Disability annotation on documents from the biomedical domain

Pol Alvarez Vecino*, Lluís Padró†
*Barcelona Supercomputing Center. Barcelona, Spain. pol.alvarez@bsc.es
†TALP Research Center. Universitat Politècnica de Catalunya. Barcelona, Spain. padro@cs.upc.edu

Abstract

This paper describes the UPC_2 system participation in DIANN (*Disability annotation on documents from the biomedical domain*) shared task, framed in the IBEREVAL 2018 evaluation workshop[a]. The system tackles the detection of disabilities using a CRF to perform IOB Named Entity Recognition (NER). Regarding the detection of negated disabilities, the out-of-the-box NegEx rule-based system is used.

[a]http://nlp.uned.es/diann

1. Introduction: Task

- **Annotate disabilities** and their **negation** on documents from the biomedical domain.
- Input: **short texts** with the disabilities and negations **tagged with XML**.
- Simple disability annotation:

```
... reliability of the MCA in Spanish to identify <dis>mild cognitive
  impairment</dis> (<dis>MCI</dis>)...
```

- Negated disability annotation:

```
... <scp> <neg>without</neg> <dis>dementia</dis> </scp>, significant
  differences were obtained in terms ...
```

2. Approach

- The approach to the problem was to solve it in two steps:
- **Disabilities:**
  - Convert all words into tuples: *(word, POS, IOB-tag)*.
  - Use Conditional Random Fields (CRF) to predict the disability IOB-tags.
  - Convert IOB-tags to sentences/disabilities.
- **Negation:**
  - Feed each tuple sentence/disability to a negation software.
  - Filter out the probable false positives.
  - Convert input back again into XML tagged files.

3. Creating the training data

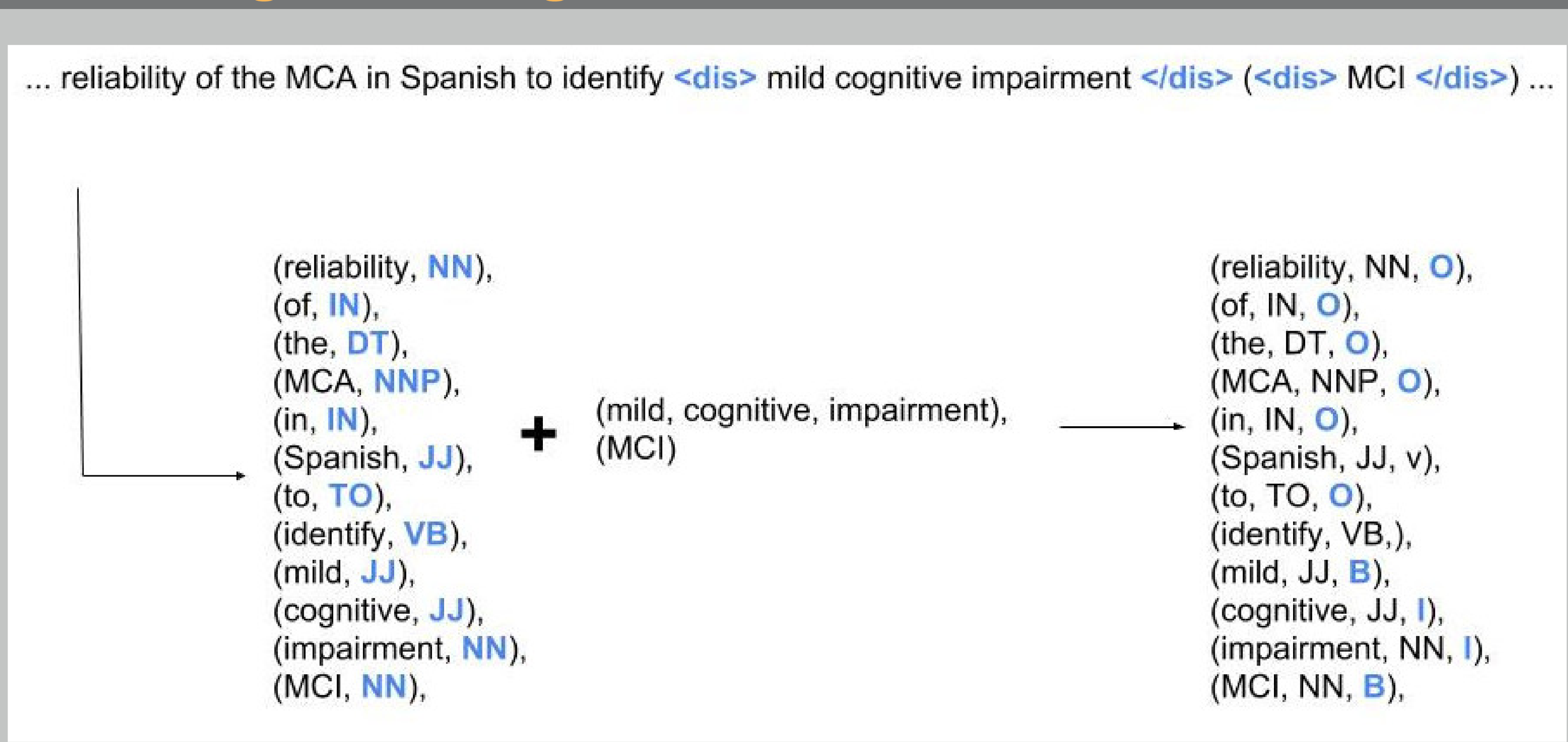
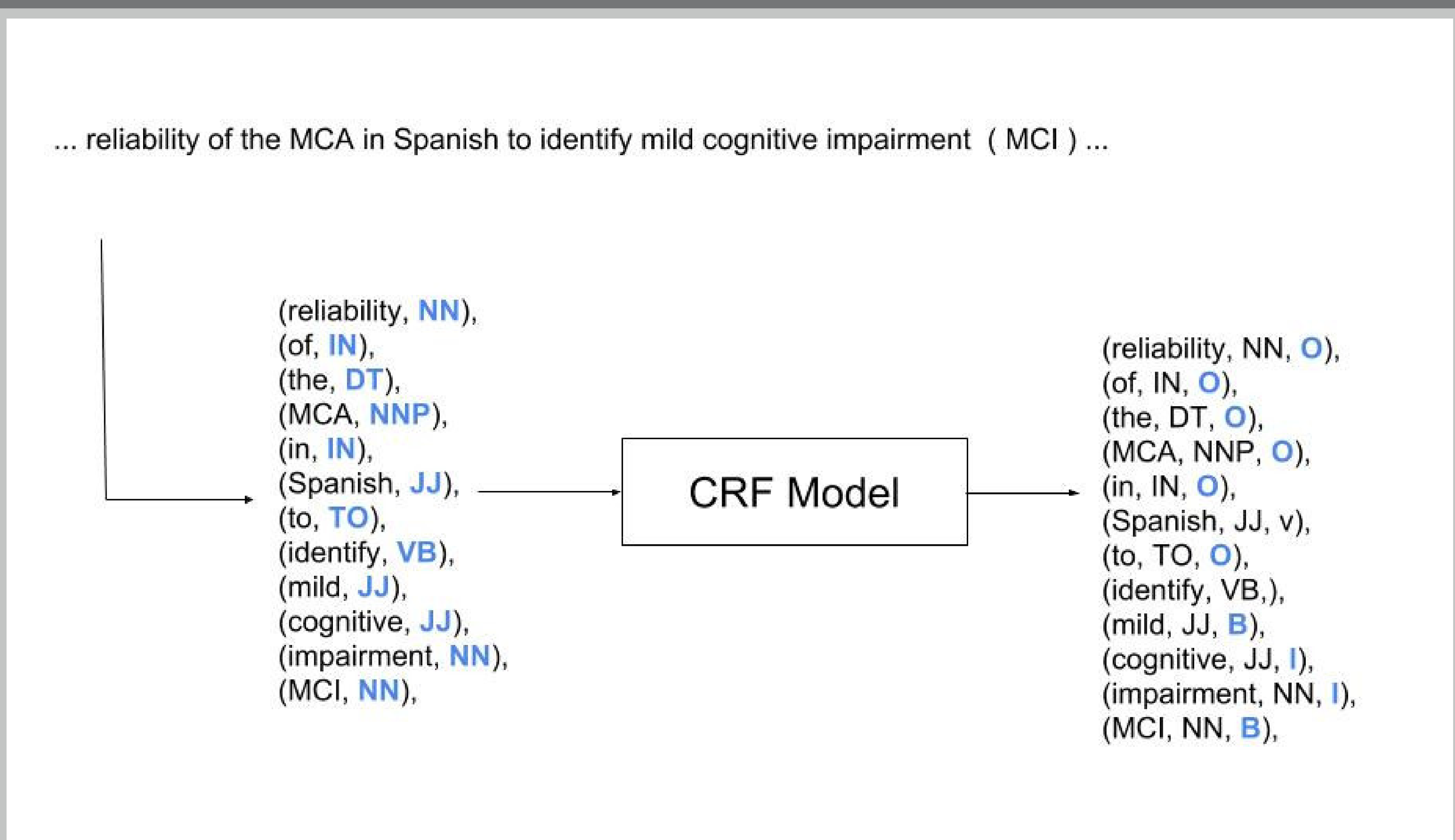


Figure: Pipeline to create the training data for the CRF model.

4. CRF Model & Features



Groups of features:

- 1. Derived from the **curent word** like word, POS, lemma or all-caps.
- 2. Build from **entitites/acronyms lists** such as is-acronym.
- 3. Derived from the **previous/next words** like word, POS, or lemma.
- 4. Concatenation of **current word's feature and next/previous'** one.
- 5. Concatenation of the **two previous/next** words' features.

5. CRF Model: Training

- Feature Selection[a]:
  - Start with all groups activated and with all features per group
  - Deactivate a group and check if precision increases/decrease
  - If precision increases:
    - Reactivate the group
    - Deactivate each feature of the group and reactivate it only if precision decreases
  - If precision decreases just remove the group from the feature's set
- Lists' Creation:
  - During model evaluation, entities/acronyms lists are created out of the training fold
  - Once the model is chosen, built the lists with all the training data

6. Negation Detection

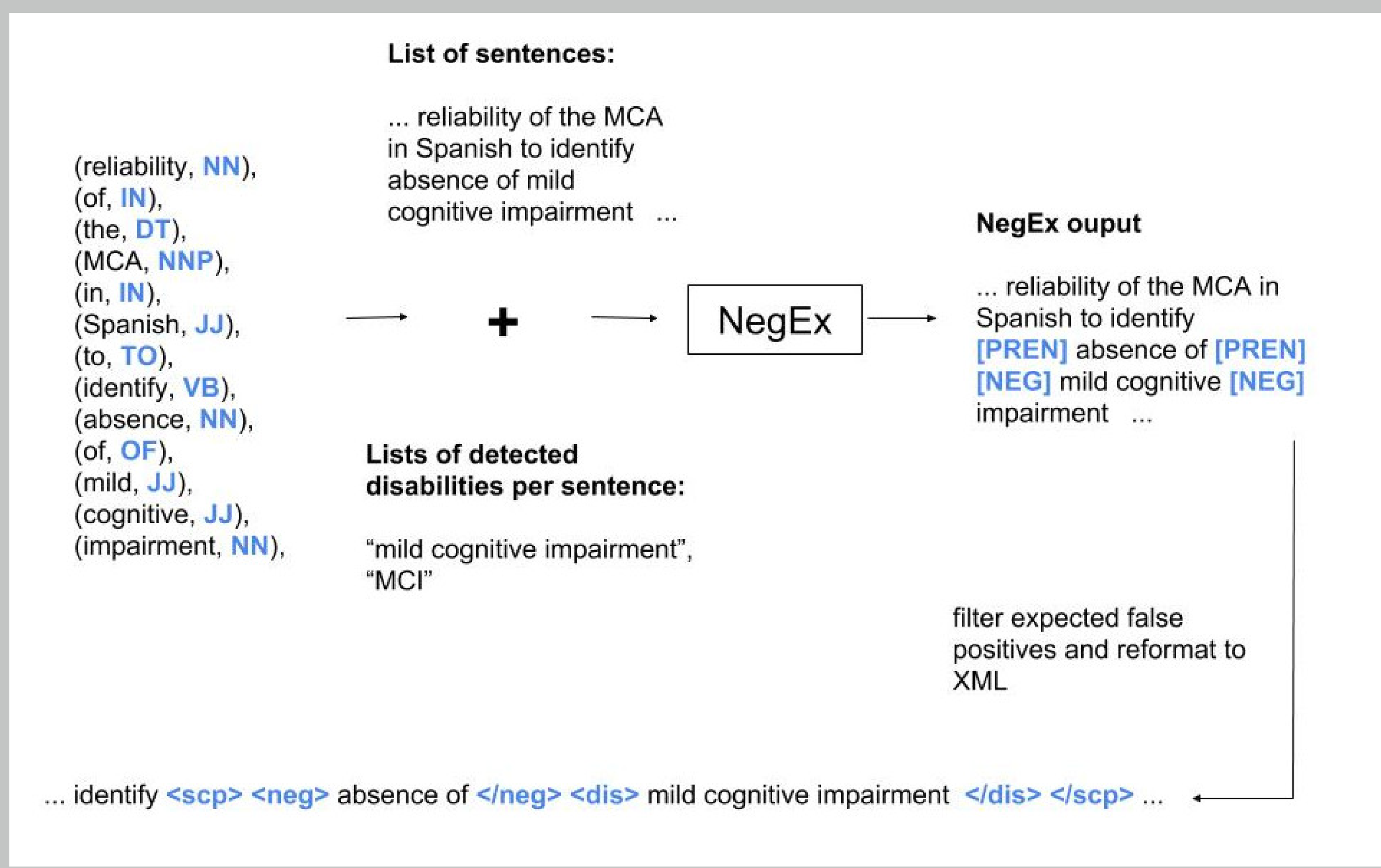


Figure: Pipeline to tag negated phrases and final XML formatting.

7. Experiments and Results

	Spanish			English		
	Precision	Recall	F1 score	Precision	Recall	F1 score
Group disabled: 2						
Negation	0.50	0.55	0.52	0.46	0.35	0.40
Disability	0.72	0.63	**0.68**	0.72	0.58	0.64
Group disabled: 3						
Negation	0.50	0.50	0.50	**0.48**	0.35	0.41
Disability	0.73	0.51	0.60	**0.75**	0.56	0.64
Group disabled: 4						
Negation	0.51	**0.58**	**0.54**	**0.48**	0.40	**0.44**
Disability	**0.74**	0.59	0.65	0.74	0.65	0.70
Group disabled: 5						
Negation	0.51	0.55	0.53	**0.48**	0.38	0.42
Disability	0.71	0.59	0.64	**0.75**	0.65	0.69
Group disabled: None						
Negation	**0.52**	0.55	0.53	0.47	**0.41**	0.43
Disability	**0.74**	0.62	**0.68**	**0.75**	**0.67**	**0.71**

Table: Results of cross-validation experiments deactivating one feature group at a time.

	Exact Match			Partial Match		
	Precision	Recall	F1 score	Precision	Recall	F1 score
English						
Disability	0.756	0.560	0.643	0.822	0.588	0.686
Negated Disability	0.647	0.478	0.550	0.941	0.696	0.800
Non-negated + Negated Disability	0.724	0.519	0.604	0.822	0.588	0.686
Spanish						
Disability	0.732	0.502	0.596	0.828	0.568	0.674
Negated Disability	0.737	0.636	0.683	0.895	0.773	0.829
Non-negated + Negated Disability	0.710	0.480	0.573	0.819	0.555	0.661

Table: Final testing results with the full-featured model.

[a]All validation results use 10-fold cross-validation