes. A little orthographic error can change the meaning of all the phrase so good written reports are recommended.

We can say that an orthographic corrector it's an algorithm that checks every word into a database and says if this word is correct or if it's incorrect and in this case, says some corrections ordered by a score. The best of all these algorithm is the one who says the better suggestions. To increase the suggestions ratio, they use some different methods like:

- Distance between words
  With this kind of improvement, we check how far are some words from another words and the closest words are probably the correct one. There are some different ways to develop this issue. Here are some of them:
  - Differences between characters
  - Hamming distance [4]
  - Levenshtein distance [6]
  - Damerau-Levenshtein distance [2]
  - Keyboard distance as Manhattan distance using Needleman-Wunsch algorithm [5] [8]
- Phonetic algorithms
  All of this improvements check how close are different phonetic sounds between all the phonetic sounds from the two words. The different algorithms for this issue are:
  - Soundex [10]
  - New York State Identification and Intelligence System (NYSIIIS) [13]
  - Metaphone [9]

2.5 Negation detection.

When we talk and write, we don't say every phrases affirmed, we use negative phrase to emphasize different things. On medical reports, negated phrases are very important to detect, because they say some concepts that the patient doesn't have and if we misinterpret this phrases and concepts, we will go in the wrong direction.

We use 2 approximations to solve this problem. The first was use the NegEx algorithm as is distributed, and the second option was use the MetaMap NegEx algorithm implementation.

NegEx algorithm The NegEx algorithm [1] uses regular expressions to detect when a phrase is negated and say witch word of all causes this detection.

The algorithm uses as regular expressions, four kind of concepts:

- Negative words: these are the words that negate all the phrase that appear right after them.
- PostNegative words: these are the words that negate the phrase right before them.
- Conjunctions: these are the words that connect different negative phrases.
- Pseudo Negations: these are the words that we may think that make a phrase negative, but the really don't do that, so when we detect this words, we skip them.

NegEx on MetaMap MetaMap include the NegEx algorithm, but in the year 2009, the MetaMap developers improve the algorithm[1].

The improvements where this:

- Add new words to detect negations like: *other than, otherwise, to account for, to explain* and *then.*
- Add to MetaMap databases, some negated concepts. If any of this concepts appear in the text, then the phrase is negated.
- Join different negated concepts into one, to reduce noise.

2.6 Speculation detection.

The speculation detection it's a very important task in biomedical report. Sometimes, specialist aren't really sure about what are they detecting, because some different illness have the same symptoms.

To resolve this issue, after a web search, we learn how to adapt the NegEx algorithm to detect the speculation phrases into a medical report [3].

[1] http://metamap.nlm.nih.gov/MM09_Release_Notes.shtml

3 Framework evaluation

Participants will be provided training and test datasets. The evaluation for all tasks will be conducted using the withheld test data. Participating teams are asked to stop development as soon as they download the test data. Teams are allowed to use any outside resources in their algorithms. However, system output for systems that use annotations outside of those provided for Tasks 1 will be evaluated separately from system output generated without additional annotations.

3.1 Evaluation Measures

Task 1 - Named entity recognition and normalization of disorders

A. Boundary detection of disorders: identify the span of all named entities that could be classified by the UMLS semantic group Disorder (excluding the semantic type Findings)

Evaluation measure: F1-score

- F1-score = (2 * Recall * Precision) / (Recall + Precision)
- Recall = TP / (TP + FN)
- Precision = TP / (TP + FP)
- TP = same span
- FP = spurious span
- FN = missing span

Exact F1-score: span is identical to the reference standard span

Overlapping F1-score: span overlaps reference standard span

B. Named entity recognition and normalization of disorders: identify the boundaries of disorders and map them to a SNOMED-CT code.

Evaluation measure: Accuracy

- Accuracy = Correct/Total
- Correct = Number of disorder named entities with strictly correct span and correctly generated code
- Total = Number of disorder named entities, depending on strict or relaxed setting:
  - Strict: Total = Total number of reference standard named entities. In this case, the system is penalized for incorrect code assignment for annotations that were not detected by the system.
  - Relaxed: Total = Total number of named entities with strictly correct span generated by the system. In this case, the system is only evaluated on annotations that were detected by the system.

4 Processing

As we have just say, we are developing another system that extract affirmative, negative and speculative concepts from medical reports and when we decide participate on Clef tasks, we made a few changes in our software to satisfy the requisites. Here we will explain what our system does and which changes we made to adjust to Clef tasks.

4.1 The software tool

The first action that is done in our system it's transform the medical report to a XML file, so we can use it easily. To do this little task, we use different and configurable regular expressions, where we define different "Beginning words" witch means the beginning of any different sections in our report. For example, we have this sections: "Allergies", "Family history". When a "Beginning word" is detected, we assume that in this moment, a new sections begins and it continues until the end of the report or another "Beginning Word".

After that, we made a orthographic correction using the Hunspell's algorithms and dictionaries. We only made a correction if there is a lot of score in the suggestion, because if we correct all the things that Hunspell suggest, we'll probably introduce some noise that in future actions will be very dangerous.

The next step that we do, it's detect acronyms and abbreviation, using for that 3 kinds of detectors:

1. A Database that we have developed, witch contains some different acronyms and abbreviation and a few rules to make a little disambiguation.
2. The MetaMap acronym and abbreviation detector.
3. Using MetaMap with an UDA file.

The next step in our system it's the negation detection. To do this issue first we think in the NegEx algorithm but MetaMap includes a modified NegEx algorithm so we use this last option.

After this last step, we have a speculation detector which works very easily. We have a NegEx algorithm modified to detect some speculative words as regular expressions.

Finally, we have the concept detector.

To do this finally task, we use MetaMap to detect the different concepts and to know their CUI. MetaMap retrieves some concepts, so to reduce the noise, we only take the concepts with the most score of all of them. We also use the MedPost/SKR server included in MetaMap to have a disambiguation between concepts. Another noise reductive technique that we develop, was a custom dataset of concepts using to create it the MetamorphoSys tool and the DataFileBuilder developed by the MetaMap group.

After all this job, we save the results in a XML file to use them in the future for different things, like use Lucene to search between different medical reports.

4.2 Modifications for CLEF task

We made a several modifications to make our system complete the Clef tasks.

First we disable the language corrector, because we assume that all the reports will be right written.

We also disable the acronym and abbreviation detector because this is the task 2 target.

We disable the negation and speculative phrase detector because this will be useful in task 3, but to detect concepts, it only make the system run slowly.

Finally, we make the system to accept only the next semantic types:

- Congenital Abnormality
- Acquired Abnormality
- Injury or Poisoning
- Pathologic Function
- Disease or Syndrome
- Mental or Behavioral Dysfunction
- Cell or Molecular Dysfunction
- Experimental Model of Disease
- Anatomical Abnormality
- Neoplastic Process
- Signs and Symptoms

5 Results

For participating in Task 1 we have adapted our system in order to obtain the mentions that belong to the UMLS semantic group Disorders. The approach is based on using MetaMap to detect the concepts and the spans. We have submitted runs with no external annotations, two for task 1a and two for task 1b. The difference between the runs is only the DB used. We used the 2012AA USAbase strict model for the first run and the 2011AA USAbase strict model for the second run. Our best results for task 1a shown a 0.504 F1 score with strict evaluation, and a 0.660 F1 score with relaxed evaluation. In strict evaluation 18th and 20th of 27 runs. In relaxed evaluation, 21th and 22th of 28 runs. Explain the differences between strict and relaxed evaluation. Explain the differences between using 2012AA or 2011AA.

Table 1. Task 1A. No external annotations. Strict

Team,Country	Precision	Recall	F1-Score
UTHealth_CCB.2, UT, USA	0.800	0.706	0.750
NIL-UCM.2, Spain	0.617	0.426	0.504
NIL-UCM.1, Spain	0.621	0.416	0.498
FAYOLA.1, VW, USA	0.024	0.446	0.046

Table 2. Task 1A. No external annotations. Relaxed

Team,Country	Precision	Recall	F1-Score
UTHealth_CCB.2, UT, USA	0.925	0.827	0.873
NIL-UCM.2, Spain	0.809	0.558	0.660
NIL-UCM.1, Spain	0.812	0.543	0.651
FAYOLA.1, VW, USA	0.504	0.043	0.079

Table 3. Task 1B. No external annotations. Strict

Team,Country	Accuracy(sn2012)	Accuracy(sn2011)
NCBI.2, MD, USA	0.589	0.584
NIL-UCM.2, Spain	0.362	0.362
NIL-UCM.1, Spain	0.362	0.362
NCBI.2, MD, USA	0.006	0.006

Table 4. Task 1B. No external annotations. Relaxed

Team,Country	Accuracy(sn2012)	Accuracy(sn2011)
AEHRC.1, QLD, Australia	0.939	0.939
NIL-UCM.1, Spain	0.871	0.870
NIL-UCM.2, Spain	0.850	0.850
UTHealth_CCB.1, UT, USA	0.728	0.772

6 Errors and Analysis

With respect to Task 1a, our system have a lot of false positive, because the only word sense disambiguation that we use, it's with the MedPost/SKR server included in MetaMap distribution, and better with the databases of 2011.

With respect to Task 1b, we have a very good ratio detection of CUIs. MetaMap returns a lot of possible candidates, but we decide to save only the one with bigger score and in case of there are more than one with the same score, the first to appear in the list. We fail in this task for two reasons:

1. We don't take a look on the other results that MetaMap retrieve, so maybe the good one it's the second with most score.
2. We have the same error than in task 1a, we don't have a good word sense disambiguation, so we have lots of false positive detections.

Acknowledgements

We want to acknowledge the support given by the Shared Annotated Resources (ShARe) project, funded by the United States National Institutes of Health with grant number R01GM090187.

References

1. Chapman, W. W., Bridewell, W., Hanbury, P., Cooper, G. F., and Buchanan, B. G.: A simple algorithm for identifying negated findings and diseases in discharge summaries. Journal of biomedical informatics, vol. 34, no. 5, pp. 301-310, (2001)
2. Damerau, F. J.: A technique for computer detection and correction of spelling errors. Communications of the ACM, vol. 7, no. 3, pp. 171-176, (1964)
3. Farkas, R., Vincze, V., Mra, G., Csirik, J., and Szarvas, G.: The CoNLL-2010 shared task: learning to detect hedges and their scope in natural language text. in Proceedings of the Fourteenth Conference on Computational Natural Language Learning-Shared Task. Association for Computational Linguistics, (2010), pp. 1-12
4. Hamming, R. W.: Error detecting and error correcting codes. Bell System technical journal, vol. 29, no. 2, pp. 147-160, (1950)
5. Krause, E. F.: Taxicab geometry: An adventure in non-Euclidean geometry. Courier Dover Publications, (1987)
6. Levenshtein, V. I.: Binary codes capable of correcting deletions, insertions and reversals. in Soviet physics doklady, vol. 10, (1966), p. 707
7. Neches, R., Fikes, R., Finin, T., Gruber, T., Patil, R., Senator, T., and Swartout, W.: Enabling technology for knowledge sharing. AI magazine, vol. 12, no. 3, p. 36, (1991)
8. Needleman, S. B. and Wunsch, C. D.: A general method applicable to the search for similarities in the amino acid sequence of two proteins. Journal of molecular biology, vol. 48, no. 3, pp. 443-453, (1970)
9. Philips, L.: Hanging on the metaphone. Computer Language, vol. 7, no. 12 (December), (1990)
10. Russell, R. and Odell, M.: Soundex. US Patent, vol. 1, (1918)

11. Smith, L., Rindflesch, T., and Wilbur, W. J.: MedPost: a part of speech tagger for biomedical text. Bioinformatics, (2004)
12. Suominen, H., Salantera, S., Velupillai, S.: Three Shared Tasks on Clinical Natural Language Processing. Proceedings of CLEF 2013. To appear
13. Taft, R. L.: Name search techniques. Bureau of Systems Development, New York State Identification and Intelligence System, (1970), no. 1