End-to-End Negation Resolution as Graph Parsing

Robin Kurtz, Stephan Oepen, & Marco Kuhlmann

Linköping University Department of Computer and Information Science
University of Oslo, Department of Informatics
robin.kurtz@liu.se, oe@ifi.uio.no, marco.kuhlmann@liu.se

Abstract

We present a neural end-to-end architecture for negation resolution based on a formulation of the task as a graph parsing problem. Our approach allows for the straightforward inclusion of many types of graph-structured features without the need for representation-specific heuristics. In our experiments, we specifically gauge the usefulness of syntactic information for negation resolution. Despite the conceptual simplicity of our architecture, we achieve state-of-the-art results on the Conan Doyle benchmark dataset, including a new top result for our best model.

1 Introduction

Negation resolution (NR), the task of detecting negation and determining its scope, is relevant for a large number of applications in natural language processing, and has been the subject of several contrastive research efforts (Morante and Blanco, 2012; Oepen et al., 2017; Fares et al., 2018). In this paper we cast NR as a graph parsing problem. More specifically, we represent negation *cues* and corresponding *scopes* as a bi-lexical graph and learn to predict this graph from the tokens. Under this representation, we may apply any dependency graph parser to the task of negation resolution. The specific parsing architecture that we use in this paper extends that of Dozat and Manning (2018).

Contributions This work (a) rationally reconstructs the previous state of the art in negation resolution; (b) develops a novel approach to the problem based on general graph parsing techniques; (c) proposes and evaluates different ways of integrating 'external' grammatical information; (d) gauges the utility of morpho-syntactic preprocessing at different levels of accuracy; (e) shifts experimental focus (back) to a complete, end-to-end perspective on the task; and (f) reflects on uncertainty in judging experimental findings, including thourough significance testing.

Paper Structure In the following Section 2, we review selected related work on negation resolution. Section 3 describes the specific NR task that we address in this paper. In Section 4 we present our new encoding of negations and our parsing model, followed by the description of our experiments and results in Section 5. We discuss these results in Section 6 and summarize our findings in Section 7.

2 Related Work

While there exist a variety of datasets that annotate negation (Jiménez-Zafra et al., 2020), the BioScope (Szarvas et al., 2008) and Conan Doyle datasets (ConanDoyle-neg; Morante and Daelemans, 2012) are most commonly used for evaluation. The latter was created for the shared task at *SEM 2012 (Morante and Blanco, 2012), where competing systems needed to predict both negation *cues* (linguistic expressions of negation) and their corresponding *scopes*, i.e. the part of the utterance being negated. Cues can be simple negation markers (such as *not* or *without*), but may also consist of multiple words (i.e. *neither* ... *nor*), or be mere affixes (i.e. *in*frequent or clue*less*). In contrast to other datasets, ConanDoyle-neg also annotates negated *events* that are part of the scopes.

The analysis of negation is divided into two related sub-tasks, *cue detection* and *scope resolution*. While cue detection is mostly dependent on lexical or morphological features, relating cues to scopes is a structured prediction problem and will likely benefit from an analysis of morpho-syntactic or surface-semantic properties. The UiO_2 system SHERLOCK (Lapponi et al., 2012), the winner of the open track of the *SEM 2012 shared task, uses morpho-syntactic parts of speech and syntactic dependencies to classify tokens as either

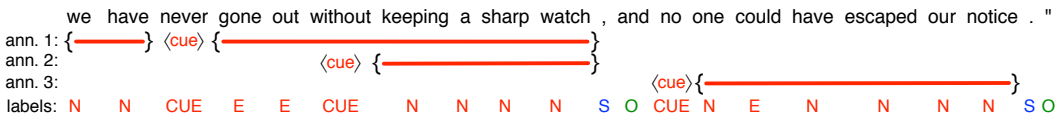

Figure 1: An example of how overlapping ConanDoyle-neg annotations are converted to flat sequences of labels in SHERLOCK. In this example, an in-scope token is labeled with N, a cue with CUE, a negated event with E, a negation stop with S, and an out-of-scope token with O. Illustration taken from Lapponi et al. (2017).

in-scope or out-of-scope using a conditional random field (CRF). Another CRF further classifies scope tokens as events, and a heuristic is applied to distribute scope tokens to their respective cues. The SHERLOCK system was subsequently used by Elming et al. (2013) to evaluate various dependency conversions, and similarly served as one of three reference 'downstream' applications in the 2017 Extrinsic Parser Evaluation initiative (EPE; Oepen et al., 2017). The best results from this evaluation define the state of the art in NR.

Deviating from the original *SEM 2012 setup, Packard et al. (2014) simplified the task to only evaluate the performance on finding scope tokens (assuming gold-standard information about negation cues). Fancellu et al. (2016, 2018) continued this trend, additionally treating each negation instance separately, and successfully used BiLSTM (bidirectional long-short term memory recurrent neural networks; Hochreiter and Schmidhuber, 1997). Recently, Sergeeva et al. (2019) used pretrained transformers (Vaswani et al., 2017), namely BERT (Devlin et al., 2019), to further improve performance, albeit on a derivative of the original dataset (Liu et al., 2018). The 2018 follow-up to the EPE shared task (Fares et al., 2018) again used SHERLOCK to evaluate parsing performance, this time restricting itself to participating systems in the co-located 2018 CoNLL Shared Task on Universal Dependency Parsing (Zeman et al., 2018).

3 Task and Data

We target the original *SEM 2012 shared task and aim to predict both negation cues and their scopes. We compare our approach with the baseline SHERLOCK system and the state-of-the-art systems identified through the EPE shared tasks.

3.1 Data

The negation data of *SEM 2012 consists of selected Sherlock Holmes stories from the works of Arthur Conan Doyle, and contains 3,644 sentences in the training set, 787 sentences in the development set, and 1,089 sentences in the evaluation set. The corpus annotates a total of 1,420 instances of negation. Several sentences contain two or more instances of negation, while 4,294 sentences do not contain any at all.

Negation instances are annotated as tri-partite structures: Negation *cues* can be full tokens, multi-word expressions, or affixal sub-tokens. For each cue, its *scope* is defined as the possibly discontinuous sequence of (sub-)tokens affected by the negation. Additionally, a subset of in-scope tokens can be marked as negated *events* (or *states*), provided that the sentence is factual and the events in question did not take place. For sentences containing multiple negation instances, their respective scope and event spans may nest or overlap.

The systems submitted to the EPE 2017 and 2018 tasks work on 'raw', unsegmented text, and apply different segmentation strategies. To evaluate these systems in the context of negation resolution, the gold-standard negation annotations have to be retrofitted to each system's output. Each system is then tested against their own 'personalized' gold standard sets. For more information on this projection procedure, we refer to Lapponi et al. (2017).

3.2 Baseline System

As in the EPE shared tasks, our baseline is the SHERLOCK system of Lapponi et al. (2012, 2017), which approaches NR as a token-based sequence labeling problem and uses a Conditional Random Field (CRF) classifier (Lavergne et al., 2010). The token-wise negation annotations contain multiple layers of information. Tokens may or may not be negation cues; they can be in or out of scope for a specific cue; in-scope tokens may or may not be negated events. Moreover, as already stated, multiple negation instances may be (partially or fully) overlapping. Before presenting the CRF with the annotations, SHERLOCK 'flattens' all negation instances in a sentence, assigning a six-valued extended begin–inside–outside labeling scheme, as indicated in Figure 1. After classification, hierar-

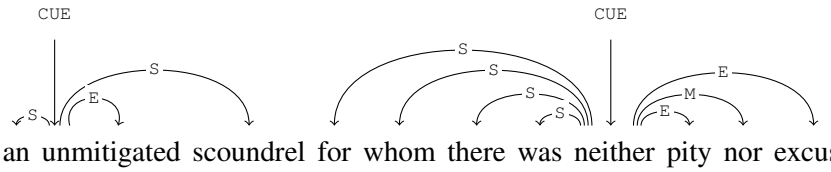

Figure 2: An example of how ConanDoyle-neg annotations are converted to a labeled dependency graph structure. We omit the special root node r_0 and mark roots instead with vertical arcs. The arcs are labeled for scope (S), event (E), and multi-word-cue (M).

chical (overlapping) negation structures are reconstructed using a set of post-processing heuristics.

The features of the classifier include different combinations of token-level observations, such as surface forms, part-of-speech tags, lemmas, and dependency labels. In addition, SHERLOCK employs features encoding both token and dependency distance to the nearest cue, together with the full shortest dependency paths. In the EPE context, gold-standard negation cues were provided as input to SHERLOCK.[1]

4 Approach

In this section we define our graph-based encoding of negation structures, and present our parsing system and training procedure.

4.1 Negation Graphs

Instead of labeling each token sequentially with cue, scope or event markers, we reformulate NR as a parsing task, creating dependency-style *negation graphs* with lexicalized nodes and bilexical arcs $i \rightarrow j$ between a *head* i and a *dependent* j as target structures. This allows us to more naturally encode the relationship between tokens and their cue(s), while being able to easily differentiate between regular scopes and events.

An example for a *negation graph* is shown in Figure 2. We adopt a convention from dependency parsing and visualize negation graphs with their nodes laid out as the words of the respective sentence, and their arcs drawn above the nodes. When transforming negation annotations into graphs, we mark negation cues i by special arcs $r_0 \rightarrow i$ emanating from an artificial root node r_0. Scope and event tokens are marked by appropriately labeled arcs from their respective cue(s). For multi-word cues, only the first cue token is assigned as a root, while the remaining tokens are connected to the first with arcs labeled *mwc*. Since we do not split tokens into subtokens, we mark the full token containing an affixal cue as root. The negated part of the token is (by convention) annotated as an event, and thus marked by an appropriately labeled loop.

The resulting graphs thus contain unconnected nodes, multiple structural roots (dependents of the artificial root node r_0), loops, and nodes with multiple incoming arcs. Sentences that do not contain any negations are represented by empty graphs.

4.2 Neural Model

With the translation of the negation annotation into graphs, we can use parsers that learn how to jointly predict cues and their respective scopes, avoiding a cascade of classifiers and heuristics as in the SHERLOCK system. Specifically, we use a reimplementation of the neural parser by Dozat and Manning (2018), which in turn is based on (Kiperwasser and Goldberg, 2016). At the heart of this parser is a bidirectional recurrent neural network with Long Short-Term Memory cells (BiLSTM; Hochreiter and Schmidhuber, 1997). Given a sequence of word embeddings $\boldsymbol{w}_i$ that correspond to the input sequence $\boldsymbol{x} = x_1, \ldots, x_n$, the network outputs a sequence of context-dependent embeddings $\boldsymbol{c}_i$:

$$\boldsymbol{c}_1, \ldots, \boldsymbol{c}_n = \text{BiLSTM}(\boldsymbol{w}_1, \ldots, \boldsymbol{w}_n)$$

We augment the input word embeddings with additional part-of-speech tag and lemma embeddings, embeddings created by a character-based LSTM, and 100-dimensional GloVe (Pennington et al., 2014) embeddings. Two feedforward neural networks (FNN) create specialized representations of each word as a potential head and dependent.

$$\boldsymbol{h}_i = \text{FNN}_h(\boldsymbol{c}_i) \qquad \boldsymbol{d}_i = \text{FNN}_d(\boldsymbol{c}_i)$$

[1] One main focus of the EPE task was the downstream evaluation of different syntactic representations; but the subtask of cue detection is relatively insensitive to grammatical structure (Velldal et al., 2012).

These new representations are then scored via a bilinear model with weight tensor $\boldsymbol{U}$:

$$\text{score}(\boldsymbol{h}_i, \boldsymbol{d}_j) = \boldsymbol{h}_i^\top \boldsymbol{U} \boldsymbol{d}_j$$

The inner dimension of the tensor $\boldsymbol{U}$ corresponds to the number of labels plus a special label indicating the absence of an arc (NONE), and thus predicts arcs and labels jointly.

4.3 Adding External Graph Features

Similarly to SHERLOCK, our neural model is able to process external morpho-syntactic or surface-semantic analyses of the input sentence in the form of dependency graphs. Inspired by Kurtz et al. (2019), we extend the contextualized embeddings that are computed by our parser by information derived from the external graph. For this we use three approaches: (i) attaching the sum of heads; (ii) scaled attention on the heads; and (iii) Graph Convolutional Networks (Kipf and Welling, 2017). In the following, we view the external graph in terms of its $n \times n$ adjacency matrix $\boldsymbol{A}$ and the contextualized embeddings as an $n \times d$ matrix $\boldsymbol{C}$.

Sum of Heads The first method generalizes that of Kurtz et al. (2019), who concatenate to each contextualized embedding the contextualized embedding of its head. This only works when the graphs are trees, that is, when every node has one incoming arc. When there is more than one incoming arc, we instead sum up all respective contextual embeddings. We express this as a matrix product

$$\text{sumoh}(\boldsymbol{A}, \boldsymbol{C}) = \boldsymbol{A}\boldsymbol{C}\,.$$

Scaled Attention The second approach is inspired by Vaswani et al. (2017), who compute the (scaled) dot product attention $\boldsymbol{Q}\boldsymbol{K}^\intercal$ between a matrix of queries $\boldsymbol{Q}$ and a matrix of keys $\boldsymbol{K}$, and normalize it by a row-wise softmax function, which yields probabilistic weights on potential values. Noting the similarity between this normalized attention matrix and a probabilistic adjacency matrix, we replace $\boldsymbol{Q}\boldsymbol{K}^\intercal$ with the matrix $\boldsymbol{A}$:

$$\text{scatt}(\boldsymbol{A}, \boldsymbol{C}) = \text{softmax}\left(\frac{\boldsymbol{A}}{\sqrt{d}}\right)\boldsymbol{C}$$

Here, d is the size of the contextualized embeddings. In our case, where we merely want extract features from a given graph, the matrix $\boldsymbol{A}$ is known and sparse; but the same scaled attention model could also be used also in a multi-task setup to jointly learn to parse syntactico-semantic graphs and negations, in which case $\boldsymbol{A}$ would be learned and dense.

Graph Convolutional Networks Graph Convolutional Networks (GCNs; Kipf and Welling, 2017) generalize convolutional networks to graph-structured data. While they were developed with graphs much larger than our negation graphs in mind, Marcheggiani and Titov (2017) showed their usefulness for semantic role labeling. With $\boldsymbol{X}^0 = \boldsymbol{C}$ at the first level, we compute, for each level $l > 0$, a combined representation of parents (P), children (C) and the nodes themselves (S), weighted by layer-specific weight matrices $\boldsymbol{W}^l$:

$$\boldsymbol{X}^l = \text{ReLU}\Big(\big(\boldsymbol{A}\boldsymbol{W}_P^l + \boldsymbol{A}^\intercal\boldsymbol{W}_C^l + \boldsymbol{W}_S^l\big)\boldsymbol{X}^{l-1}\Big)$$

When applying the next layer l, each node is updated with respect to its representation $\boldsymbol{X}^{l-1}$ from the previous layer, thus taking grandparents and grandchildren indirectly into account. As this method is the only one that not only uses a node's head but also its children, we expect it to benefit the most from external graph features.

5 Experiments

In this section we describe our experiments, shortly reviewing our baselines, methodology, and reported results.

5.1 Training

Our parser is trained with a softmax cross-entropy loss using the Adam optimizer (Kingma and Ba, 2015) and mini-batching. The training objective for our negation parsing system does not directly match the official evaluation measures, but is instead based on labeled per-arc F_1 scores (i.e. the harmonic mean of precision and recall), which measures the amount of (in)correctly predicted arcs and labels. For model selection, we train for 200 epochs and choose the model instance that performs best on the development set.

Our network sizes, dropout rates, and training parameters are shown in Table 1. Even though our model only has less than half as many trainable parameters than the original model by Dozat and Manning (2018), it is still prone to overfitting, partly because of the rather small size of the training data. We thus apply only marginally smaller dropout rates than Dozat and Manning (2018). Following Gal and Ghahramani (2016), we apply variational

Network sizes	Embeddings	100
	Char LSTM	1 @ 100
	Char embedding	80
	BiLSTM	3 @ 200
	Arc/Label FNN	200
	GCN Levels	2
Dropout rates	Embeddings	20%
	Char LSTM feedforward	30%
	Char LSTM recurrent	30%
	Char Linear	30%
	BiLSTM feedforward	40%
	BiLSTM recurrent	20%
	Arc FNN	20%
	Arc scorer	20%
	Label FNN	30%
	Label scorer	30%
Training parameters	Epochs	200
	Mini-batch size	50
	Adam β_1	0
	Adam β_2	0.95
	Learning rate	$1 \cdot 10^{-3}$
	Gradient clipping	5
	Interpolation constant	0.025
	L_2 regularization	$3 \cdot 10^{-9}$

Table 1: Network sizes, dropout rates, and training parameters of our neural models.

dropout sharing the same dropout mask between all time steps in a sequence, and DropConnect (Wan et al., 2013; Merity et al., 2017) on the hidden states of the LSTM.

5.2 Evaluation Measures

Standard evaluation measures for the original *SEM 2012 task include 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 of the relevant class (Morante and Blanco, 2012). FN is the strictest of these measures (and the primary evaluation metric for the NR part of the EPE shared task), counting as true positives only perfectly retrieved full scopes, including an exact match on negated events.

5.3 Baselines

In order to have a fair comparison with the previous results of the 2017 and 2018 EPE shared tasks, we use the same data as their respective best-performing systems. For the 2017 edition of EPE, the best-performing system, STANFORD-PARIS-06 (Schuster et al., 2017), uses enhanced Universal Dependencies (v1) and data from the Penn Treebank (Marcus et al., 1993), the Brown Corpus (Francis and Kučera, 1985) and the GENIA treebank (Tateisi et al., 2005). In contrast to this, the best performing system for the negation task in 2018, the TURKUNLP submission of Kanerva et al. (2018), only uses the English training data provided by the co-located UD parsing shared task. Both systems use the parser and hyperparameters of Dozat et al. (2017), the winning submission of the CoNLL 2017 Shared Task on parsing Universal Dependencies.

In the overview paper for the 2018 EPE shared task (Fares et al., 2018), the organizers report that the version of the SHERLOCK negation system that was used for EPE 2017 had a deficiency that could leak gold-standard scope and event annotations into system predictions, leading to potentially inflated scores.[2] The EPE 2018 version of SHERLOCK corrected this problem and added automated hyperparameter tuning, which Fares et al. (2018) suggest largely offset the negative effect on overall scores from the bug fix, at least when averaging over all submissions. They did not, however, re-run the EPE 2017 evaluation with the corrected and enhanced version of SHERLOCK, leaving substantive uncertainty about current state-of-the-art results. We address this problem by applying the improved (i.e. 2018) version of the baseline system, including the exact same tuning procedure described by Fares et al. (2018), to the originally best-performing STANFORD-PARIS dependeny graphs. In this replication study, we observe a large (5 points FN) drop in performance compared to the originally reported results. While STANFORD-PARIS still outperforms TURKUNLP, the margin between the two systems is narrowed down to less than 2 points FN.

5.4 Experiments

We report two sets of experiments. For all experiments, we run each of our neural network models 10 times with different random seeds and chose the best performing model with respect to performance on the development set in terms of FN F_1.

Gold-standard cues Even though our approach can easily predict negation cues on its own, for our first set of experiments, we follow the setup of the EPE tasks and predict only scopes and events, adding gold-standard cues as external graph features after training. Overlapping the gold-cue inputs with the additional graph inputs is not optimal but avoids adding any further complexity to the model. Similar to SHERLOCK, we handle affixal cues in post-processing, splitting and classifying

[2] This problem also applies to Elming et al. (2013).

Data	Model	Extra	Development				Evaluation			
			SM	ST	ET	FN	SM	ST	ET	FN
STANFORD-PARIS	SHERLOCK		80.43	88.82	71.64	61.60°	78.83	88.31	67.09	61.42
	sumoh	w/o syntax	78.70	86.35	73.74	69.43	78.54	89.62	62.10	62.15
		with syntax	74.62	86.86	72.3	64.85	75.19	88.74	63.87	57.68
	scatt	w/o syntax	79.57	88.86	75.92	69.93	78.24	89.35	59.74	58.45
		with syntax	76.92	87.68	70.94	68.94	77.34	88.96	63.32	62.15
	gcn	w/o syntax	76.92	87.53	77.36	68.94	77.04	88.99	66.86	61.05
		with syntax	80.00	88.84	76.85	70.89*	78.54	89.71	65.00	**64.27**
TURKUNLP	SHERLOCK		77.38	87.19	72.36	59.91°	80.48	89.36	65.36	59.74
	sumoh	w/o syntax	79.14	87.36	78.26	71.85	78.43	89.10	61.63	60.48
		with syntax	76.47	87.28	73.73	66.92	75.69	88.90	64.83	57.45
	scatt	w/o syntax	80.85	88.5	75.24	70.89	79.32	89.56	65.61	60.48
		with syntax	80.43	88.76	73.93	68.44*	78.13	89.74	66.46	**61.58**
	gcn	w/o syntax	78.26	87.31	74.75	67.94	77.23	88.89	62.50	60.85
		with syntax	78.26	88.76	76.70	69.43	76.00	88.98	64.17	58.99

Table 2: Results of our NR parser on the STANFORD-PARIS and TURKUNLP versions of the ConanDoyle-neg development and evaluation sets when gold-standard cues are provided. We compare our *gcn with syntax* model for STANFORD-PARIS and our *scatt with syntax* model for TURKUNLP with the respective SHERLOCK models using bootstrap significance testing. Only the $*$-marked measures are significantly different from their $\circ$-marked counterparts.

Data	Model	Development					Evaluation				
		CUE	SM	ST	ET	FN	CUE	SM	ST	ET	FN
Read et al. (2012)		–	–	–	–	–	91.31	70.39	82.37	67.02	57.63
Packard et al. (2014)		–	77.80	82.40	–	–	91.31	73.10	85.40	–	–
STANFORD-PARIS	no syntax	91.76	73.76	86.57	71.96	65.69	91.76°	75.81	87.69	60.66	**59.40**
	sumoh	91.62	74.10	85.63	69.43	63.39	91.62	68.64	86.66	59.74	52.58
	scatt	91.51	76.16	84.72	70.00	66.42	91.51	72.25	86.09	61.21	57.64
	gcn	93.26	73.11	84.22	72.40	65.19	93.26*	73.83	86.89	63.69	58.07
TURKUNLP	no syntax	92.98	76.22*	85.61	71.11	66.42	92.98	72.39	86.92	59.88	**55.18**
	sumoh	92.49	73.45	85.03	75.35	62.31	92.49	71.73	87.91	60.06	53.19
	scatt	92.54	76.60	85.34	71.03	64.15	92.54	71.85	87.06	57.23	52.08
	gcn	91.02	72.46°	84.90	73.49	63.15	91.02	71.43	86.57	63.40	54.54

Table 3: Results of our NR parser on the STANFORD-PARIS and TURKUNLP versions of the ConanDoyle-neg development and evaluation sets when cues are predicted. We test for significant differences between our *gcn with syntax* models for STANFORD-PARIS and TURKUNLP and respective models using no additional inputs. Only the $*$-marked measures are significantly different from their $\circ$-marked counterparts.

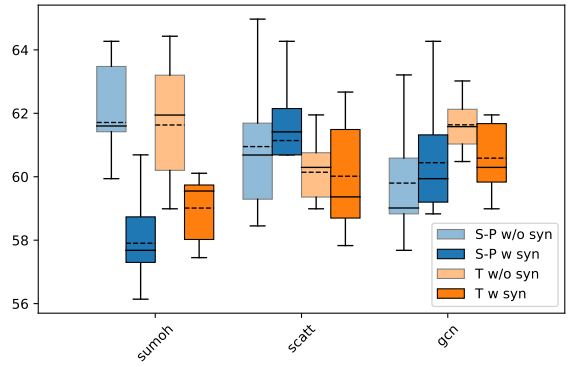

Figure 3: Boxplot visualizing the variance of performance across the three different methods using and not using additional syntactic information on the evaluation set.

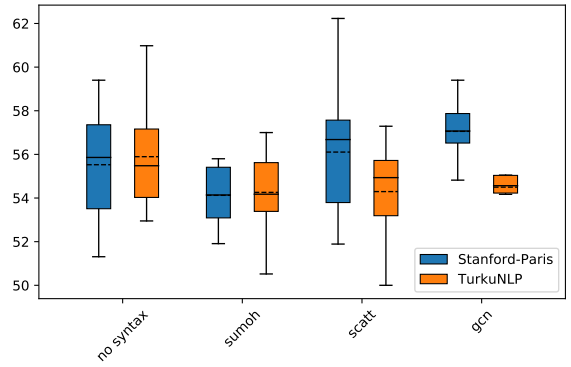

Figure 4: Boxplot visualizing the variance of performance across the four models using no syntax and using syntax with each of the three methods on the evaluation set.

five known prefixes and one suffix as cues, and the remainder as the negated event. The results for these experiments are reported in Table 2.

Predicted cues For the second set of experiments, we also predict negation cues, and additionally report the F_1 for cues (CUE). In order to put these results into perspective, we contrast them with the winning system of the *SEM 2012 shared task by Read et al. (2012), and additionally with the MRS Crawler of Packard et al. (2014). The results for these experiments are reported in Table 3.

Significance testing Given the rather small size of the dataset, we follow the advice of Dror et al. (2018) and test for signficance using the bootstrap method (Berg-Kirkpatrick et al., 2012). We compare our best-performing system for both STANFORD-PARIS and TURKUNLP with the respective SHERLOCK sytems, resampling the test sets 10^6 times and setting our threshold to 5%, following standard methodology. For the second set of experiments, where we additionally predict the negation cue, we compare our best system with our second-best performing system. We additionally visualize the variance of performance across all 10 systems on the evaluation set in Figures 3 and 4.

6 Discussion

In this section we discuss the results of our experiments and place them in the broader context of the research literature on NR.

6.1 Gold Cues

We first discuss our results when using gold-standard cues, as in the EPE tasks.

Effect of Pre-Processing Using the updated SHERLOCK system to re-evaluate the performance of the best system of the best system of the 2017 edition of the EPE shows that the previous results were in fact unattainable. While the data processed by STANFORD-PARIS enables SHERLOCK to still perform better than with TURKUNLP-processed data, their performance difference is small. Our approach outperforms SHERLOCK in both cases and also performs better with STANFORD-PARIS processed data, even when no additional syntactic input is used. The tokenization, part-of-speech tagging and lemmatization done by STANFORD-PARIS thus seem to better fit the NR task, and have likely also benefitted from the larger and more diverse data used during training.

Handling Additional Inputs The most efficient method to handle gold cues at input time, it turns out, is our simplest method, concatenating each contextual token with the sum of its heads. A likely explanation for this is that this method is able to directly read off the gold-cue information. This method however is clearly not able to handle additional syntactic inputs, motivating the use of either of the more advanced techniques. Combining STANFORD-PARIS syntactic trees with the GCN, clearly performs best here, but does not point towards a general trend; the plots in Figure 3 rather show that most of the systems perform similarly, with the exception of the *sum-of-heads* method when using additional syntactic inputs.

6.2 Predicted Cues

When we task our system to also predict cues, as in *SEM 2012, our best system outperforms Read

et al. (2012) and Packard et al. (2014) on most measures. Our neural graph parsing approch is better at identifying the relevant scope tokens (ST), due to its pairwise classification approach, which generally also results in better performance for matching complete scopes (SM). The system does however struggle with telling events and regular scopes apart, and is clearly outperformed by Read et al. (2012) on that measure (ET). Our system differentiates between scopes and events using arc labels, and might not have seen enough training data to sufficiently train the labeling part of the network.

Even though our best systems for both the STANFORD-PARIS and the TURKUNLP version of the evaluation data used additional syntactic inputs when gold-standard cues were provided, our best systems for also predicting cues do not rely on additional syntactic inputs at all.

6.3 Significant Learning

Using neural networks often comes at a price, that is paid in many different initializations. While the boxplots in Figures 3 and 4 show the same general trends as our particular systems in Tables 2 and 3, they also point out the variance of performance for each run. Choosing the final system with regards to performance on the development set may lead to state-of-the-art performance on the evaluation set with more than two points of FN F_1 difference.

This does however, not have to be the case. Our system using gold cues with scaled attention on STANFORD-PARIS for example, performs more than two points FN F_1 worse than the average on the evaluation set, while performing three points better on the development set. Good performance on the development set is not necessarily and indication for good performance on the final evaluation set. This notion is further reinforced by the lack of significant difference in performance of our best systems, compared to SHERLOCK. Even more than three points of FN F_1 do not constitue a significant difference. This ties in well with the somewhat erratic performance differences across different settings and runs.

The NLP community has recently realized the importance of proper testing in favour of simple comparisons of benchmark scores (Gorman and Bedrick, 2019). This is especially important when working with deep learning architectures, as model selection becomes more complicated (Moss et al., 2019). Particularly when working with smaller datasets, such as this one, it is important to correctly analyse the results before any conclusions about optimal new systems are drawn (Dror et al., 2017).

Recasting the negation resolution task as a graph-parsing problem, allows us to straightforwardly use a variety of existing tools. With most of these now using neural networks, we can extend them to employ massive pre-trained models such as BERT (Devlin et al., 2019) or ELMo (Peters et al., 2018). This would allow us to leverage their general power into more specific tasks that have only limited data available.

7 Conclusion

We have introduced a novel approach to negation resolution that remodels negation annotations into depedency graphs. These negation graphs more directly encode the pairwise cue-scope relationships, and thus enable our neural network to more easily learn them. We extended an already powerful neural graph-parsing approach further to additionally use arbitrary dependency graph structures as inputs. In order to validate our approach, we revisit the EPE 2017 and 2018 shared tasks and the full *SEM 2012 shared task on negation resolution, outperforming each previously best system distinctly, albeit not with statistical significance.

We believe that our approach can be used to restructure other tasks as dependency graphs in similar fashion, and thus reuse existing systems as general purpose tools.

References

Taylor Berg-Kirkpatrick, David Burkett, and Dan Klein. 2012. An Empirical Investigation of Statistical Significance in NLP. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 995–1005, Jeju Island, Korea. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Timothy Dozat and Christopher D. Manning. 2018. Simpler but more accurate semantic dependency

parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 484–490, Melbourne, Australia. Association for Computational Linguistics.

Timothy Dozat, Peng Qi, and Christopher D. Manning. 2017. Stanford's graph-based neural dependency parser at the CoNLL 2017 shared task. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 20–30, Vancouver, Canada. Association for Computational Linguistics.

Rotem Dror, Gili Baumer, Marina Bogomolov, and Roi Reichart. 2017. Replicability Analysis for Natural Language Processing: Testing Significance with Multiple Datasets. *Transactions of the Association for Computational Linguistics*, 5:471–486.

Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. 2018. The Hitchhiker's Guide to Testing Statistical Significance in Natural Language Processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1383–1392, Melbourne, Australia. Association for Computational Linguistics.

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

Federico Fancellu, Adam Lopez, and Bonnie Webber. 2016. Neural Networks For Negation Scope Detection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 495–504, Berlin, Germany. Association for Computational Linguistics.

Federico Fancellu, Adam Lopez, and Bonnie Webber. 2018. Neural Networks for Cross-lingual Negation Scope Detection. *arXiv:1810.02156 [cs]*.

Murhaf Fares, Stephan Oepen, Lilja Øvrelid, Jari Björne, and Richard Johansson. 2018. The 2018 Shared Task on Extrinsic Parser Evaluation. On the downstream utility of English Universal Dependency parsers. In *Proceedings of the 22nd Conference on Natural Language Learning*, page 22 – 33, Brussels, Belgia.

W. Nelson Francis and Henry Kučera. 1985. Frequency Analysis of English Usage: Lexicon and Grammar. *Journal of English Linguistics*, 18(1):64–70.

Yarin Gal and Zoubin Ghahramani. 2016. A Theoretically Grounded Application of Dropout in Recurrent Neural Networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pages 1027–1035, Barcelona, Spain. Curran Associates Inc.

Kyle Gorman and Steven Bedrick. 2019. We Need to Talk about Standard Splits. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2786–2791, Florence, Italy. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-term Memory. *Neural computation*, 9:1735–80.

Salud María Jiménez-Zafra, Roser Morante, María Teresa Martín-Valdivia, and L. Alfonso Ureña-López. 2020. Corpora annotated with negation: An overview. *Computational Linguistics*, 46(0):1–56. Early access.

Jenna Kanerva, Filip Ginter, Niko Miekka, Akseli Leino, and Tapio Salakoski. 2018. Turku Neural Parser Pipeline: An End-to-End System for the CoNLL 2018 Shared Task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 133–142, Brussels, Belgium. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Eliyahu Kiperwasser and Yoav Goldberg. 2016. Simple and accurate dependency parsing using bidirectional LSTM feature representations. *Transactions of the Association for Computational Linguistics*, 4:313–327.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Robin Kurtz, Daniel Roxbo, and Marco Kuhlmann. 2019. Improving Semantic Dependency Parsing with Syntactic Features. In *Proceedings of the First NLPL Workshop on Deep Learning for Natural Language Processing*, pages 12–21, Turku, Finland. Linköping University Electronic Press.

Emanuele Lapponi, Stephan Oepen, and Lilja Øvrelid. 2017. EPE 2017: The Sherlock negation resolution 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*, page 25 – 30, Pisa, Italy.

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

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*, page 504 – 513, Uppsala, Sweden.

Qianchu Liu, Federico Fancellu, and Bonnie Webber. 2018. NegPar: A parallel corpus annotated for negation. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Diego Marcheggiani and Ivan Titov. 2017. Encoding Sentences with Graph Convolutional Networks for Semantic Role Labeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1506–1515, Copenhagen, Denmark. Association for Computational Linguistics.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Stephen Merity, Nitish Shirish Keskar, and Richard Socher. 2017. Regularizing and Optimizing LSTM Language Models. *CoRR*, abs/1708.02182.

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*, page 265 – 274, Montréal, Canada.

Roser Morante and Walter Daelemans. 2012. ConanDoyle-neg: Annotation of negation cues and their scope in Conan Doyle stories. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 1563–1568, Istanbul, Turkey. European Language Resources Association (ELRA).

Henry Moss, Andrew Moore, David Leslie, and Paul Rayson. 2019. FIESTA: Fast IdEntification of State-of-The-Art models using adaptive bandit algorithms. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2920–2930, Florence, Italy. Association for Computational Linguistics.

Stephan Oepen, Lilja Øvrelid, Jari Björne, Richard Johansson, Emanuele Lapponi, Filip Ginter, 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*, page 1 – 16, Pisa, Italy.

Woodley Packard, Emily M. Bender, Jonathon Read, Stephan Oepen, and Rebecca Dridan. 2014. Simple negation scope resolution through deep parsing: A semantic solution to a semantic problem. In *Proceedings of the 52nd Meeting of the Association for Computational Linguistics*, page 69 – 78, Baltimore, MD, USA.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

Jonathon Read, Erik Velldal, Lilja Øvrelid, and Stephan Oepen. 2012. UiO1. Constituent-based discriminative ranking for negation resolution. In *Proceedings of the 1st Joint Conference on Lexical and Computational Semantics*, page 310 – 318, Montréal, Canada.

Sebastian Schuster, Éric Villemonte de la Clergerie, Marie Candito, Benoît Sagot, Christopher Manning, and Djamé Seddah. 2017. Paris and Stanford at EPE 2017: Downstream Evaluation of Graph-based Dependency Representations. In *EPE 2017 - The First Shared Task on Extrinsic Parser Evaluation*, pages 47–59.

Elena Sergeeva, Henghui Zhu, Amir Tahmasebi, and Peter Szolovits. 2019. Neural Token Representations and Negation and Speculation Scope Detection in Biomedical and General Domain Text. In *Proceedings of the Tenth International Workshop on Health Text Mining and Information Analysis (LOUHI 2019)*, pages 178–187, Hong Kong. Association for Computational Linguistics.

György Szarvas, Veronika Vincze, Richárd Farkas, and János Csirik. 2008. The BioScope corpus: Annotation for negation, uncertainty and their scope in biomedical texts. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing*, pages 38–45, Columbus, Ohio. Association for Computational Linguistics.

Yuka Tateisi, Akane Yakushiji, Tomoko Ohta, and Jun'ichi Tsujii. 2005. Syntax Annotation for the GENIA Corpus. In *Companion Volume to the Proceedings of Conference Including Posters/Demos and Tutorial Abstracts*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*, pages 5998–6008.

Erik Velldal, Lilja Øvrelid, Jonathon Read, and Stephan Oepen. 2012. Speculation and negation: Rules, rankers, and the role of syntax. *Computational Linguistics*, 38(2):369 – 410.

Li Wan, Matthew Zeiler, Sixin Zhang, Yann LeCun, and Rob Fergus. 2013. Regularization of Neural Networks Using Dropconnect. In *Proceedings*

of the 30th International Conference on International Conference on Machine Learning - Volume 28, ICML'13, pages III–1058–III–1066, Atlanta, GA, USA. JMLR.org.

Daniel Zeman, Jan Hajič, Martin Popel, Martin Potthast, Milan Straka, Filip Ginter, Joakim Nivre, and Slav Petrov. 2018. CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–21, Brussels, Belgium. Association for Computational Linguistics.