UCM-2: a Rule-Based Approach to Infer the Scope of Negation via Dependency Parsing

Miguel Ballesteros, **Alberto Díaz**, Virginia Francisco, Pablo Gervás, Jorge Carrillo de Albornoz and Laura Plaza

Montreal, *SEM Shared Task 2012

Introduction

Motivation

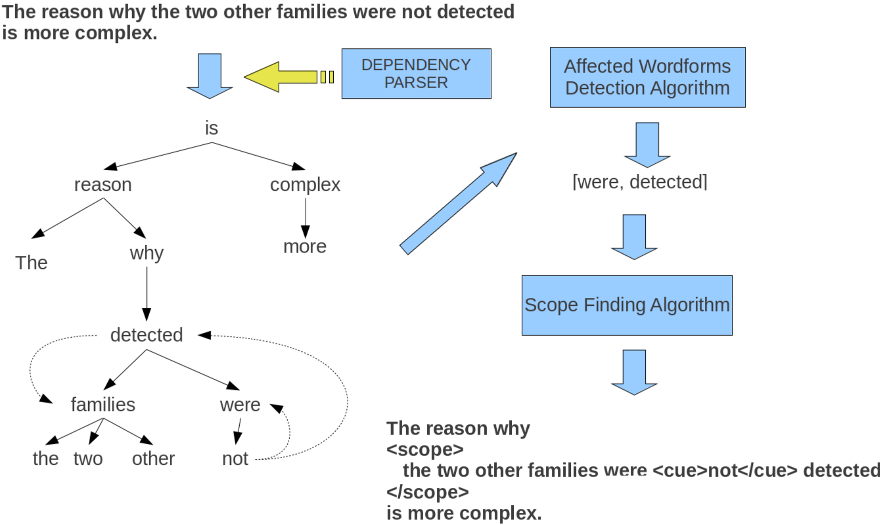
- Negation, simple in concept, is a complex but essential phenomenon in any language.
- It turns an affirmative statement into a negative one, changing the meaning completely.
- We believe that being able to handle and classify negations we would be able to improve several text mining applications.

Introduction

Goal

- The system presented in this paper is a modification of other previous system that was build to handle the sentences included in the Bioscope corpus (Ballesteros et al. 2012).
- This original system consisted of two algorithms:
  - **Affected Wordforms Detection Algorithm**: it is capable of inferring words affected by the negation cues by traversing dependency trees
  - **Scope Finding Algorithm**: it classifies the scope of the negations by using a rule-based approach that studies linguistic clause boundaries and the outcomes of the previous algorithm

Example



The reason why
<scope>
the two other families were <cue>not</cue> detected
</scope>
is more complex.

Methodology

- The system presented to the Shared Task consist of four algorithms:
    - **Affected Wordforms Detection Algorithm**: with only Negation Cue Lexicon modified.
    - **Scope Finding Algorithm**: modified to make it able to handle the more complex negation structures in the Conan Doyle Corpus.
    - **Negation Event Handling**: a very naive rule-based approach to annotate the negated event.
    - **Post-Processing Step**: translates the outcomes in BioScope format to the final format for the Shared Task.

Negation Cue Lexicon

An Excerpt of the Static Lexicon

not	no	neither..nor
unnecessary	unoccupied	unpleasant
unpractical	unsafe	unseen
unshaven	windless	without

Table : Excerpt of the lexicon

- In addition to the previous lexicon (less than 20 different negation cues), we analysed the training set and development sets and extracted 153 new different negation cues.

Affected Wordforms Detection Algorithm

The algorithm runs through the dependency tree of a sentence and does the following steps:

1. It detects all the nodes that are contained in the lexicon of cues, and marked them as cues.
    - If the negation cue is not a verb, the algorithm marks the main verb (if exists) that govern the structure, as affected.
2. For the rest of nodes, if a node directly depends on any of the ones previously marked as affected, the system marks it as affected.
3. The detection of affected wordforms is propagated through the dependency graph until terminals are found.

The Scope Finding Algorithm

Scope opening

The algorithm runs through the original sentence linearly token by token and applies the next rules:

- If the token is contained in the set of nodes affected by a negation cue (and the scope is not open): the system opens the scope at the token and establishes that the scope for the cue involved is already opened.
- If the token is a cue: the system goes backward and opens the scope when it finds the subject involved or a marker that indicates another statement, like a comma.
    - In Bioscope, only the sentences in passive voice include the subject inside the scope. However, the Conan Doyle corpus does not contain this exception always including the subject in the scope when it exists.
    - We modified the decision that fires this rule, and we apply the way of annotating sentences in passive voice for all the negation cues, either passive or active voice sentences.

The Scope Finding Algorithm

Scope closing

- If the token is a punctuation symbol, followed by some wordforms that indicate another statement, such as *but*: the system closes the scope just after the token.
- If the token is any wordform and all the nodes that are marked as affected by the negation cue are already included in the scope: the system closes the scope just before the token.
- If the token is at the end of sentence: the system closes the scope at the end of the sentence.

The Scope Finding Algorithm

Affixal negations

We also added a new rule to handle the negation cues that are prefix or suffix of another word, such as *meaning-less*.

- If the system finds a cue word like this, it then annotates the suffix or prefix as the cue (such as *less*) and the rest of the word as part of the scope.
- Note that the Affected Wordforms Detection Algorithm detects the whole word as a cue word.

Negated Event Algorithm

In order to come up with a solution that could provide at least some results in the negated event handling, we decided to do the following:

- When the cue word contains a negative prefix or a negative suffix, we annotate the word as the negated event.
- When the cue word is either *not* or *n't* and the next word is a verb, according to the part-of-speech annotation of the Conan Doyle corpus, we annotate the verb as the negated event.

PostProcessing Step

Processes the annotated sentence with Bioscope style, in order to provide the expected output.

- If the token contains the string <*scope*>, the system just starts a new scope column reserving three new columns and it puts the word in the first free "scope" column.
- If the token is between a <*cue*> annotation, the system puts it in the corresponding free "cue" column of the scope already opened.
- If the token is annotated as "negated event", the system just puts the word in the last column of the scope already opened.

Note that these three rules are not exclusive and can be fired for the same token, but in this case they are fired in the same order as they are presented.

Results

Test set	gold	system	tp	fp	fn	precision (%)	recall (%)	F1 (%)
Cues:	264	235	170	39	94	81.34	64.39	71.88
Scopes(cue match):	249	233	96	47	153	67.13	38.55	48.98
Scopes(no cue match):	249	233	96	48	152	66.90	38.96	49.24
Scope tokens(no cue match):	1805	2096	1222	874	583	58.30	67.70	62.65
Negated(no cue match):	173	81	36	42	134	46.15	21.18	29.03
Full negation:	264	235	29	39	235	42.65	10.98	17.46

Table : Test set results.

Development	gold	system	tp	fp	fn	precision (%)	recall (%)	F1 (%)
Cues:	173	161	115	16	58	87.79	66.47	75.66
Scopes(cue match):	168	160	70	17	98	80.46	41.67	54.90
Scopes(no cue match):	168	160	70	17	98	80.46	41.67	54.90
Scope tokens(no cue match):	1348	1423	1012	411	336	71.12	75.07	73.04
Negated(no cue match):	122	71	35	31	82	53.03	29.91	38.25
Full negation:	173	161	24	16	149	60.00	13.87	22.53

Table : Development set results.

Error Analysis

The main problem is related with the management of sentences with more than one scope. In some cases all the cues are assigned to all the scopes detected in the same sentence, generating false positives (1). In other cases, the cues of the second and subsequent scopes are ignored, generating false negatives (2), (3).

- (1) But no [one can glance at your toilet and attire without [seeing that your disturbance dates from the moment of your waking .. ’]]
- (2) [You do]n’t [mean] - . [you do] n’t [mean that I am suspected] ? ”
- (3) Our client smoothed down [his] un[brushed hair] and felt [his] un[shaven chin].

Error Analysis

A different kind of false positives is related to modality cues, dialogue elements and special cases.

- (4) " You traced him through the telegram , no [doubt]., " said Holmes .
- (5) " All you desire is a plain statement , [is it] not ? '.
- (6) Telegraphic inquiries ... that [Marx knew] nothing [of his customer save that he was a good payer] .

Error Analysis

- Problems with affixal negations, that is, bad separation of the affix and root of the word.
  - (7) He said little about the case, but from that little we gathered that [he also was not di[ssatisfied] at the course of events].
- Annotation errors?: incredible is not annotated as negation cue in (8)
  - (8) “Have just had most *incredible* and grotesque experience.
  - it appears 5 times in the training corpus, 2 times is labelled as a cue, but 3 times is not.

Conclusions

- We have adapted a rule-based system that was designed to be used in a different domain and with a different way to annotate the scope.
- The process has been complicated, but we are tempted to say that our results are competitive.
- But we have a lot of room for improvement:
    - improve the management of sentences with more than one scope.
    - replacing the dependency parser with a state-of-the-art parser in order to get higher performance.
    - proposing a different way of getting a reliable lexicon of cues, by using a semantic approach.

THANK YOU