EPE 2017:
The Sherlock Negation Resolution Downstream Application

Emanuele Lapponi♣, Stephan Oepen ♣♠, and Lilja Øvrelid♣♠

♣ University of Oslo, Department of Informatics

♠ Center for Advanced Study at the Norwegian Academy of Science and Letters

{emanuel|oe|liljao}@ifi.uio.no

Abstract

This paper describes Sherlock, a generalized update to one of the top-performing systems in the *SEM 2012 shared task on Negation Resolution. The system and the original negation annotations have been adapted to work across different segmentation and morpho-syntactic analysis schemes, making Sherlock suitable to study the downstream effects of different approaches to pre-processing and grammatical analysis on negation resolution.

1 Introduction & Motivation

Negation Resolution (NR) is the task of determining, for a given sentence, which part of the linguistic signal is affected by a negation cue. The 2012 shared task at the First Joint Conference on Lexical and Computational Semantics (*SEM) is a notable effort in NR research (Morante and Blanco, 2012), providing the field with a sizable human-annotated corpus for negation (the first outside the biomedical domain), a standardized set of evaluation metrics, as well as empirical NR results from eight competing teams. Our NR system, Sherlock (Lapponi et al., 2012b), ranked first and second in the open and closed tracks, respectively. It has later been used as a pre-processor for Sentiment Analysis (Lapponi et al., 2012a) and, due to its reliance on dependency-based features, as a means of evaluating different dependency representations extrinsically (Elming et al., 2013; Ivanova et al., 2013).

These latter efforts served as an inspiration for the 2017 shared task on Extrinsic Parser Evaluation (EPE 2017; Oepen et al., 2017). Here, participants are invited to provide fully pre-processed and syntactically parsed inputs to three dowstream systems addressing different tasks: biological event extraction (Björne et al., 2017) and fine-grained opinion analysis (Johansson, 2017), in addition to NR. Although Sherlock and the *SEM 2012 negation data have already been used for extrinsic dependency parsing evaluation, the novelty of the current work lies in the fact that the aforementioned earlier work assumed dependency graphs obtained over uniform, gold-standard sentence and token boundaries, as defined by the original token-level annotations of Morante and Daelemans (2012). In contrast, for use of Sherlock in conjunction with a diverse range of parsers that each start from 'raw', unsegmented text, the NR set-up had to be generalized to allow 'projection' of the original, token-level annotations to variable segmentations, both during training and evaluation. In the remainder of this paper we will provide an overview of the task of NR as defined by the annotations in the *SEM 2012 negation data, describe the process of generalizing the gold-standard negation annotations to arbitrary character spans, summarize the generalized Sherlock pipeline, and discuss the EPE 2017 end-to-end results for negation resolution.

2 The Conan Doyle Data

The *SEM 2012 negation data annotate a collection of fiction works by Sir Arthur Conan Doyle (Morante and Daelemans, 2012), henceforth CD. The CD data is comprised of the following annotated stories: a training set of 3644 sentences drawn from The Hound of the Baskervilles, a development set of 787 sentences taken from Wisteria Lodge, and a held-out evaluation set of 1089 sentences from The Cardboard Box and The Red Circle.

The negation annotations in these sets are comprised of so-called negation *cues* (linguistic signals of negation), which can be either full tokens (e.g. *not* or *without*) or sub-tokens (*un* in *unfortunate* or *n't* in contracted negations like *can't*); for each cue,

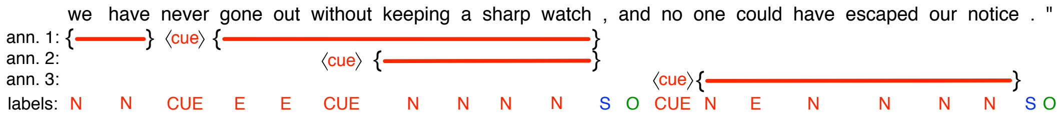
we have never gone out without keeping a sharp watch , and no one could have escaped our notice . "
ann. 1:
ann. 2:
ann. 3:
⟨cue⟩
⟨cue⟩
⟨cue⟩
labels: N N CUE E E CUE N N N N S O CUE N E N N N N S O

Figure 1: An example of how overlapping CD scope annotations are converted to flat sequences of labels. In this example, an in-scope token is labeled with N, a cue with CUE, a negated event with E, a negation stop with S, and an out-of-scope token with O.

the annotations further comprise its *scope*, i.e. the (sub-)tokens that are affected by the negation. Additionally, in-scope tokens are marked as *negated events* or *states*, provided that the sentence in question is factual, and the events in question did not take place. Consider the two following examples from the data, where cues are shown in angle brackets, in-scope tokens in braces, and negated events are underlined:

(1) Since {we have been so} ⟨un⟩{fortunate as to miss him} [. . .]

(2) If {he was} in the hospital and yet ⟨not⟩ {on the staff} he could only have been a house-surgeon or a house-physician: little more than a senior student.

Notice that negation scopes extend to full propositions (the prefix *un* in example (1) negates that *we have been so fortunate as to miss him*), and that example (2) annotates no negated event, since the sentence is non-factual. Scopes may further be discontinuous, as in example (2). Oftentimes there can be multiple instances of negation within one sentence, and their respective scopes may overlap or nest within each other.

3 Annotation Projection

One generalization that had to be made to Sherlock for use in the EPE 2017 shared task is related to segmentation into ‘sentences’ and tokens. The original *SEM 2012 negation data is annotated in a token-oriented format, inspired by a series of shared tasks at the conferences for Computational Natural Language Learning (CoNLL), where basic units of annotation are tokens—one per line, in a plain text file, with annotations separated from surface tokens by tabulator characters. Conversely, the EPE 2017 task design starts from ‘raw’ running text, i.e. participating parsers are expected to apply their own sentence splitting and tokenization. Thus, no specific segmentation conventions are imposed on parser outputs.

In order to use the *SEM 2012 negation data over arbitrary and diverse base segmentations, we developed a separate ‘projection’ step that (a) converts the gold-standard negation annotations into character-level (stand-off) spans, (b) projects these spans onto a dependency graph provided by a participating parser, and (c) serializes the enriched graph in the token-oriented *SEM 2012 file format, for Sherlock training and evaluation. In other words, annotation projection creates a ‘personalized’ version of the negation annotations for each individual segmentation, i.e. each distinct parser output. Annotation projection crucially depends on accurate character-level stand-off pointers into the underlying ‘raw’ document. As these were not available for the original *SEM 2012 negation data, we adapted the alignment tool of Dridan and Oepen (2013) to determine the correspondences from surface tokens in the annotations to sub-strings of the original documents.[1]

Conceptually, annotation projection is fairly straightforward: The *SEM 2012 negation annotations include both sub-token and multi-token negation cues and scopes, for example the prefix *un* or the multi-word *by no means*. Projection of negation annotations onto a different segmentation (with fewer, additional, or just different sentence and token boundaries) may thus move some negation instances into or out of the sub-token and multi-token categories, but both types are treated transparently in Sherlock as well as in the official *SEM 2012 scorer. In principle, we could evaluate final negation predictions (by Sherlock, for a specific parser) against the gold-standard segmentation, by applying a ‘reverse’ projection from the enriched dependency graph. However, for practical simplicity we opt to evaluate on the ‘native’ segmentation of the parser directly, i.e. invoke the *SEM 2012 negation scorer on the projected, ‘personalized’

[1] The alignment tool applies dynamic programming to compute the globally optimal solution, using the Needleman–Wunsch algorithm, taking into account common normalizations applied during tokenization, e.g. conversion from multi-character ASCII sequences for different-length dashes or various quote marks to corresponding Unicode glyphs.

Features	bigram	trigram	+token	+lemma
token	•	•		
lemma				
pos-tag	•	•	•	
first-order dependency pos-tag				
second-order dependency pos-tag				
dependency relation				
right token distance from cue				
left token distance from cue				
dependency distance from cue				
dependency path from cue				•

Table 1: Features used to train the conditional random field models (on the left), combined with token/lemma, bigram, and trigram features as indicated by the dots. Both bigram and trigram features include backward (e.g. $w_i \wedge w_{i-1}$) and forward variants ($w_i \wedge w_{i+1}$).

gold standard and the actual system output. To ensure that results are comparable across different parsers, we have confirmed that the counts of negation instances remain unaffected, so as to guard against the theoretical possibility of spurious sentence boundaries separating (parts of) a negation cue from (parts of) its arguments.

4 System Description

The task of Negation Resolution, in the context of the CD annotations, is comprised of three subtasks: negation cue identification, scope resolution, and negated event resolution. Sherlock tackles the two latter tasks (assuming that cue identification is either provided by a separate module or accepting gold-standard cues in its input), and basically looks at NR as a classical sequence labeling problem. The main component in the Sherlock pipeline, hence, is Wapiti (Lavergne et al., 2010), an open-source implementation of a Conditional Random Field (CRF) classifier, a discriminative model for sequence labeling.

The token-wise annotations in CD contain multiple layers of information. Tokens may or may not be negation cues and they can be either in- or out-of-scope; in-scope tokens may or may not be negated events, and are associated with each of the cues they are negated by. Moreover, scopes may be (partially) overlapping, as in Figure 1, where the scope of *without* is contained within the scope of *never*.

Before presenting the CRF with the annotations, Sherlock flattens the scopes, converting the CD representation internally by assigning one of six labels to each token: out-of-scope, cue, substring cue, in-scope, event, and negation stop (defined as the first out-of-scope token after a sequence of in-scope tokens), as shown in the final row of Figure 1. Using a fine-grained set of labels (rather than a minimal one, with only out-of-scope, in-scope and event labels) has been shown to yield better performance in this task (Lapponi, 2012). The models for events and in-scope tokens are trained separately; in the event model all N-labeled tokens in Figure 1 have an O label, and all E-labeled tokens in the scope model have an N label.

The features used in the CRF model are listed in Table 1.[2] By default, Sherlock utilizes the same feature set used by Lapponi et al. (2012b) (albeit without the constituents available in the original data), and runs Wapiti with default settings. Sherlock was originally developed to deal with fully connected, single headed dependency trees, and it was updated to be robust to the wider range of dependency graphs submitted to the EPE shared task. The *dependency relation* feature now records the full set of relations for a token (so if token *x* is both *y*'s *a* and *z*'s *b*, its dependency relation feature would be *a,b*). The *dependency distance* and *path from cue* features now assume graphs with (possible) re-entrancies and unconnected nodes, and only record one of possibly several equally shortest paths. If a path from a token to a cue is not found, we simply record a -1 feature.

[2] Wapiti comes with a built-in pattern-based feature expansion system. The patterns used for EPE Sherlock runs are available at `https://github.com/ltgoslo/sherlock/tree/master/patterns`

	UiO_2	Elming et al.	Stanford–Paris #6	Szeged #0	Paris–Stanford #7
ST	85.75	—	**88.57**	86.64	88.19
SM	80.00	**81.27**	80.43	78.42	80.14
ET	**80.55**	76.19	76.55	75.47	71.77
FN	66.41	**67.94**	65.37	62.15	60.48

Table 2: Results of the top-three performers at EPE 2017 (across all tasks), compared to the original UiO_2 submission to *SEM 2012 and the best-performing configuration of Elming et al. (2013).

After classification, the full (overlapping) annotations are reconstructed using a set of post-processing heuristics. It is important to note that one of these heuristics in previous Sherlock builds took advantage of the original annotations directly to help with factuality detection; when a token classified as a negated event appeared within a certain range of a token tagged as a modal (the *MD* tag), its label was changed from negated event to in-scope. This post-processing step has been removed in order to accommodate arbitrary tag sets. The remaining post-processing steps remain unchanged from (Lapponi et al., 2012b). In short, we (1) scan negation cues from left to right; (2) if b is found to the left of a within a fixed-size window, with no punctuation or S-labeled tokens in between, mark it as negated by a; (3) assign all N-negated tokens to the closest cue (again, breaking at punctuation and S-labels); (4) if cue a negates b, assign all of its N-labeled tokens to a as well. The current, EPE-ready release of Sherlock is open source and available for public download.[3]

5 EPE Shared Task Results in Context

Sherlock runs for the EPE shared tasks are evaluated on a subset of the original *SEM evaluation metrics: scope tokens (ST), scope match (SM), event tokens (ET), and full negation (FN) F_1 scores. ST and ET are token-level scores for in-scope and negated event tokens, respectively, where a true positive is a correctly retrieved token instance of the relevant class. The remaining measures are stricter, counting true positives as perfectly matched full scopes (SM), and requiring both a perfect scope and event match in the strictest 'full negation' (FN) metric. For the purpose of ranking participating submissions, the EPE 2017 shared task considered the FN metric as primary.

One important difference between previously published Sherlock results is that EPE runs on the held-out data set rely on gold-standard rather than predicted cues, making it hard to relate evaluation results directly. Table 2 shows development set F_1 results from the original *SEM shared task runs (here called UiO_2, the name of the original system), the best configuration from Elming et al. (2013), and the top three overall EPE submissions; Table 3 shows the full batch of F_1 scores for all teams and runs, for both the CD development and evaluation sets.

Unlike the EPE runs, UiO_2 and Elming et al. (2013) in Table 2 share the same set of pre-processors, and differ only in terms of dependency graphs. The former parses the data using the default MaltParser English model (Nivre et al., 2007), while the latter uses the Mate parser (Bohnet, 2010) converting the resulting phrase-structure trees into dependencies using the Yamada-Matsumoto conversion scheme. Both parsers are trained on Sections 2–21 of the Wall Street Journal portion of the venerable Penn Treebank; additionally, the MaltParser English model is augmented with data from Question Bank.

In-depth analysis and discussion of the EPE shared task results is an ongoing (and daunting) task. It is important to take into consideration that the system was designed and tuned around the original set of sentences, tokens, lemmas, tags, and their conversion to 'basic' Stanford Dependencies from the PTB-style constituent trees in the original CD data. This means that features, label sets, and heuristics were tested (and discarded) empirically, considering the Stanford scheme for syntactic dependency trees. In the extreme, 'chasing' the best possible results in the EPE context would mean repeating a similar process of feature engineering for each submission. With that in mind, simply 'plugging in' a new set of pre-processing annotations nevertheless yields better ST and SM performance than the original system (as shown in

[3] https://github.com/ltgoslo/sherlock.

		Development Set				Evaluation Set			
Team	Run	SM	ST	ET	FN	SM	ST	ET	FN
ECNU	0	80.85	89.10	73.83	62.69	80.10	88.78	66.87	62.33
	1	79.57	87.98	76.63	63.78	80.10	89.14	66.25	62.33
	2	80.00	89.36	73.58	62.69	80.38	88.37	68.30	62.33
	3	79.14	88.69	72.90	61.60	80.10	89.11	68.75	62.69
	4	80.43	87.77	75.96	65.37	78.35	88.28	67.69	60.89
Paris–Stanford	0	76.92	86.82	70.94	61.04	79.72	87.89	65.39	59.78
	1	78.14	87.04	73.08	59.35	78.28	87.45	63.98	58.29
	2	80.43	87.88	69.86	61.04	78.35	87.18	62.00	55.98
	3	80.43	88.24	72.30	61.04	78.64	87.88	61.69	56.75
	4	80.43	88.94	70.47	60.48	78.64	86.53	61.33	55.59
	5	78.26	85.95	68.90	57.61	79.62	87.31	62.20	55.59
	6	77.82	86.90	70.48	58.78	78.93	88.37	59.67	56.75
	7	80.14	88.19	71.77	60.48	78.35	88.42	61.44	56.36
	8	79.14	88.74	69.90	58.20	78.93	87.47	63.40	57.14
	9	78.70	87.71	70.87	59.91	78.93	88.21	63.52	55.98
	10	78.26	88.50	67.96	58.78	77.45	87.00	59.33	53.18
	11	80.43	88.87	72.12	61.04	80.95	88.61	63.79	56.75
Peking	0	80.00	88.01	75.83	63.78	79.33	88.23	67.73	60.89
	1	78.26	87.10	71.22	59.35	78.84	88.80	67.50	61.26
	2	78.26	87.36	73.27	59.35				
	3	79.57	87.37	70.64	61.04				
	4	79.29	87.05	75.60	64.31	79.43	88.42	64.99	58.67
	5	77.38	86.67	70.36	59.35	79.14	88.53	65.84	59.41
Prague	0	76.47	86.85	72.12	58.78	79.13	88.41	63.95	58.83
	1	77.82	87.94	73.93	61.60	77.86	88.16	68.50	61.62
	2	74.62	86.26	73.93	58.78	80.29	89.43	63.75	59.95
	3	77.38	87.71	71.77	59.35	78.54	88.08	64.82	59.95
	4	71.75	85.73	71.22	54.00	69.61	86.74	60.97	50.85
Stanford–Paris	0	80.85	88.23	76.28	64.85	82.08	**89.65**	69.70	65.13
	1	80.85	88.83	75.83	64.31	80.10	88.53	68.69	63.05
	2	81.27	88.34	75.36	63.78	80.38	88.92	68.32	63.75
	3	79.57	88.18	74.88	62.69	81.52	89.56	67.69	64.10
	4	78.70	87.30	75.60	61.60	79.52	88.73	69.38	63.05
	5	80.43	88.95	75.93	63.78	80.67	88.70	**70.34**	64.80
	6	80.43	88.57	76.55	65.37	**82.63**	89.11	**70.34**	**66.16**
	7	80.43	89.93	76.19	62.69	81.23	88.92	68.52	63.75
	8	80.43	89.18	75.00	61.60	82.35	89.71	69.75	64.45
	9	80.00	88.72	74.64	62.15	79.52	89.11	67.29	61.62
	10	**82.10**	**89.99**	77.21	**65.89**	81.80	89.13	70.34	65.13
Szeged	0	78.42	86.64	75.47	62.15	80.00	89.17	67.90	61.98
	1	77.98	87.28	76.78	63.24	79.14	88.19	67.71	60.53
	2	77.98	87.38	72.90	59.91	81.14	89.27	65.20	61.26
	3	78.86	87.07	76.14	63.78	80.38	88.75	64.05	59.78
	4	77.98	85.97	74.26	62.15	79.72	88.91	63.52	59.05
UPF	0	77.38	86.59	73.36	62.69	79.14	88.68	66.66	59.78
	1	44.35	71.09	61.63	32.85	42.46	73.70	53.04	33.34
	2	39.07	67.65	58.33	26.13	38.75	71.16	52.81	30.67
UW	0	76.47	85.79	**77.67**	62.15	77.67	86.99	63.72	56.75

Table 3: Final F_1 scores for all Sherlock runs submitted by the eight participating teams.

Table 2, Stanford–Paris run #6 compared to UiO_2; recall that negated event resolution in the original system was aided by ad-hoc heuristics on the CD tags), which is an encouraging point of departure for further analysis and comparison of the wealth of pre-processing and parsing approaches provided by the EPE shared task.

6 Conclusion & Outlook

In this paper we presented Sherlock, an updated version of one of the top-performing systems in the 2012 *SEM shared task on Negation Resolution. The system was augmented to accept arbitrary tokenization and dependency graphs, and serves as one of three extrinsic evaluators in the EPE 2017 shared task. More in-depth discussion and analysis across different downstream applications is ongoing work; for future work we would like to conduct both quantitative and qualitative error analysis, grounded in a contrastive analysis of which negation instances are comparatively easy or difficult for a majority of systems. Furthermore, we plan to re-tune and calibrate the system around a subset of the EPE submissions, attempting to make the most of the individual strengths of the different segmentations and morpho-syntactic analysis approaches.

Acknowledgments

The development of Sherlock and adaptation for the EPE 2017 shared task was in part funded by the Nordic e-Infrastructure Collaboration (NeIC) through their support to the Nordic Language Processing Laboratory (NLPL; http://www.nlpl.eu). We are grateful to our NLPL colleagues and to the Nordic tax payers.

References

Jari Björne, Filip Ginter, and Tapio Salakoski. 2017. EPE 2017: The Biomedical event extraction downstream application. In *Proceedings of the 2017 Shared Task on Extrinsic Parser Evaluation at the Fourth International Conference on Dependency Linguistics and the 15th International Conference on Parsing Technologies*. Pisa, Italy, page 13 – 20.

Bernd Bohnet. 2010. Top accuracy and fast dependency parsing is not a contradiction. In *Proceedings of the 23rd International Conference on Computational Linguistics*. Beijing, China, page 89 – 97.

Rebecca Dridan and Stephan Oepen. 2013. Document parsing. Towards realistic syntactic analysis. In *Proceedings of the 13th International Conference on Parsing Technologies*. Nara, Japan.

Jacob Elming, Anders Johannsen, Sigrid Klerke, Emanuele Lapponi, Hector Martinez, and Anders Søgaard. 2013. Down-stream effects of tree-to-dependency conversions. In *Proceedings of Human Language Technologies: The 2013 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Atlanta, GA, USA, page 617 – 626.

Angelina Ivanova, Stephan Oepen, Rebecca Dridan, Dan Flickinger, and Lilja Øvrelid. 2013. On different approaches to syntactic analysis into bi-lexical dependencies. An empirical comparison of direct, PCFG-based, and HPSG-based parsers. In *Proceedings of the 13th International Conference on Parsing Technologies*. Nara, Japan, page 63 – 72.

Richard Johansson. 2017. EPE 2017: The Trento–Gothenburg opinion extraction system. In *Proceedings of the 2017 Shared Task on Extrinsic Parser Evaluation at the Fourth International Conference on Dependency Linguistics and the 15th International Conference on Parsing Technologies*. Pisa, Italy, page 27 – 35.

Emanuele Lapponi. 2012. *Why Not! Sequence Labeling the Scope of Negation Using Dependency Features*. Master's thesis, University of Oslo, Oslo, Norway.

Emanuele Lapponi, Jonathon Read, and Lilja Øvrelid. 2012a. Representing and resolving negation for sentiment analysis. In *Proceedings of the 2012 ICDM Workshop on Sentiment Elicitation from Natural Text for Information Retrieval and Extraction*. Brussels, Belgium.

Emanuele Lapponi, Erik Velldal, Lilja Øvrelid, and Jonathon Read. 2012b. UiO2: sequence-labeling negation using dependency features. In *Proceedings of the 1st Joint Conference on Lexical and Computational Semantics*. Montréal, Canada, page 319 – 327.

Thomas Lavergne, Olivier Cappé, and François Yvon. 2010. Practical very large scale CRFs. In *Proceedings of the 48th Meeting of the Association for Computational Linguistics*. Uppsala, Sweden, page 504 – 513.

Roser Morante and Eduardo Blanco. 2012. *SEM 2012 Shared Task. Resolving the scope and focus of negation. In *Proceedings of the 1st Joint Conference on Lexical and Computational Semantics*. Montréal, Canada, page 265 – 274.

Roser Morante and Walter Daelemans. 2012. ConanDoyle-neg. Annotation of negation in Conan Doyle stories. In *Proceedings of the 8th International Conference on Language Resources and Evaluation*. Istanbul, Turkey.

Joakim Nivre, Johan Hall, Jens Nilsson, Atanas Chanev, Gülşen Eryiğıt, Sandra Kübler, Svetoslav Marinov, and Erwin Marsi. 2007. MaltParser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering* 13(2).

Stephan Oepen, Lilja Øvrelid, Jari Björne, Richard Johansson, Emanuele Lapponi, Ginter Filip, and Erik Velldal. 2017. The 2017 Shared Task on Extrinsic Parser Evaluation. Towards a reusable community infrastructure. In *Proceedings of the 2017 Shared Task on Extrinsic Parser Evaluation at the Fourth International Conference on Dependency Linguistics and the 15th International Conference on Parsing Technologies*. Pisa, Italy, page 1 – 12.