UWashington: Negation Resolution using Machine Learning Methods

James Paul White
University of Washington
Department of Linguistics, Box 354340
Seattle, WA 98195, USA
jimwhite@uw.edu

Abstract

This paper reports on a simple system for resolving the scope of negation in the closed track of the *SEM 2012 Shared Task. Cue detection is performed using regular expression rules extracted from the training data. Both scope tokens and negated event tokens are resolved using a Conditional Random Field (CRF) sequence tagger – namely the SimpleTagger library in the MALLET machine learning toolkit. The full negation F_1 score obtained for the task evaluation is 48.09% (P=74.02%, R=35.61%) which ranks this system fourth among the six submitted for the closed track.

1 Introduction

Resolving the scope of negation is an interesting area of research for Natural Language Processing (NLP) systems because many such systems have used methods that are insensitive to polarity. As a result it is fairly common to have a system that treats "X does Y" and "X does not Y" as having the same, or very nearly the same, meaning[1]. A few application areas that have been addressing this issue recently are in sentiment analysis, biomedical NLP, and recognition of textual entailment. Sentiment analysis systems are frequently used in corporate and product marketing, call center quality control, and within "recommender" systems which are all contexts where it is important to recognize that "X does like Y" is contrary to "X does not like Y". Similarly in biomedical text such as research papers and abstracts, diagnostic procedure reports, and medical records it is important to differentiate between statements about what is the case and what is not the case.

The *SEM 2012 Shared Task is actually two related tasks run in parallel. The one this system was developed for is the identification of three features of negation: the cue, the scope, and the factual negated event (if any). The other task is concerned with the focus of negation. Detailed description of both subtasks, including definition of the relevant concepts and terminology (negation, cue, scope, event, and focus) appears in this volume (Morante and Blanco, 2012). Roser Morante and Eduardo Blanco describe the corpora provided to participants with numbers and examples, methods used used to process the data, and briefly describes each participant and analyzes the overall results.

Annotation of the corpus was undertaken at the University of Antwerp and was performed on several Sherlock Holmes works of fiction written by Sir Arthur Conan Doyle. The corpus includes all sentences from the original text, not just those employing negation. Roser Morante and Walter Daelemans provide a thorough explanation of those gold annotations of negation cue, scope, and negated event (if any) (Morante and Daelemans, 2012). Their paper explains the motivations for the particular annotation decisions and describes in detail the guidelines, including many examples.

2 Related Work

Recognition of phrases containing negation, particularly in the medical domain, using regular expressions has been described using several different approaches. Systems such as Negfinder (Mutalik et

[1] A one token difference between the strings surely indicating at least an inexact match.

*First Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 335–339,
Montréal, Canada, June 7-8, 2012.

al, 2001) and NegEx (Chapman et al, 2001) use manually constructed rules to extract phrases from text and classify them as to whether they contain an expression of negation. Rokach et al evaluate several methods and show their highest level of performance (an F_1 of 95.9 ± 1.9%) by using cascaded decision trees of regular expressions learned from labelled narrative medical reports (Rokach et al, 2008).

Those systems perform a different function than that required for this task though. They classify phrases extracted from plain text as to whether they contain negation or not, while the requirement of this shared task for negation cue detection is to identify the particular token(s) or part of a token that signals the presence of negation. Furthermore, those systems only identify the scope of negation at the level of phrasal constituents, which is different than what is required for this task in which the scopes are not necessarily contiguous.

Conditional Random Field (CRF) sequence taggers have been successfully applied to many scope resolution problems, including those of negation. The NegScope system (Agarwal and Yu, 2010) trains a CRF sequence tagger on labelled data to identify both the cue and scope of negation. However, that system only recognizes a whole word as a cue and does not recognize nor generalize negation cues which are affixes. There are also systems that use CRF sequence taggers for detection of hedge scopes (Tang et al, 2010, Zhao et al, 2010). Morante and Daelemans describe a method for improving resolution of the scope of negation by combining IGTREE, CRF, and Support Vector Machines (SVM) (Morante and Daelemans, 2009).

3 System Description

This system is implemented as a three stage cascade with the output from each of the first two stages included as input to the subsequent stage. The stages are ordered as cue detection, scope detection, and finally negated event detection. The format of the inputs and outputs for each stage use the shared task's CoNLL-style file format. That simplifies the use of the supplied gold-standard data for training of each stage separately.

Because this system was designed for the closed track of the shared task, it makes minimal language-specific assumptions and learns (nearly) all language-specific rules from the gold-labelled training data (which includes the development set for the final system).

The CRF sequence tagger used by the system is that implemented in the SimpleTagger class of the MALLET toolkit, which is a Java library distributed under the Common Public License[2].

The system is implemented in the Groovy programming language, an agile and dynamic language for the Java Virtual Machine[3]. The source code is available under the GNU Public License on GitHub[4].

3.1 Cue Detection

Cues are recognized by four different regular expression rule patterns: affixes (partial token), single (whole) token, contiguous multiple token, and gappy (discontiguous) multiple token. The rules are learned by a two pass process. In the first pass, for each positive example of a negation cue in the training data, a rule that matches that example is added to the prospective rule set. Then, in the second pass, the rules are applied to the training data and the counts of correct and incorrect matches are accumulated. Rules that are wrong more often than they are right are removed from the set used by the system.

A further filtering of the prospective rules is done in which gappy multiple token rules that match the same word type more than once are removed. Those prospective rules are created to match cases in the supplied training data where the a repetition has occurred and then encoded by the annotators as a single cue (and thus scope) of negation[5].

The single token and multiple token rules match both the word string feature (ignoring case) and the part-of-speech (POS) feature of each token. And because a single token rule might also match a cue that belongs to a multiple token rule, multiple token rules are checked first.

Affix rules are of two types: prefix cues and non-prefix cues. The distinction is that while prefix cues must match starting at the beginning of the word string, the non-prefix cues may have a suffix following them in the word string that is not part of the cue. Affix rules only match against the word

[2] http://mallet.cs.umass.edu/

[3] http://groovy.codehaus.org/

[4] https://github.com/jimwhite/SEMST2012

[5] Such as baskervilles12 174: "Not a whisper, not a rustle, rose..." which has a cue annotation of "Not" *gap* "not".

string feature of the tokens and are insensitive to the POS feature.

In order to generalize the affix rules, sets are accumulated of both base word strings (the substring following a prefix cue or substring preceding a non-prefix cue) and suffixes (the substring following non-prefix cues, if any). In addition, all other word strings and lemma strings in the training corpus that are at least four characters long are added to the set of possible base word strings[6]. A set of negative word strings is also accumulated in the second pass of the rule training to condition against false positive matches for each affix rule.

A prefix cue rule will match a token with a word string that starts with the cue string and is followed by any of the strings in the base word set. Similarly a suffix cue rule will match a token whose word string contains the cue string preceded by a string in the base word set and is either at the end of the string or is followed by one of the strings in the suffix string set. Affix rules, unlike the other cue-matching rules, also output the string for matched base word as the value of the scope for the matched token. In any case, if the token's word string is in the negative word string set for the rule then it will not be matched.

Following submission of the system outputs for the shared tasked I discovered that a hand written regular expression rule that filters out the (potential) cues detected for "(be|have) no doubt" and "none the (worse|less)" was inadvertently included in the system. Although those rules could be learned automatically from the training data (and such was my intention), the system as reported here does not currently do so.

3.2 Negation Scope Resolution

For each cue detected, scope resolution is performed as a ternary classification of each token in the sentence as to whether it is part of a cue, part of a scope, or neither. The classifier is the CRF sequence tagger implemented in the SimpleTagger class of the MALLET toolkit (McCallum, 2002). Training is performed using the gold-standard data including the gold cues. The output of the tagger is not used to determine the scope value of a token in those cases where an affix rule in the cue detector has matched a token and therefore has supplied the matched base word string as the value of the scope for the token.

For features that are computed in terms of the cue token, the first (lowest numbered) token marked as a cue is used when there is more than one cue token for the scope.

Features used by the scope CRF sequence tagger are:

- Of the per-token data: word string in lowercase, lemma string in lowercase, part-of-speech (POS) tag, binary flag indicating whether the token is a cue, a binary flag indicating whether the token is at the edge of its parent non-terminal node or an internal sibling, a binary flag indicating whether the token is a cue token, and relative position to the cue token in number of tokens.
- Of the cue token data: word string in lowercase, lemma string in lowercase, and POS tag.
- Of the path through the syntax tree from the cue token: an ordered list of the non-terminal labels of each node up the tree to the lowest common parent, an ordered list of the non-terminal labels of each node down the tree from that lowest common parent, a path relation value consisting of the label of the lowest common parent node concatenated with an indication of the relative position of the paths to the cue and token in terms of sibling order.

3.3 Negated Event Resolution

Detection of the negated event or property is performed using the same CRF sequence tagger and features used for scope detection. The only difference is that the token classification is in terms of whether each token in the sentence is part of a factual negated event for each negation cue.

3.4 Feature Set Selection

A comparison of the end-to-end performance of this system using several different sets of per token feature choices for the scope and negated event classifiers is shown in Table 1. In each case the training data is the entire training data and the dev data is the entire dev data supplied by the organizers for this shared task. The scores are computed

[6]This "longer than four character" rule was manually created to correct for over-generalization observed in the training data. If the affix rule learner selected this value using the correct/incorrect counts as it does with the other rule parameters then this bit of language-specific tweaking would be unnecessary.

		Gold	System	TP	FP	FN	Precision (%)	Recall (%)	F1 (%)
Baseline	(train)	984	1034	382	56	602	87.21	38.82	53.73
	(dev)	173	164	34	9	139	79.07	19.65	31.48
Set 1	(train)	984	1034	524	56	460	90.34	53.25	67.00
	(dev)	173	164	60	9	113	86.96	34.68	49.59
Set 2	(train)	984	1034	666	56	318	92.24	67.68	78.07
	(dev)	173	164	61	9	112	87.14	35.26	50.21
System	(train)	984	1034	644	56	340	92.00	65.45	76.49
	(dev)	173	164	68	9	105	88.31	39.31	54.40

Table 1: Comparison of full negation scores for various feature sets.

by the evaluation program also supplied by the organizers. The baseline features are those provided in the data, with the exception of the syntactic tree fragment: word string in lowercase, lemma in lowercase, and POS tag. The "set 1" features are the remainder of the features described in section 3.2, with the exception of those of the path through the syntax tree from the cue token. The "set 2" features are the three baseline features plus the three features of the path through the syntax tree from the cue token: list of non-terminal labels from cue up to the lowest common parent, lowest common parent label concatenated with the relative distance in nodes between the siblings, list of non-terminals from the lowest common parent down to the token. The "system" feature set is the union of set 1 and set 2, and is the one used by the submitted system.

The baseline score is an F_1 of 31.5% (P=79.1%, R=19.7%) on the dev data. Using either feature set 1 or 2 results in substantially better performance. They achieve nearly the same score on the dev set with an F_1 of 50±0.5% (P=87±0.2%, R=35±0.3%) in which the difference is that between one case of true positive vs. false negative out of 173. The combination of those feature sets is better still though with an F_1 of 54.4% (P=88.3%, R=39.3%).

4 Results

Table 2 presents the scores computed for the system output on the held-out evaluation data. The F_1 for full negation is 48.1% (P=74%, R=35.6%), which is noticeably lower than that seen for the dev data (54.4%). That reduction is to be expected because the dev data was used for system tuning. There was also evidence of significant over-fitting to the training data because the F_1 for that was 76.5% (P=92%, R=65.5%). The largest component of the fall off in performance is in the recall.

The worst performing component of the system is the negated event detection which has an F_1 of 54.3% (P=58%, R=51%) on the evaluation data. One contributor to low precision for the negated event detector is that the root word of an affix cue is always output as a negated event, bypassing the negated event CRF sequence classifier. In the combined training and dev data there is a total of 1157 gold cues (and scopes) of which 738 (63.8%) are annotated as having a negated event. Of the 1198 cues the system outputs for that data, 188 (15.7%) are affix cues, each of which will also be output as a negated event. Therefore it would be reasonable to expect that approximately 16 (27.7%) of the false positives for the negated event in the evaluation (60) are due to that behavior.

	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

Table 2: System evaluation on held-out data.

5 Conclusion

This paper describes the system I implemented for the closed track of the *SEM 2012 Shared Task for negation cue, scope, and event resolution. The system's performance on the held-out evaluation data, an F_1 of 48.09% (P=74.02%, R=35.61%) for the full negation, relative to the other entries for the task is fourth among the six teams that participated.

The strongest part of this system is the scope resolver which performs at a level near that of the best-performing systems in this shared task. I think it is likely that the performance on scope resolution would be equivalent to them with a better negation cue detector. That is supported by the "no cue match" version of the scope resolution evaluation for which this system has the highest F_1 (72.4%).

Clearly the weakest link is the negated event detector. Since one obvious source of error is that the root word extracted when an affix cue is detected is always output as a negated event, a promising approach for improvement would be to instead utilize that as a feature for the negated event's CRF sequence tagger so that they have a chance to be filtered out in non-factual contexts.

Acknowledgements

I want to thank Roser Morante and Eduardo Blanco for organizing this task, the reviewers for their thorough and very helpful suggestions, and Emily Bender for her guidance.

References

Shashank Agarwal and Hong Yu. 2010. Biomedical negation scope detection with conditional random fields. *Journal of the American Medical Informatics Association*, *17*(6), 696–701. doi:10.1136/jamia.2010.003228

Chapman, W. W., Bridewell, W., Hanbury, P., Cooper, G. F., & Buchanan, B. G.. 2001. A simple algorithm for identifying negated findings and diseases in discharge summaries. *Journal of Biomedical Informatics*, *34*(5), 301–310. doi:10.1006/jbin.2001.1029

Andrew McCallum. 2002. MALLET: A Machine Learning for Language Toolkit. Retrieved from http://mallet.cs.umass.edu

Roser Morante and Eduardo Blanco. 2012. *SEM 2012 Shared Task: Resolving the Scope and Focus of Negation. *Proceedings of the First Joint Conference on Lexical and Computational Semantics*. Presented at the *SEM 2012, Montreal, Canada.

Roser Morante and Walter Daelemans. 2009. A Metalearning Approach to Processing the Scope of Negation. *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)* (pp. 21–29). Boulder, Colorado: Association for Computational Linguistics.

Roser Morante and Walter Daelemans. 2012. Conan-Doyle-neg: Annotation of negation in Conan Doyle stories. *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC)*.

Pradeep G. Mutalik, Aniruddha Deshpande, and Prakash M. Nadkarni. 2001. Use of general-purpose negation detection to augment concept indexing of medical documents: a quantitative study using the UMLS. *Journal of the American Medical Informatics Association: JAMIA*, *8*(6), 598–609.

Lior Rokach, Roni Romano, and Oded Maimon. 2008. Negation recognition in medical narrative reports. *Information Retrieval*, *11*(6), 499–538. doi:10.1007/s10791-008-9061-0

Buzhou Tang, Xiaolong Wang, Xuan Wang, Bo Yuan, and Shixi Fan. 2010. A Cascade Method for Detecting Hedges and their Scope in Natural Language Text. *Proceedings of the Fourteenth Conference on Computational Natural Language Learning* (pp. 13–17). Uppsala, Sweden: Association for Computational Linguistics.

Qi Zhao, Chengjie Sun, Bingquan Liu, and Yong Cheng. 2010. Learning to Detect Hedges and their Scope Using CRF. *Proceedings of the Fourteenth Conference on Computational Natural Language Learning* (pp. 100–105). Uppsala, Sweden: Association for Computational Linguistics.