NECLA at the Medical Natural Language Processing Pilot Task (MedNLP)

Pierre-François Laquerre
Department of Machine Learning
NEC Laboratories, Princeton, New Jersey USA
pierre.francois@nec-labs.com

Christopher Malon
Department of Machine Learning
NEC Laboratories, Princeton, New Jersey USA
malon@nec-labs.com

ABSTRACT

This paper gives an overview of NECLA's submitted systems for the De-Identification and Complaint & Diagnosis subtasks of the Medical Natural Language Processing Pilot Task (MedNLP)[5]. Our systems combine features derived from Part of Speech (POS) tags, a domain-specific dictionary, the Unified Medical Language System (UMLS) metathesaurus and semantic network, and a small set of heuristics based on trigger-words and polarity propagation through sentence dependency parse trees.

Team Name

NECLA

Subtasks

De-Identification
Complaint & Diagnosis

Keywords

natural language processing, clinical records

1. APPROACH

Each sentence was tokenized using MeCab[3] with the ipadic dictionary, augmented with a small domain-specific user dictionary to ensure proper segmentation of medical terms. Tokens were post-processed in order to group digits separated by dots as single floating point numbers instead of three distinct tokens. Finally, all numeric tokens were normalized into one of FLOAT, NUMBER_MONTH, NUMBER_DAY, NUMBER_YEAR or NUMBER based on their range. This rough grouping cut down the dictionary size by slightly over 800 tokens, and prevented the classifier from overfitting to specific numbers on age and time mentions.

These normalized tokens formed the basis of the sequences on which we trained a Conditional Random Field (CRF)[4][2] to predict the IOBES annotations specific to each task. MeCab's primary POS (e.g. noun, particle) and secondary POS (e.g. appellative vs. proper noun) tags were also included as features.

1.1 Domain-specific features

Given the domain-specific nature of this task — especially the second subtask — we derived features based on two knowledge bases from the medical domain.

The Life Science Dictionary (LSD)[7] is a collection of 31,382 Japanese words frequently used in MEDLINE. Using a greedy longest-match algorithm, tokens were tagged as either outside the dictionary (O) or inside (B-LSD and I-LSD, following the IOB2 annotation standard).

The second source of domain-specific information is the UMLS metathesaurus[6]. UMLS is an extensive compilation of controlled biomedical science vocabularies. The metathesaurus is comprised of Concept Unique Identifiers (CUIs), which uniquely represent a meaning. Each CUI is mapped to various string representations in the vocabularies included in UMLS, such as ICD-10, MeSH, SNOMED CT and many others. Other languages are included as well. A Japanese translation of Medical Subject Headings (MeSH) and Medical Dictionary for Regulatory Activities Terminology (MedDRA) is included as a source, totalling 179881 distinct strings that map to 58528 distinct concepts. On top of these concepts is a semantic network which provides a broad categorization, as well as relationship information such as is-a, broader-than, etc.

UMLS also provides a sophisticated concept detector MetaMap[1] which maps text to CUIs. This tool is powerful, but is specific to the English language. Therefore, we segmented each sentence into unilingual segments of either Japanese or English. English segments were fed through MetaMap, whereas Japanese segments were matched against the strings in UMLS using the same greedy algorithm as for LSD. From this, we derived a membership feature (O, B-UMLS, I-UMLS). We also exploited UMLS's semantic network to group concepts into one of 11 high-level categories (body part, disease, medication, etc.) that generalize UMLS's 133 semantic types.

1.2 Heuristic features

Because the contest provided very little training data for some of the mention classes, we found it useful to design extra features based on our prior knowledge. For the de-identification tasks, we could enumerate a reasonable set of known trigger words. Gender identification consisted of a strict check of the words "male" and "female", which covered all four examples in the training set with perfect precision. One or more numbers followed by a time suffix ("month", "year", "hour", *etc.*) triggered a feature indicating a possible time. Additionally, prefixes ("the same", "now", "the present", "previous", "next") plus a time suffix triggered the feature. When we found a time expression, we also assigned the feature to any adjacent suffix "from," "until," or "around".

De-identification of hospitals and locations was a bit more customized to patterns in the training set. Complete hospital expressions ("this institute," "a nearby doctor", "the same institute”) triggered a hospital feature by themselves. If the suffixes “hospital” or “clinic” occurred, we would assign the hospital feature and push the feature back to preceding words that were probably part of the title, such as “university,” “general,” or “of X.” This heuristic did not always cover the entire span of the hospital expression, but it provided an indicator that the CRF could learn to push back to preceding words when necessary.

Items marked as locations in the training set tended to be anonymized (rewritten as “X”). Therefore, our “location” heuristic was triggered whenever we encountered X and the hospital rules were not satisfied.

Heuristics for the condition tags were motivated less by a lack of training data than by a need to incorporate syntactic information from faraway words. Assuming that a condition of some modality was recognized, the task of the condition heuristics was to decide whether the modality should be “family,” “negation,” or “suspicion.” The family rule was simply triggered by any occurrence of a family word (son, daughter, mother, father, younger/older sister/brother, grandfather, grandmother) in the same sentence. The negation rule was the result of maintaining a polarity through the dependency parse tree[9][8] of the sentence. Each time a negative word or suffix (not (*-nai* and *-nu*), weak, improved) was encountered, the polarity of any the current clause and any dependent clauses was reversed. Suspicion tagging also followed the dependency parse tree. Whenever a suspicion trigger word (“doubt,” “think,” “possible”) occurred, dependent clauses were marked as suspicions, except for those clauses ending in “because” (*-kara* or *-yori*) or “in” (*-ni* or *-de*). Those exceptional clauses tended to indicate a reason for the suspicion, in which any condition mentioned was usually a definite finding, not a suspicion itself.

In all these cases, the result of the heuristics was not a final decision, but a feature. This way, the classifier could learn to use the heuristic suggestions in concert with the other features, adding more flexibility to the final decision.

1.3 Submitted systems

For the de-identification subtask, our three systems were all based around the same features — the normalized tokens, the corresponding primary and secondary POS, and the heuristics described in 1.2 — but differ in the combinations given as input to the CRF. The *simple* system considers co-occurrences between feature values within a window of width 5 around the current token. *extras* considers further co-occurrences between the heuristic features. Finally, *extras-b* also considers bigrams of output tags.

For the complaint and diagnosis subtask, we submitted two different architectures. The first, *singlestage*, consists in a regular CRF which makes full use of the following features and their combinations: normalized tokens, primary and secondary POS, LSD membership, the detected language (English or Japanese), UMLS membership, UMLS categories, and the heuristics. Our two other systems, *2stage* and *2stage-extras*, implement a two stage approach. The first stage consists in a CRF which has been trained to detect complaints, regardless of the modality. The second stage uses the heuristics to decide which modality to assign. The difference between *2stage* and *2stage-extras* is that the latter takes into account co-occurrences between the UMLS membership features and the normalized tokens.

2. RESULTS

Parameter and feature selection was done through 5-fold cross-validation on the training set, with a focus on the global F-1 score. Table 1 shows the average performance on the training set for task 1. Table 2 shows the final performance on the test set, broken down by class. The corresponding performance information for task 2 is in Tables 3 and 4.

System	P	R	F_1
tokens	91.00	79.35	84.74
with pos	90.82	80.96	85.55
with hints (*simple*)	90.69	87.08	88.78
with hints (*extras*)	90.56	87.79	89.11
with hints (*extras-b*)	90.58	87.12	88.77

Table 1: Task 1 average performance on the training set. Each row includes the features from above it.

Class	System	P	R	F_1
All	simple	91.67	86.57	89.05
	extras	90.82	87.04	88.89
	extras-b	90.05	87.96	88.99
A	simple	90.00	84.38	87.10
	extras	90.00	84.38	87.10
	extras-b	93.33	87.50	90.32
H	simple	97.06	86.84	91.67
	extras	91.89	89.47	90.67
	extras-b	89.47	89.47	89.47
L	simple	00.00	00.00	00.00
	extras	00.00	00.00	00.00
	extras-b	00.00	00.00	00.00
T	simple	91.30	89.36	90.32
	extras	91.24	88.65	89.93
	extras-b	90.65	89.36	90.00
X	simple	100.00	50.00	66.67
	extras	100.00	100.00	100.00
	extras-b	100.00	100.00	100.00

Table 2: Task 1 performance on the test set, broken down by class.

3. CONCLUSION

Despite the simplicity of the string matching approach, the domain-specific features boosted performance. A more elaborate concept detection algorithm would certainly help here.

The heuristics which incorporated prior knowledge helped on both tasks, with the exception of the location mentions. This is due to the scarcity of training data for the CRF to generalize over: there were only 2 location mentions in the training corpus. More generally, one could argue that, given more training data, the heuristics for task 1 would become progressively less relevant. However, the ones for task 2 would remain useful, as they look at longer range information than a CRF can by itself.

Stages	System	P	R	F_1
No modalities	tokens	88.81	66.17	75.82
	with pos	87.41	74.20	80.21
	with dicts	87.15	79.19	82.97
	with hints	87.35	80.60	83.83
	with hints+extras	87.25	80.81	83.90
Single stage	tokens	81.31	56.37	66.57
	with pos	79.40	63.17	70.33
	with dicts	76.41	67.04	71.42
	with hints	79.87	70.60	74.93
	with hints+extras*	79.65	71.06	75.09
2 stages	tokens	77.35	57.61	66.02
	with pos	75.56	64.16	69.36
	with dicts	74.44	67.64	70.87
	with hints*	74.12	68.38	71.13
	with hints+extras*	74.29	68.79	71.43

Table 3: Task 2 average performance on the training set. Each row includes the features from above it. The first part focuses on complaint detection with no modality, i.e. the first part of the 2-stage architecture. The two other parts show the performance on the full task. Submitted systems are marked with a *.

Class	System	P	R	F_1
No modalities	singlestage	89.76	77.81	83.36
	2stage	89.01	78.90	83.65
	2stage-extras	89.68	79.98	84.55
All	singlestage	81.15	70.35	75.36
	2stage	75.70	67.10	71.14
	2stage-extras	75.97	67.75	71.62
Positive	singlestage	80.92	73.28	76.91
	2stage	80.61	67.20	73.30
	2stage-extras	80.80	68.00	73.85
Family	singlestage	82.35	63.64	71.79
	2stage	71.43	68.18	69.77
	2stage-extras	65.22	68.18	66.67
Negation	singlestage	84.50	68.42	75.62
	2stage	74.32	66.80	70.36
	2stage-extras	75.23	67.61	71.22
Suspicion	singlestage	50.00	30.00	37.50
	2stage	36.36	66.67	47.06
	2stage-extras	35.85	63.33	45.78

Table 4: Task 2 performance on the test set, broken down by class.

4. REFERENCES

[1] A. R. Aronson. Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. *Proceedings / AMIA ... Annual Symposium. AMIA Symposium*, pages 17–21, 2001.

[2] T. Kudo. Crf++: Yet another CRF toolkit (`http://code.google.com/p/crfpp/`).

[3] T. Kudo. MeCab: Yet another Japanese dependency structure analyzer (`http://mecab.sourceforge.net/`).

[4] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th International Conf. on Machine Learning*, pages 282–289. Morgan Kaufmann, San Francisco, CA, 2001.

[5] M. Morita, Y. Kano, T. Ohkuma, M. Miyabe, and E. Aramaki. Overview of the NTCIR-10 MedNLP task. In *Proceedings of NTCIR-10*, 2013.

[6] NLM. UMLS 2012AA.

[7] L. S. D. Project. Life science dictionary (`www.life-science-dictionary.com`).

[8] N. Yoshinaga and M. Kitsuregawa. Polynomial to linear: Efficient classification with conjunctive features. In *EMNLP*, pages 1542–1551. ACL, 2009.

[9] N. Yoshinaga and M. Kitsuregawa. Kernel slicing: Scalable online training with conjunctive features. In C.-R. Huang and D. Jurafsky, editors, *COLING*, pages 1245–1253. Tsinghua University Press, 2010.