Understanding the Polarity of Events in the Biomedical Literature: Deep Learning vs. Linguistically-informed Methods

Enrique Noriega-Atala, Zhengzhong Liang,
John A. Bachman[†], Clayton T. Morrison, Mihai Surdeanu
University of Arizona, Tucson, Arizona, USA
[†]Harvard Medical School, Boston, Massachusetts, USA
{enoriega,zhengzhongliang,claytonm,msurdeanu}@email.arizona.edu
john_bachman@hms.harvard.edu

Abstract

An important task in the machine reading of biochemical events expressed in biomedical texts is correctly reading the polarity, i.e., attributing whether the biochemical event is a promotion or an inhibition. Here we present a novel dataset for studying polarity attribution accuracy. We use this dataset to train and evaluate several deep learning models for polarity identification, and compare these to a linguistically-informed model. The best performing deep learning architecture achieves 0.968 average F1 performance in a five-fold cross-validation study, a considerable improvement over the linguistically informed model average F1 of 0.862.

1 Introduction

Recent advances in information extraction (IE) have resulted in high-precision, high-throughput systems tailored to the reading of biomedical scientific publications (Valenzuela-Escárcega et al., 2018; Peng et al., 2017; Quirk and Poon, 2016; Kim et al., 2013; Björne and Salakoski, 2013; Hakala et al., 2013; Bui et al., 2013, *inter alia*). This, in turn, has resulted in the use of machine reading systems as the foundation of more complex, higher-level inference applications in specific domains such as cancer research (Valenzuela-Escárcega et al., 2018).

However, the presence of noise in pipelined systems that use IE as an initial component may seriously hinder the quality of downstream results. In particular, biomedical research literature is prone to noise caused by the mischaracterization of the *polarity* (e.g., promotion vs. inhibition) of biochemical interactions. This is the focus of this work.

The identification of polarity in the biomedical domain is complicated by the fact that the language used is often hedged through multiple negations to stay closer to the complex biology underneath. For example, consider the statement: *The inactivation of Bad is sufficient to antagonize p38 MAPK.* Under the (simplified but commonly used) representation of polarized interactions, a naive IE system would extract a negative interaction between the two proteins: `Bad inhibits p38 MAPK`, due to the presence of the negative predicate *antagonize*. However, a more careful reading of this text indicates that the better representation for this extraction is a positive interaction: `Bad promotes p38 MAPK`,[1] due to the interaction of two predicates with negative semantics, *inactivation* and *antagonize*. This situation is exacerbated by the fact that statements in this domain may contain three and even four inter-related predicates that affect polarity (as observed in Section 8).

This paper analyzes the identification of polarity of biomedical interactions, from the perspective of multiple possible methods. In particular, the contributions of this work are:

(1) We introduce a novel dataset that annotates the polarity of biomedical interactions. The dataset comes in multiple variants. A first variant was derived using *distant supervision* (DS) (Mintz et al., 2009) by aligning a knowledge base (KB) of protein interactions (Perfetto et al., 2015) with the outputs of a machine reader (Valenzuela-Escárcega et al., 2018). This dataset contains 52,779 promotion and 35,177 inhibition interactions. To account for the noise introduced through the DS process, we provide a second variant of this dataset consisting of a sample of the full dataset

[1] This representation is better but not perfect. The correct representation should be: `(decrease of Bad) causes (decrease of p38 MAPK)`. However, the promotes/inhibits representation is widely used both in IE datasets and by a domain expert, so we continue to use it in this work.

that was manually curated by domain experts. We divide this sample into an *Easy* partition where the IE system initially agreed with the KB, and a *Challenge* partition where the IE system's extractions conflicted with the KB. These manually-curated partitions contain 62 and 67 data points, respectively.

(2) We compare several approaches for polarity identification, including a linguistically-informed method (Valenzuela-Escárcega et al., 2018), and several deep learning (DL) approaches. The DL methods incorporate: (a) multiple sequence models that capture the text before/after arguments/predicate, (b) attention models, and (c) explicit features from the linguistically-informed method. Our analysis indicates that: (a) the simpler DL methods perform better than the more complicated ones, (b) all DL approaches outperform the standalone linguistically-informed method, and (c) the difference between the two strategies grows larger with the complexity of the text.

2 Related work

The rate of scientific publishing has grown substantially each year, reaching a level that exceeds the human capacity to read and process. For example, PubMed, a search engine of biomedical publications[2] now indexes over 25 million papers, 17 million of which were published between 1990 and the present. Domain-agnostic approaches, such as open information extraction (OpenIE) (Angeli et al., 2015) can begin to mitigate this by extracting information in the form of relation triples. However the widely varied language used by authors means that extractions can be difficult to aggregate and utilize.

On the other hand, there have been significant efforts to develop domain-specific information extraction approaches that are tailored to scientific publications. These approaches range from rule-based to machine learning-based, and hybrid approaches (Valenzuela-Escárcega et al., 2018; Peng et al., 2017; Quirk and Poon, 2016; Kim et al., 2013; Björne and Salakoski, 2013; Hakala et al., 2013; Bui et al., 2013).

On top of the extractions produced by these methods, causal influence crucially relies on the *polarity* of the influence interactions, i.e., whether one factor *promotes* or *inhibits* another factor. Biological models have been assembled from these interactions and used for domain-specific applications (Gyori et al., 2017). Here we propose an approach for automatically detecting this polarity.

Polarity detection has been explored in several other natural language processing tasks, perhaps most notably in sentiment analysis (e.g., Pang et al., 2008; Liu, 2012; Liu and Zhang, 2012), where the polarity of a text is measured on a spectrum from negative to positive sentiment. Similarly, in Wilson et al. (2005), the authors frame the problem of extracting opinion polarity explicitly as a sentiment analysis task. Our work is similar in spirit, but it focuses on the polarity of scientific statements. In (Lauscher et al., 2017), the authors investigate the polarity polarity of citations within the context of bibliometric analysis. In contrast, our work addresses the polarity of *content*, i.e., events extracted from the biomedical literature.

To summarize, our approach is inspired by this previous work, but it differs in two ways: first, we focus on statements in the biomedical domain, and, second, we extract polarity for specific, structured events rather than unstructured texts.

3 Linguistically-informed polarity identification approach

In preliminary analyses, we observed that the arguments of biomedical events are generally correctly identified, but the polarity of the interactions is often incorrect due to the complex language used (see, for example, the example in Section 1). Based on this observation, all the methods introduced in this paper assume that an *unlabeled* event is provided, e.g., `Bad interacts_with p38 MAPK`, and the methods then label the event with a polarity type, e.g., `Bad promotes p38 MAPK`.

The first method analyzed, which extends the approach in Valenzuela-Escárcega et al. (2018), relies on linguistic cues. The approach takes the following steps:

1. First, it extracts the syntactic dependency path between the participants in the interaction.

2. Then, the path is expanded to include modifiers of the words along the above path.

3. Finally, the method counts the number of polarity-carrying words and affixes (Bach-

[2] http://www.ncbi.nlm.nih.gov/pubmed

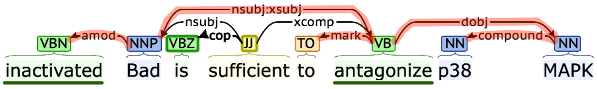

Figure 1: Example of the linguistically-informed approach. From the syntactic dependency tree, the approach extracts the shortest undirected path between the participants in the interaction, *Bad* and *p38 MAPK*: `nsubj:xsubj> dobj>`, where the > and < markers indicate the direction of a dependency arc. Then, the path is extended with modifiers of the elements on the path: `amod< nsubj:xsubj> mark< dobj> compound<`. (The complete path is highlighted and the negative words are underlined.) Lastly, the approach counts the number of polarity-carrying words along this path. An odd number indicates negative event polarity; otherwise the polarity is positive. In this example, the polarity is positive because there is an even number of polarity words: *inactivated* and *antagonize*.

man et al., 2018) from a defined lexicon. This lexicon contains 33 elements, such as "inhibition" and "loss". The event is labeled with negative polarity (inhibits) if the count of these words is odd. Otherwise, the polarity of the event is positive (promotes).

Figure 1 shows a walkthrough of this algorithm for the sentence *Inactivated Bad is sufficient to antagonize p38 MAPK*, which contains an event connecting the two entities *Bad* and *p38 MAPK*. Step 2 of this algorithm is crucial, as many polarity-carrying words, e.g., *inactivated*, do not appear along the syntactic dependency path between the event arguments, but rather modify terms on the path.

We extended the original algorithm in Valenzuela-Escárcega et al. (2018) as follows:

- We made the polarity lexicon case-insensitive.

- We changed the algorithm to match the words in the polarity lexicon only if they occur as a full word or as a prefix, instead of any substring of a word. For example, in the text *Reduction of triglyceride synthesis without affecting **ALLN-inhibitable** protease*, the original algorithm generates a false positive by matching *inhibit* in *ALLN-inhibitable*.

- We handle verb particles, which were ignored in the original algorithm. For example, in the text *The Wip1 gene is overexpressed by **switching off** p53*, the polarity of the interaction cannot be detected from the predicate alone (*switching*) without its attached particle (*off*).

- We adjusted the polarity lexicon, e.g., we removed *target*; and we added the suffix *-KD* (Bachman et al., 2018), which stands for *knockdown*.

4 Deep learning polarity identification approaches

We propose several deep learning approaches for the classification of event polarity. In general, all proposed approaches use recurrent neural network (RNN) architectures, which incorporate both lexical and structural information into the learning process by considering one or more sequences of words from the source sentence for the given event.

In each of the RNN model variants we investigate, the input sentence is represented as a sequence of word embeddings. Every word w_t triggers a recurrent state that generates a hidden vector h_t, which encodes information about the input word subsequence $1..t$. The output of the RNN is a sequence $\{h_t\}$ of hidden vectors, one for each of the input words.

The hidden vector sequence is then aggregated using one of a couple of different strategies (as described in the next two sub-sections), and then passed forward as the input to a multi-layer perceptron (MLP) that performs binary classification of the event's polarity: positive or negative.

Because our approach applies to biochemical events, we use the result of the underlying IE method to encode the predicate of the event, or its *trigger*, as a feature in the MLP. That is, if the trigger belongs to the lexicon of positive-polarity terms, such as *promotes* or *activates*, the network uses this as evidence for positive polarity. Conversely, if the trigger belongs to the negative-polarity lexicon, such as *inhibits*, the trigger is evidence of negative polarity. We use the same dictionary of polarity-carrying words as the linguistically-informed method.

We investigated two families of architectures: (a) passing the entire input sentence to a single recurrent network, and (b) splitting the sentence into several semantic segments and passing these fragments to independent RNNs (see Figure 2).

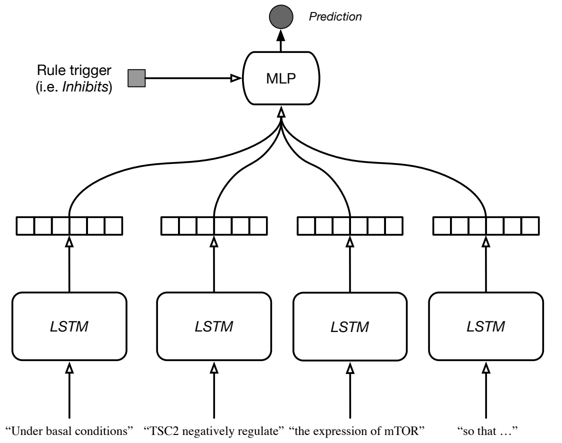

Figure 2: Four-segment LSTM architecture for polarity identification. The four segments model: the text before the left-most event argument, the text between the left-most argument and the event predicate, the text between the predicate and the right-most argument; and the text after the right-most argument. The outputs of the four LSTMs are integrated through a MLP, which also uses the polarity of the event trigger as an explicit feature.

We describe these approaches in the next two subsections.

4.1 Single-segment architecture

For this variant of the architecture, we consider the input sequence, consisting of the span of text that belongs to the event as a single unit, and take the last vector of the hidden sequence as input to the MLP, discarding the rest of the sequence's hidden states. The output of the MLP directly labels the polarity of the event on top of this hidden state vector.

4.2 Four-segment architecture

The structure of the biochemical events modeled here have the following elements: *controller* (or cause), *trigger* (or predicate), and *controlled* (or theme). These elements are text-bounded and partition the source sentence into *four* regions: a window of text *before* the controller, up to three words, the text *between* the controller and the trigger, the text *between* the trigger and the controlled and the window of text *after* the controlled. If the trigger appears before or after both, controller and controlled (i.e. *the phosphorylation of ERK by MEK*), then the event text is considered as a single segment instead of two.

Each of the four sections of the source sentence is then fed to an independent LSTM using the same strategy as in Section 4.1. Figure 2 illustrates how the sentence *Under basal conditions, TSC2 negatively regulates the expression of mTOR, so that ...* is split and processed by this approach. The last vectors of the four hidden sequences are concatenated and passed as input to MLP for polarity classification.

4.3 Additional enhancements

We implemented and tested the following enhancements with both the single-segment and four-segment architectures from Sections 4.1 and 4.2 respectively.

Pre-trained word embeddings

We used Word2Vec (Mikolov et al., 2013) to pre-initialize the word embeddings. We pre-trained these embeddings over the open-access subset of PubMed Central[3]. We used dimension 100 for these vectors.

Character-level embeddings

To capture information present in the morphological structure of a word, we extended our approaches to use *character-level embeddings*. Each word w in an input sentence is enhanced by adding character-level embeddings to its word embedding e^w.

Given the characters of word w, each is mapped to an embedding e^c. The resulting sequence of character embeddings $\{e^c_t\}$ is then passed forwards and backwards through a *bi-directional GRU* (Goldberg, 2017). Then, the last hidden vectors of the forward and backward GRUs are concatenated into the word's characters embedding e^{wc}.

The word embedding and the word's characters embedding are then concatenated into an enhanced word embedding $e^{w'} = [e^w; e^{wc}]$, which is passed as input for the current word of our polarity network architecture.

Attention mechanisms for aggregation

So far, in all proposed approaches the last element of a sequence has been used as input to the MLP for classification. By doing this, the remaining sequence leading to the selected hidden vector is discarded with respect to classification. To account for this potential limitation, we implemented attention mechanisms (Bahdanau et al., 2014) to ag-

[3] `https://www.ncbi.nlm.nih.gov/pmc/tools/openftlist/`

gregate all the hidden vectors into the classification step of the network.

The attention mechanism functions as a weighted average of a sequence $\{h_t\}$ of vectors dictated by $\sum_t^T \alpha_t h_t$. The weight parameters $\{\alpha_t\}$ are learnt jointly with the rest of the network parameters. The scalar coefficient α_t for the vector h_t is computed using the linear combination: $a_t = W_a h_t + b_a$, where the parameters W_a and b_a are shared for all the observations passed through the network. The resulting sequence of coefficients $\{a_t\}$ is normalized with the *softmax* function by $\alpha_t = softmax(\{a_t\})$ to enforce that the weights sum up to 1.

The single-segment architecture is enhanced with this mechanism on top of the sequence of hidden vectors produced by the recurrent network. For the four-segment architecture, we tested an attention mechanism for the hidden vector sequences of each segment of the sentence (*shallow attention*) and an additional approach that also includes an attention mechanism to aggregate, instead of concatenate the four resulting sequence vectors before the MLP step (*deep attention*). This deep attention approach computes a weighted average of the four sequences $\{s_i\}$, dictated by $\sum_i^4 \beta_i s_i$. Similarly to the weights of the hidden vectors, each individual weight in $\{\beta_i\}$ is computed by the linear combination $b_t = W_b h_i + b_0$, where the parameters W_b and b_0 are shared and later normalized by $\beta_i = softmax(\{b_i\})$.

Bidirectional LSTMs

At any given index t of a source sentence, the LSTM network considers only the sequence $1..t$ of words to compute the hidden state vector of w_t. Clearly, this formulation discards information from words to the right of t. To address this limitation, we modified our architecture to use a bidirectional LSTM (Graves et al., 2013) as a drop-in replacement of the vanilla LSTM wherever it is used. Similarly to the bidirectional GRU, the bidirectional LSTM contains two distinct LSTM networks that process the input sentence left-to-right (forward) and right-to-left (backward). The last hidden vectors of both are concatenated and used for the next step in our architectures.

5 Dataset

To analyze the performance of the above approaches, we assembled a dataset of sentences associated with protein-protein interaction events, as well as polarity labels. The dataset was constructed through distant supervision (Mintz et al., 2009), by aligning events extracted from biomedical literature by Reach, a biomedical IE system (Valenzuela-Escárcega et al., 2018), with polarity labels from the SIGNOR database (Perfetto et al., 2015).

SIGNOR contains approximately 20,000 manually curated protein interactions, the majority of which are annotated with the polarity of the effect of the interaction on the downstream protein (activation or inhibition). These signed interactions were used to establish the true polarities for each pair of proteins in the database. A potential issue with this approach is that an interaction among proteins may have more than one possible polarity depending on the biological context: for example, protein A may activate protein B in cell type X, but inhibit protein B in cell type Y. To mitigate this, we filtered the relations in SIGNOR for those annotated with only a single, unambiguous polarity, under the assumption that for the relatively well-characterized interactions prioritized for curation in a pathway database, the assignment of a single polarity would be a good indicator of "ground truth" for the majority of texts. Processing the SIGNOR database in this way yielded 17,163 protein-protein interactions among with a single polarity, composed of the following interaction types: 13,302 interactions with positive polarity, and 3,861 interactions with negative polarity.

We extracted protein-protein-interaction events from text by running the Reach IE system over all full-text articles in PubMed Central[4], the PubMed Central Author's Manuscript collection[5], and MEDLINE[6] abstracts (for articles not included in the full-text datasets). We kept all information about the events (e.g., triggers, participants, overall interaction type), but discarded polarity information. We assigned polarity labels by aligning these events with SIGNOR interactions that involved the same two proteins and the same overall interaction type, irrespective of sign (e.g., regulation of activity or regulation of phosphorylation). From this dataset we removed: (a) duplicate sentences, and (b) sentences containing events where at least one of the participating pro-

[4] https://www.ncbi.nlm.nih.gov/pmc/tools/openftlist/

[5] https://www.ncbi.nlm.nih.gov/pmc/about/mscollection/

[6] https://www.nlm.nih.gov/bsd/medline.html

tein names could not be grounded to an entry in the UniProt protein database[7]. This process produced 68,935 polarity-labeled events (with supporting sentences). For 54,105 of these events, the original polarity detector in Reach agreed with the SIGNOR polarity label (a strong indication that these sentences are easier to classify). For 14,830 events, Reach's polarity disagreed with SIGNOR (an indication that these sentences are more challenging). We call this dataset the *DS dataset* (from distant supervision). Table 1 lists the distribution of labels for the DS dataset on both the Easy and Challenge partitions and overall.

	Positive polarity	***Negative polarity***
Easy	40,339	13,766
Challenge	7,262	7,568
Total	47,601	21,334

Table 1: Label distribution on the *DS* dataset.

The distant supervision process is potentially noisy (Yao et al., 2011). To control for this noise, we also created two smaller hand-curated datasets, as follows:

1. We randomly sampled 100 sentences from the sentences where Reach agreed with SIGNOR, and 100 from the sentences where Reach disagreed with SIGNOR. Based on the intuition mentioned in the previous paragraph, we call these partitions *Easy* and *Challenge*.

2. Because the focus of this work is on polarity identification *given* a correct event, we eliminated the false positive events from both partitions, i.e., events extracted by Reach that were not supported by the corresponding underlying sentence. Further, we removed sentences containing events where at least one of the participating protein names could not be grounded to UniProt. This reduced the size of the dataset to 62 Easy and 67 Challenge examples.

3. The remaining sentences were manually curated by a domain expert. The expert corrected 2 polarity labels in the Easy partition, and 53 labels in the Challenge partition, confirming our expectation that the latter partition is harder than the former.

To facilitate reproducibility, we will release all these datasets (and the software) upon acceptance.

6 Results

We performed a five-fold cross validation experiment on the DS dataset introduced in Section 5 to assess the performance of the linguistically-informed baseline (Section 3) and of the various neural models previously described in Section 4. Note that this dataset contains all the elements from both Easy and Challenge partitions. The data was split randomly and the experiment was repeated with five different random seeds and the numbers reported are the corresponding averages from all the trials. Table 2 reports these average scores as well as the standard deviations for all the approaches analyzed.

Tables 3 and 4 contain the results on the manually-curated Easy and Challenge partitions, when the corresponding models were trained on the entire DS dataset.

The code and data used to generate these results are available at this URL: https://github.com/clulab/releases/tree/master/naacl-essp2019-polarity.

7 Discussion

7.1 Discussion of the main results

Table 2 shows that the linguistically-informed approach performs reasonably well overall, with a F1 score of 0.862. This is encouraging, but also somewhat misleading. The DS dataset consists of mostly Easy examples, where Reach agreed with SIGNOR labels. As discussed in Section 5, the distribution of the examples in the DS dataset is 78.4/21.6% Easy/Challenge. Tables 3 and 4 show that the performance of the linguistically-informed approach, which is an improved version of the method in Reach, drops to 0.143 F1 when evaluated solely on challenging sentences.

On the other hand, the results summarized in Tables 2 through 4 demonstrate that overall, deep learning architectures that incorporate bidirectional recurrence with character-level embeddings perform the best. The reasonable explanation for this is that those specific enhancements are aimed at capturing more global information from the sentence, instead of just the information

[7] https://www.uniprot.org

Architecture variant	***F1 (st. dev.)***	***Precision (st. dev.)***	***Recall (st. dev.)***
Linguistically-informed approach	0.862	0.859	0.865
Single-segment architecture			
– biLSTM, char embed, no pretrained embed, no attention, trigger	**0.968(0.001)**	**0.967(0.001)**	**0.969(0.000)**
– biLSTM, char embed, no pretrained embed, no attention, no trigger	**0.968(0.001)**	**0.967(0.001)**	0.968(0.000)
– LSTM, char embed, no pretrained embed, no attention, trigger	0.966(0.001)	0.964(0.001)	0.967(0.001)
– LSTM, no char embed, no pretrained embed, no attention, trigger	0.961(0.000)	0.959(0.001)	0.963(0.001)
– LSTM, char embed, no pretrained embed, attention, trigger	0.954(0.001)	0.954(0.001)	0.955(0.002)
– LSTM, char embed, pretrained embed, no attention, trigger	0.948(0.001)	0.944(0.002)	0.952(0.001)
– LSTM, no char embed, pretrained embed, no attention, trigger	0.943(0.000)	0.938(0.001)	0.948(0.001)
– biLSTM, char embed, no pretrained embed, no attention, trigger, mask	0.874(0.001)	0.852(0.010)	0.897(0.012)
Four-segment architecture			
– LSTM, char embed, no pretrained embed, no attention, trigger	0.956(0.000)	0.956(0.001)	0.956(0.000)
– LSTM, char embed, no pretrained embed, attention$_{\text{deep}}$	0.948(0.000)	0.949(0.001)	0.947(0.001)
– LSTM, char embed, no pretrained embed, attention$_{\text{shallow}}$	0.948(0.000)	0.951(0.001)	0.945(0.001)

Table 2: Deep learning scores from a five-fold cross-validation experiment on the larger DS dataset. The "mask" option indicates that event participants have been masked (please see Section 7.3 for details).

Architecture variant	***F1 (st. dev.)***	***Precision (st. dev.)***	***Recall (st. dev.)***
Linguistically-informed approach	**0.989**	**0.979**	**1.0**
Single-segment architecture			
– biLSTM, char embed, no pretrained embed, no attention, no trigger	0.983(0.009)	0.978(0.000)	0.987(0.017)
– biLSTM, char embed, no pretrained embed, no attention, trigger	0.980(0.011)	0.978(0.000)	0.983(0.021)
– LSTM, no char embed, pretrained embed, no attention, trigger	0.974(0.005)	0.987(0.011)	0.961(0.009)
– LSTM, char embed, no pretrained embed, no attention, trigger	0.972(0.006)	0.978(0.000)	0.965(0.011)
– LSTM, no char embed, no pretrained embed, no attention, trigger	0.972(0.006)	0.978(0.000)	0.965(0.011)
– LSTM, char embed, pretrained embed, no attention, trigger	0.971(0.005)	0.987(0.011)	0.957(0.000)
– LSTM, char embed, no pretrained embed, attention, trigger	0.964(0.011)	0.987(0.011)	0.943(0.022)
– biLSTM, char embed, no pretrained embed, no attention, trigger, mask	0.942(0.017)	0.964(0.017)	0.922(0.029)
Four-segment architecture			
– LSTM, char embed, no pretrained embed, no attention, trigger	0.974(0.006)	0.978(0.000)	0.970(0.011)
– LSTM, char embed, no pretrained embed, attention$_{\text{shallow}}$	0.960(0.005)	0.973(0.008)	0.948(0.011)
– LSTM, char embed, no pretrained embed, attention$_{\text{deep}}$	0.958(0.017)	0.965(0.010)	0.952(0.035)

Table 3: Performance of all approaches on the *Easy* partition. The "mask" option indicates that event participants have been masked (please see Section 7.3 for details).

Architecture variant	***F1 (st. dev.)***	***Precision (st. dev.)***	***Recall (st. dev.)***
Linguistically-informed approach	0.143	0.138	0.148
Single-segment architecture			
– LSTM, char embed, no pretrained embed, no attention, trigger	**0.757(0.022)**	0.659(0.019)	**0.889(0.033)**
– biLSTM, char embed, no pretrained embed, no attention, no trigger	0.752(0.007)	**0.665(0.011)**	0.867(0.018)
– biLSTM, char embed, no pretrained embed, no attention, trigger	0.748(0.031)	0.658(0.032)	0.867(0.030)
– LSTM, no char embed, no pretrained embed, no attention, trigger	0.733(0.008)	0.648(0.008)	0.844(0.015)
– LSTM, char embed, no pretrained embed, attention, trigger	0.703(0.010)	0.628(0.008)	0.800(0.030)
– LSTM, char embed, pretrained embed, no attention, trigger	0.690(0.025)	0.607(0.024)	0.800(0.030)
– LSTM, no char embed, pretrained embed, no attention, trigger	0.686(0.013)	0.610(0.014)	0.785(0.028)
– biLSTM, char embed, no pretrained embed, no attention, trigger, mask	0.576(0.009)	0.472(0.008)	0.741(0.033)
Four-segment architecture			
– LSTM, char embed, no pretrained embed, no attention, trigger	0.698(0.014)	0.638(0.008)	0.770(0.028)
– LSTM, char embed, no pretrained embed, attention$_{\text{deep}}$	0.696(0.017)	0.640(0.019)	0.763(0.018)
– LSTM, char embed, no pretrained embed, attention$_{\text{shallow}}$	0.690(0.018)	0.640(0.020)	0.748(0.015)

Table 4: Performance of all approaches on the *Challenge* partition. The "mask" option indicates that event participants have been masked (please see Section 7.3 for details).

found around the dependency path representing the event. Taking into account the full, global information in the sentence as a single segment results in a simpler neural network with fewer parameters, which may also explain why the four-segment architecture, which splits the sentence

Number of negative words per sentence	*Sample size*	*Best DL approach*			*Linguistically-informed approach*		
		Precision	*Recall*	*F1*	*Precision*	*Recall*	*F1*
0	49,972	0.978	0.980	0.98	0.873	0.937	0.904
1	16,063	0.90	0.902	0.901	0.694	0.38	0.491
2	2,566	0.94	0.896	0.917	0.773	0.691	0.730
3	300	0.884	0.857	0.87	0.675	0.49	0.568
4	30	1.0	0.92	0.958	0.8	0.48	0.6
5	3	1.0	1.0	1.0	0.5	0.5	0.5
6	1	1.0	1.0	1.0	1.0	1.0	1.0

Table 5: Polarity classification results stratified by the number of polarity-carrying words in the corresponding sentence.

Number of negative words per sentence	*Sample size*	*Best DL approach*			*Linguistically-informed approach*		
		Precision	*Recall*	*F1*	*Precision*	*Recall*	*F1*
0	49,972	0.882	0.934	0.907	0.873	0.937	0.904
1	16,063	0.643	0.627	0.761	0.694	0.38	0.491
2	2,566	0.789	0.734	0.761	0.773	0.691	0.730
3	300	0.744	0.689	0.716	0.675	0.49	0.568
4	30	0.941	0.64	0.761	0.8	0.48	0.6
5	3	1.0	1.0	1.0	0.5	0.5	0.5
6	1	1.0	1.0	1.0	1.0	1.0	1.0

Table 6: Polarity classification results stratified by the number of polarity-carrying words in the corresponding sentence with masked participants.

into subsequences according to the components associated with the cause, predicate and theme, and runs each through distinct recurrent components in the architecture does not perform quite as well as the full, single-segment architecture.

Although the deep learning models generally outperform the linguistically-informed model, Tables 3 and 4 uncover an interesting pattern in the differential performance on the *Easy* and *Challenge* data sets. In particular, on the *Easy* data set, the linguistically informed approach performs exceptionally well, better than the highest performing deep learning model. The good performance of the linguistically-informed model is not surprising here because, as discussed, the instances in the data set were those for which the linguistically-informed agreed with SIGNOR. But it is encouraging that the best deep learning model manages to achieve this performance as well.

On the *Challenge* data set, however, the linguistically-informed model performance dives to an F1 of 0.143. Again, this is not a surprise given that these data were ones that specifically disagreed with a version of the linguistic model. However, the performance of the best deep learning model degrades just to 0.757 F1, demonstrating the capacity of the model to maintain relatively good performance in the face of more challenging data. We find this very encouraging, especially considering that the neural models were trained on the DS dataset, which contains distant-supervision noise. These results demonstrate that the neural models are able to generalize despite the presence of noise.

Somewhat surprisingly, no attention-based model outperformed the simpler bidirectional LSTM without attention. This highlights that the simpler LSTM method is sufficient to model polarity in this context, and that, possibly, the attention mechanisms are more likely to overfit on the distant-supervision noise present in this training data.

7.2 Analysis of complexity by negative terms

To better understand why the *Challenge* data set was more difficult, we compared the performance of the linguistically-informed approach to the best deep learning model in detail. In this experiment, we partitioned the data from the DS dataset into subsets according to how many negative polarity words (from the negative polarity lexicon described in Section 3) appeared in a sentence and evaluated each subset individually. Training for the DL approach was performed using five-fold cross-validation and the testing scores were computed only for the instances with a specific number of negative polarity words. Table 5 summarizes these results. Unsurprisingly, the scores are negatively correlated with the number of negative words in the sentence for both approaches. However, the linguistic approach suffers a much faster drop in performance as the complexity of the sentence increases. The best deep learning model, however, still attains good performance even when there are more than two negative words in the sentence. For example, the linguistically-informed method drops in performance from 0.904 F1 in sentences with zero negative words to just 0.6 F1 in sentences with four negative words, whereas the best neural model drops from 0.98 to 0.958 F1 in the same subsets. This is further proof that the neural methods are able to aggregate multiple negative-polarity hints from the larger context surrounding the events.

7.3 Masking participants

To mitigate the potential of our method to overfit to the entities present in the events analyzed, we implemented a variant of the previous analysis in which we replaced the words that belonged to a participant in a regulation event, both controller and controlled, with a predefined token that masks its identity but preserves its role in the event. For example, in the sentence *PTEN Plays a Role in the Activation of the PI3K Signaling Pathway*, the participants *PTEN* and *PI3K* will be replaced by the terms *CONTROLLER* and *CONTROLLED*, respectively.

Table 6 presents the results of this analysis. The table indicates that the performance of deep learning models decreases in general. However, the same pattern observed when not masking the participants arises. That is, the deep learning approach is not affected as much when the number of negative terms increases compared to the linguistic approach. Please note that this evaluation is more stringent and could be considered a lower-bound to what can be expected from a real world scenario. It also proves that the deep learning models do capture most their signal from the structure of the sentence in which the event is extracted, and have a degree of resilience when facing participants that were not observed during training. Tables 2–4 also show results for a model trained with masked participants in the corresponding scenario.

8 Conclusions

We have introduced a corpus for the development and assessment of approaches to assigning correct polarity to biochemical events. Using this corpus, we trained and evaluated a variety of deep learning architectures and compared them to a linguistically-informed model.

The best-performing deep learning architectures incorporate character embeddings with a bidirectional LSTM across the entire input sentence, achieving an average F1 of 0.972 in a five-fold cross-validation study. This model was found to do just as well as the linguistically-informed model on examples that the linguistically-informed model does well on, but maintains much more robust performance in the face of more difficult cases.

We also explored a deep learning architecture that splits the input sentence into components that are generally meaningful for the task, but found that this did not reach the accuracy of the single-segment input model, suggesting that there is important information spread across sentence components that should be jointly processed.

Additional work remains. Further work should be devoted to gain further F1 improvement, and the place to start is deeper analyses of the kinds of errors made by the best performing model. Another issue is speed efficiency: the linguistically-informed model processes a sentence much faster than the deep learning models, so is better-adapted for high-throughput use cases. An area of further exploration is to consider the pattern observed in Table 5 and assess the tradeoffs of using the fast linguistically-informed model for simpler sentences (with no negative words) and then use the slower deep learning model for more complex sentences.

9 Acknowledgments

This work was supported by the Defense Advanced Research Projects Agency (DARPA) under the Automated Scientific Discovery Framework (ASDF) program, grant W911NF018-1-0124. Mihai Surdeanu declares a financial interest in lum.ai. This interest has been properly disclosed to the University of Arizona Institutional Review Committee and is managed in accordance with its conflict of interest policies.

References

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D Manning. 2015. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 344–354.

John A. Bachman, Benjamin M. Gyori, and Peter K. Sorger. 2018. Famplex: a resource for entity recognition and relationship resolution of human protein families and complexes in biomedical text mining. *BMC Bioinformatics*, 19(1):248.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Jari Björne and Tapio Salakoski. 2013. Tees 2.1: Automated annotation scheme learning in the bionlp 2013 shared task. In *Proceedings of the BioNLP shared task 2013 workshop*, pages 16–25.

Quoc-Chinh Bui, David Campos, Erik van Mulligen, and Jan Kors. 2013. A fast rule-based approach for biomedical event extraction. In *proceedings of the BioNLP shared task 2013 workshop*, pages 104–108.

Yoav Goldberg. 2017. Neural network methods for natural language processing. *Synthesis Lectures on Human Language Technologies*, 10(1):1–309.

Alex Graves, Navdeep Jaitly, and Abdel-rahman Mohamed. 2013. Hybrid speech recognition with deep bidirectional lstm. In *2013 IEEE workshop on automatic speech recognition and understanding*, pages 273–278. IEEE.

Benjamin M. Gyori, John A. Bachman, Kartik Subramanian, Jeremy L. Muhlich, Lucian Galescu, and Peter K. Sorger. 2017. From word models to executable models of signaling networks using automated assembly. *Molecular Systems Biology*, 13(11):954.

Kai Hakala, Sofie Van Landeghem, Tapio Salakoski, Yves Van de Peer, and Filip Ginter. 2013. Evex in st'13: Application of a large-scale text mining resource to event extraction and network construction. In *Proceedings of the BioNLP Shared Task 2013 Workshop*, pages 26–34.

Jin-Dong Kim, Yue Wang, and Yamamoto Yasunori. 2013. The genia event extraction shared task, 2013 edition-overview. In *Proceedings of the BioNLP Shared Task 2013 Workshop*, pages 8–15.

Anne Lauscher, Goran Glavaš, Simone Paolo Ponzetto, and Kai Eckert. 2017. Investigating convolutional networks and domain-specific embeddings for semantic classification of citations. In *Proceedings of the 6th International Workshop on Mining Scientific Publications*, pages 24–28. ACM.

Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167.

Bing Liu and Lei Zhang. 2012. A survey of opinion mining and sentiment analysis. In *Mining text data*, pages 415–463. Springer.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 1003–1011. Association for Computational Linguistics.

Bo Pang, Lillian Lee, et al. 2008. Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, 2(1–2):1–135.

Nanyun Peng, Hoifung Poon, Chris Quirk, Kristina Toutanova, and Scott Yih. 2017. Cross-sentence n-ary relation extraction with graph lstms.

Livia Perfetto, Leonardo Briganti, Alberto Calderone, Andrea Cerquone Perpetuini, Marta Iannuccelli, Francesca Langone, Luana Licata, Milica Marinkovic, Anna Mattioni, Theodora Pavlidou, et al. 2015. Signor: a database of causal relationships between biological entities. *Nucleic acids research*, 44(D1):D548–D554.

Chris Quirk and Hoifung Poon. 2016. Distant supervision for relation extraction beyond the sentence boundary.

Marco A. Valenzuela-Escárcega, Özgün Babur, Gus Hahn-Powell, Dane Bell, Thomas Hicks, Enrique Noriega-Atala, Xia Wang, Mihai Surdeanu, Emek Demir, and Clayton T. Morrison. 2018. Large-scale automated machine reading discovers new cancer-driving mechanisms. *Database*, 2018.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*.

Limin Yao, Aria Haghighi, Sebastian Riedel, and Andrew McCallum. 2011. Structured relation discovery using generative models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1456–1466. Association for Computational Linguistics.