UWashington: Negation Resolution using Machine Learning Methods

James Paul White
Masters Student in Computational Linguistics
University of Washington
jimwhite@uw.edu

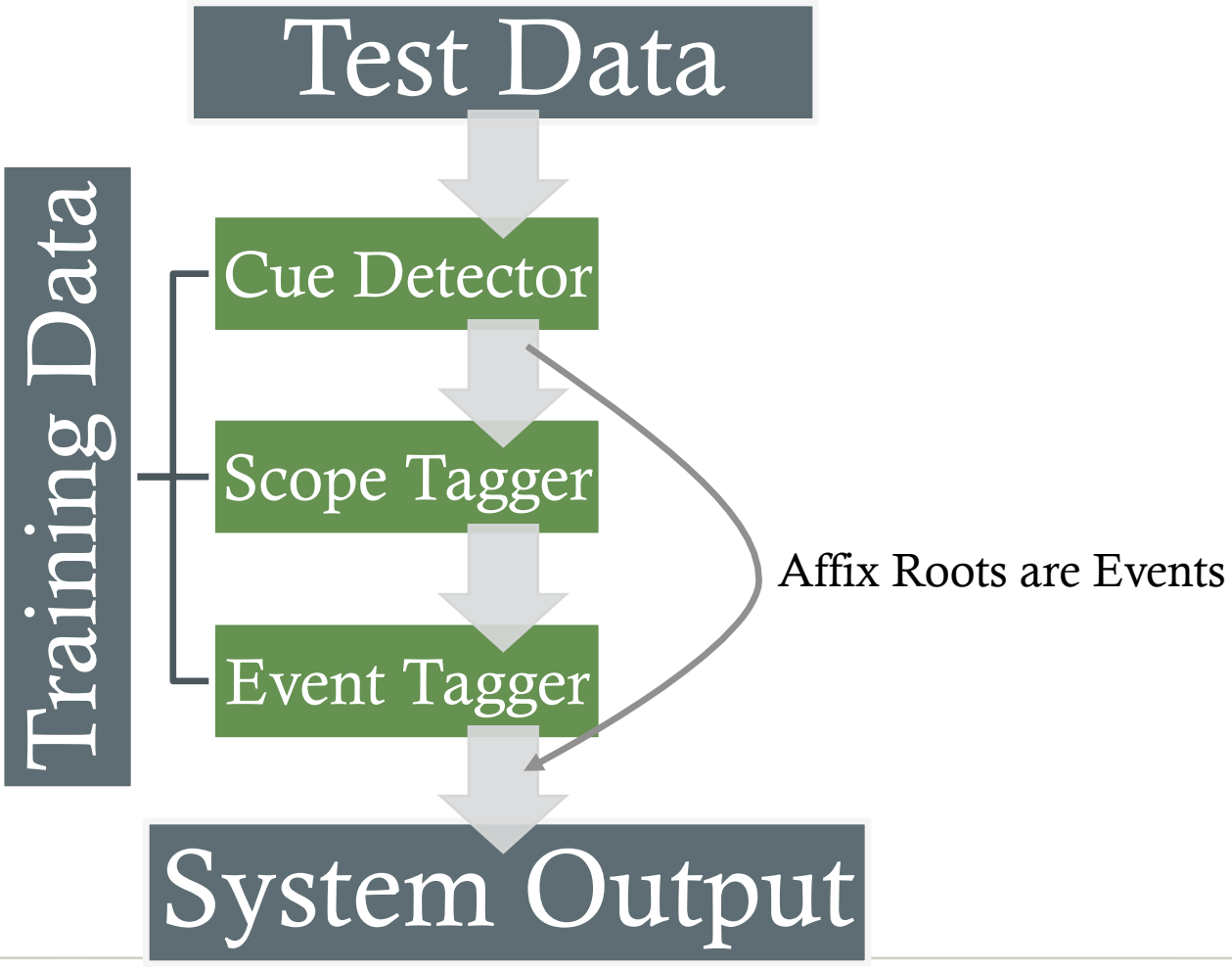
*SEM 2012 Shared Task

- Resolving the Scope of Negation
- Open Track
  - Initially investigated an approach using the semantic representation produced by the LinGO English Resource Grammar (ERG)
  - Interesting but not practical in time available
- Closed Track
  - Decided on a simple approach applying easy-to-use machine learning methods

Outline

- Architecture
- Cue Detector
- MALLET SimpleTagger
- Data Processing
- Computed Features for CRF Sequence Tagger
- Errors
- Results
- Improvement?

Architecture



Cue Detector

- Four Types of Negation Cue
  - Word : RB/**not**, DT/**no**, IN/**without**
  - Affix : VBN/**un**curtained, JJ/colour**less**, NN/care**less**ness
  - Multiple Contiguous Words : RB/**rather** IN/**than**
  - Multiple Discontiguous ("Gappy") Words :
    DT/**neither** … CC/**nor**
- Two Pass Learning of Regular Expression Patterns
  - First Pass : Add a matching pattern for every gold cue
    Patterns include POS tag in addition to word string
  - Second Pass : Accumulate positive/negative match scores
    Pattern type-specific rules condition if/when they will match

Oops

- Cue detector could learn negative contextual rules to override matches for idioms not labeled as cues
  - (be|have) **no** doubt
  - **none** the (less|worse)
- That is not implemented, but two hand-written rules to filter those two patterns were inadvertently used in the system

MALLET SimpleTagger

- Andrew McCallum. 2002. MALLET: A Machine Learning for Language Toolkit. http://mallet.cs.umass.edu
- Java, Open Source Software (Common Public License)
- Input format one line per token delimited by blank
  - Each line is list a of features present followed by label
- Output format is same
  - Each line is the tagger's output label
- The **cc.mallet.fst.SimpleTagger** class provides default settings for parameters

Computed Features for CRF Sequence Tagger

- Wisteria Lane, Chapter 01, Sentence 3

```
(S
  (NP
     (token PRP He He in_scope))
  (VP
     (token VBD made make in_scope)
     (NP
        (token wisteria01/3/0/2 DT no no cue=no in_scope)
        (token wisteria01/3/0/3 NN remark remark in_scope)))
```

- For each token (`He` in this example):
  - Relation between cue and token
    non-terminal node label plus relative position of child subtrees : **`S_X-1`**
  - Path (non-terminal node labels) from cue up to lowest common parent : **`NP_VP`**
  - Path from lowest common parent down to token : **`NP`**
  - Whether the token is an embedded child of parent
    (syntax fragment == "*") : **`in=false`**

Data Processing

CoNLL

```
wisteria01  3   0   He      He      PRP (S(S(NP*)    _    He      _
wisteria01  3   1   made    make    VBD (VP*         _    made    made
wisteria01  3   2   no      no      DT  (NP*         no   _       _
wisteria01  3   3   remark  remark  NN  *)))        _    remark  remark
```

S-expression

```
(S
  (NP
     (token wisteria01/3/0/0 PRP He He _ + - 0))
  (VP
     (token wisteria01/3/0/1 VBD made make _ + + 1)
     (NP
        (token wisteria01/3/0/2 DT no no no - - 2)
        (token wisteria01/3/0/3 NN remark remark _ + + 3))))
```

MALLET SimpleTagger

```
cpr_S_X-1 up_NP_VP down_NP dist=-2 cue_word_no cue_lemma_no cue_pos_DT in=false pos_PRP
   word_he lemma_he +

cpr_VP_X-1 up_NP down_ dist=-1 cue_word_no cue_lemma_no cue_pos_DT in=false pos_VBD
   word_made lemma_make +

cpr_NP_H0 up_ down_ dist=0 cue_word_no cue_lemma_no cue_pos_DT in=false is_cue pos_DT
   word_no lemma_no !

cpr_NP_H1 up_ down_ dist=1 cue_word_no cue_lemma_no cue_pos_DT in=false pos_NN
   word_remark lemma_remark +
```

Errors

- A clear systematic error occurs with affix cues because their roots are unconditionally output as an event
- Negated event tagging is weak
  - There are relatively few negated events because of the requirement for factuality
  - This is a data-driven approach and more data might help

Results

	Gold	System	TP	FP	FN	Precision (%)	Recall (%)	F1 (%)
Cues	264	285	243	33	21	88.04	92.05	90.00
Scopes (no cue match)	249	270	158	33	89	82.90	64.26	72.40
Scope tokens (no cue match)	1805	1816	1512	304	293	83.26	83.77	83.51
Negated (no cue match)	173	154	83	60	80	58.04	50.92	54.25
Full negation	**264**	**285**	**94**	**33**	**170**	**74.02**	**35.61**	**48.09**
Cues B	264	285	243	33	21	85.26	92.05	88.52
Scopes B (no cue match)	249	270	158	33	89	59.26	64.26	61.66
Negated B (no cue match)	173	154	83	60	80	53.9	50.92	52.37
Full negation B	264	285	94	33	170	32.98	35.61	34.24

Alternate run without idiom filter and without allowing a suffix after an affix cue:

	264	**284**	**98**	**35**	**171**	**73.68**	**34.94**	**47.4**

Improvement?

- Use CRF sequence tagger for affix cues
- Learn more complex cue patterns so that exceptional negative cases can be rejected
- Include some feature(s) to provide context to lowest common syntax node for scope tagging

Thank You

- Organizers
  - Roser Morante & Eduardo Blanco
- Reviewers
- Emily Bender
- Groovy
  - http://groovy.codehaus.org/
- Source Code
  - https://github.com/jimwhite/SEMST2012