Scientific Information Extraction with Semi-supervised Neural Tagging

Yi Luan **Mari Ostendorf** **Hannaneh Hajishirzi**
Department of Electrical Engineering, University of Washington
{luanyi, ostendor, hannaneh}@uw.edu

Abstract

This paper addresses the problem of extracting keyphrases from scientific articles and categorizing them as corresponding to a task, process, or material. We cast the problem as sequence tagging and introduce semi-supervised methods to a neural tagging model, which builds on recent advances in named entity recognition. Since annotated training data is scarce in this domain, we introduce a graph-based semi-supervised algorithm together with a data selection scheme to leverage unannotated articles. Both inductive and transductive semi-supervised learning strategies outperform state-of-the-art information extraction performance on the 2017 SemEval Task 10 ScienceIE task.

1 Introduction

As a research community grows, more and more papers are published each year. As a result there is increasing demand for improved methods for finding relevant papers and automatically understanding the key ideas in those papers. However, due to the large variety of domains and extremely limited annotated resources, there has been relatively little work on scientific information extraction. Previous research has focused on unsupervised approaches such as bootstrapping (Gupta and Manning, 2011; Tsai et al., 2013), where hand-designed templates are used to extract scientific keyphrases, and more templates are added through bootstrapping.

Very recently a new challenge on Scientific Information Extraction (ScienceIE) (Augenstein et al., 2017)[1] provides a dataset consisting of 500 scientific paragraphs with keyphrase annotations for three categories: TASK, PROCESS, MATERIAL across three scientific domains, Computer Science (CS), Material Science (MS), and Physics (Phy), as in Figure 1. This dataset enables the use of more advanced approaches such as neural network (NN) models. To that end, we cast the keyphrase extraction task as a sequence tagging problem, and build on recent progress in another information extraction task: Named Entity Recognition (NER) (Lample et al., 2016; Peng and Dredze, 2015). Like named entities, keyphrases can be identified by their linguistic context, e.g. researchers "use" methods. In addition, keyphrases can be associated with different categories in different contexts. For example, 'semantic parsing' can be labeled as a TASK in one article and as a PROCESS in another. Scientific keyphrases differ in that they can include both noun phrases and verb phrases and in that non-standard "words" (equations, chemical compounds, references) can provide important cues.

Computer Science:
This paper addresses the task of **[named entity recognition]**$_{\text{Task}}$, using **[conditional random fields]**$_{\text{Process}}$. Our method is evlauated on the **[ConLL NER Corpus]**$_{\text{Material}}$.

Physics:
[Local field effects] $_{\text{Process}}$ on spontaneous emission rates within **[nanostructure photonics material]**$_{\text{Material}}$ for example are familiar, and have been well used.

Material Science:
The **[Kelvin probe force microscopy technique]** $_{\text{Process}}$ allows **[detection of local EWF]**$_{\text{Task}}$ between an **[atomic force micorscopy]**$_{\text{Material}}$ and **[metal surface]**$_{\text{Material}}$.

Figure 1: Annotated ScienceIE examples.

Since the scale of the data is still small for supervised training of neural systems, we introduce semi-supervised methods to the neural tagging

[1] SemEval (Task 10) https://scienceie.github.io/index.html

Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 2624–2634
Copenhagen, Denmark, September 7–11, 2017.

model in order to take advantage of the large quantity of unlabeled scientific articles. This is particularly important because of the differences in keyphrases across domains. Our semi-supervised learning algorithm uses a graph-based label propagation scheme to estimate the posterior probabilities of unlabeled data. It additionally extends the training objective to leverage the confidence of the estimated posteriors. The new training treats low confidence tokens as missing labels and computes the sentence-level score by marginalizing over them.

Our experiments show that our neural tagging model achieves state-of-the-art results in the SemEval Science IE task. We further show that both inductive and transductive semi-supervised strategies significantly improve the performance. Finally, we provide in-depth analysis of domain differences as well as analysis of failure cases.

The key contributions of our work include: i) achieving state of the art in scientific information extraction SEMEVAL Task 10 by extending recent advances in neural tagging models; ii) introducing a semi-supervised learning algorithm that uses graph-based label propagation and confidence-aware data selection, iii) exploring different alternatives for taking advantage of large, multi-domain unannotated data including both unsupervised embedding initialization and semi-supervised model training.

2 Related Work

There has been growing interest in research on automatic methods to help researchers search and extract useful information from scientific literature. Past research has addressed citation sentiment (Athar and Teufel, 2012b,a), citation networks (Kas, 2011; Gabor et al., 2016; Sim et al., 2012; Do et al., 2013; Jaidka et al., 2014), summarization (Abu-Jbara and Radev, 2011) and some analysis of research community (Vogel and Jurafsky, 2012; Anderson et al., 2012; Luan et al., 2012, 2014b; Levow et al., 2014). However, due to scarce hand-annotated data resources, previous work on information extraction (IE) for scientific literature is very limited. Gupta and Manning (2011) first proposed a task that defines scientific terms for 474 abstracts from the ACL anthologhy (Bird et al., 2008) into three aspects: *domain*, *technique* and *focus* and apply template-based bootstrapping to tackle the problem. Based on this study, Tsai et al. (2013) improve the performance by introducing hand-designed features from NER (Collins and Singer, 1999) to the bootstrapping framework. QasemiZadeh and Schumann (2012) compile a dataset of scientific terms into 7 fine-grained categories for 171 abstracts of ACL anthology. Similar to our work, very recently Augenstein and Søgaard (2017) also evaluated on ScienceIE dataset, but use multi-task learning to improve the performance of a supervised neural approach. Instead, we introduce a semi-supervised neural tagging approach that leverages unlabeled data.

Neural tagging models have been recently introduced to tagging problems such as NER. For example, Collobert et al. (2011) use a CNN over a sequence of word embeddings and apply a CRF layer on top. Huang et al. (2015) use hand-crafted features with LSTMs to improve performance. There is currently great interest in using character-based embeddings in neural models. (Chiu and Nichols, 2016; Lample et al., 2016; Ballesteros et al., 2015; Ma and Hovy, 2016). Our approach also takes advantage of neural tagging models and character-based embeddings for IE in scientific articles.

Previous work on semi-supervised learning for neural models has mainly focused on transfer learning (Dai and Le, 2015; Luan et al., 2014a; Harsham et al., 2015) or initializing the model with pre-trained word embeddings (Mikolov et al., 2013; Pennington et al., 2014; Levy and Goldberg, 2014; Luan et al., 2016b, 2015, 2016a). In our work, we use pre-training but also use more powerful methods including graph-based semi-supervision (Subramanya and Bilmes, 2011; Liu and Kirchhoff, 2013, 2015, 2016a,b) and a method for leveraging partially labeled data (Kim et al., 2015). We show that the combination of these techniques gives better results than any one alone.

3 Problem Definition and Data

The purpose of this work is to extract phrases that can answer questions that researchers usually face when reading a paper: What TASK has the paper addressed? What PROCESS or method has the paper used or compared to? What MATERIALS has the paper utilized in experiments? While these fundamental concepts are important in a wide variety of scientific disciplines, the terms that are used in specific disciplines can be substantially differ-

ent. For example, MATERIALS in computer science might be a text corpus, while they would be physical materials in physics or materials science.

Data We use the SemEval 2017 Task 10 ScienceIE dataset. Fig. 1 provides examples that illustrate the variation in domains, but also show that there are common cues such as "the task of", "using", "technique," etc. A challenge with this dataset is that the size of the training data is very small. It is built from ScienceDirect open access publications and consists of 500 journal articles, but only one paragraph of each article is manually labeled. Therefore, we use a large amount of external data to leverage the continuous-space representation of language in neural network model. We explore the effect of pre-training word embedding with two different external resources: i) a data set of Wikipedia articles as a general English resource, and ii) a data set of 50k Computer Science papers from ACM.[2]

Tagging Problem Formulation The task requires detecting the exact span of a keyphrase. In order to be able to distinguish spans of two consecutive keyphrases of the same type, we assign labels to every word in a sentence, indicating position in the phrase and the type of phrase. We formulate the problem as an IOBES (Inside, Outside, Beginning, End and Singleton) tagging problem where every token is labeled either as: B, if it is at the beginning of a keyphrase; E, if it ends the phrase; I, if it is inside a keyphrase but not the first or last token; S, if it is a single-word keyphrase; or O, otherwise. For example, "named entity recognition" in first sentence of Fig. 1 is labeled as "*B-Task I-task E-task*".

4 Neural Architecture Model

We introduce an end-to-end model to categorize scientific keyphrases, building on a neural named entity recognition model (Lample et al., 2016) and adding a feature-based embedding.

4.1 Model

We develop a 3-layer hierarchical neural model to tag tokens of the documents (details of the tokenization is in Sec. 6). (1) The token representation layer concatenates three components for each token: a bi-directional character-based embedding, a word embedding, and an embedding associated with orthographic and part-of-speech features. (2) The token LSTM layer uses a bidirectional LSTM to incorporate contextual cues from surrounding tokens to derive intermediate token embeddings. (3) The CRF tagging layer models token-level tagging decisions jointly using a CRF objective function to incorporate dependencies between tags.

Character-Based Embedding. The embedding for a token is derived from its characters as the concatenation of forward and backward representations from a bidirectional LSTM. The character lookup table is initialized at random. The advantage of building a character-based embedding layer is that it can handle out-of-vocabulary words and equations, which are frequent in this data, all of which are mapped to "UNK" tokens in the Word Embedding Layer.

Word Embedding. Words from a fixed vocabulary (plus the unknown word token) are mapped to a vector space, initialized using Word2vec pretraining with different combinations of corpora.

Feature Embedding. We map features to a vector space: capitalization (all capital, first capital, all lower, any capital but first letter) and Part-of-Speech tags.[3] We randomly initialize feature vectors and train them together as other parameters.

Token LSTM Layer We apply a bidirectional LSTM at the token level taking the concatenated character-word-feature embedding as input. The token representation obtained by stacking the forward and backward LSTM hidden states is passed as input to a linear layer that project the dimension to the size of label type space and is used as input to CRF layer.

CRF Layer Keyphrase categorization is a task where there is strong dependencies across output labels (e.g., I-TASK cannot follow B-Process). Therefore, instead of making independent tagging decisions for each output, we model them jointly using conditional random field (Lafferty et al., 2001). For an input sentence $\boldsymbol{x} = (x_1, x_2, x_3, \ldots, x_n)$, we consider P to be the matrix of scores output by the bidirectional LSTM network. P is of size $n \times m$, where n is the number of tokens in a sentence, and m is the number of distinct tags. $P_{t,i}$ corresponds to the score of

[2] Due to the difficulty of data collection, experiments with external data from the other two domains is left to future work.

[3] Dependency features were investigated but did not lead to performance gains.

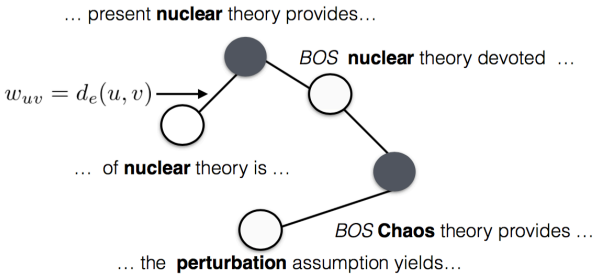

Figure 2: Label propagation. Gray nodes indicates labeled data while white nodes are unlabeled. Bold font word indicates the current token. The assumption is if two instances are similar according to the graph, the output labels should be similar.

the i-th tag of the t-th word in a sentence. We use a first-order Markov Model and define a transition matrix $\boldsymbol{T}$ where $\boldsymbol{T}_{i,j}$ represents the score from tag i to tag j. We also add y_0 and y_n as the *start* and *end* tags of a sentence. Therefore $\boldsymbol{T}$ becomes a square matrix of dimension $m + 2$.

Given one possible output $\boldsymbol{y}$, and neural network parameters θ we define the score as

$$\phi(\boldsymbol{y}; \boldsymbol{x}, \theta) = \sum_{t=0}^{n} T_{y_t, y_{t+1}} + \sum_{t=1}^{n} P_{t, y_t} \quad (1)$$

The probability of sequence $\boldsymbol{y}$ is obtained by applying a softmax over all possible tag sequences

$$p_\theta(\boldsymbol{y}|\boldsymbol{x}) = \frac{\exp(\phi(\boldsymbol{y}; \boldsymbol{x}, \theta))}{\sum_{\boldsymbol{y}' \in Y} \exp(\phi(\boldsymbol{y}'; \boldsymbol{x}, \theta))} \quad (2)$$

where Y denotes all possible tag sequences. The normalization term is efficiently computed using the forward algorithm.

Supervised Training During training, we maximize the log-probability $\mathcal{L}(\boldsymbol{Y}; \boldsymbol{X}, \theta)$ of the correct tag sequence given the corpus $\{\boldsymbol{X}, \boldsymbol{Y}\}$. Back-propagation is done based on a gradient computed using sentence-level scores.

5 Semi-supervised Learning

We develop a semi-supervised algorithm that extends self-training by estimating the labels of unlabeled data and then using those labels for re-training. Specifically, we use a graph-based algorithm to estimate the posterior probabilities of unlabeled data and develop a new CRF training to take the uncertainty of the estimated labels into account while optimizing the objective function.

5.1 Graph-based Posterior Estimates

Our semi-supervised algorithm uses the following steps to estimate the posterior. It first constructs a graph of tokens based on their semantic similarity, then uses the CRF marginal as a regularization term to do label propagation on the graph. The smoothed posterior is then used to either interpolate with the CRF marginal or as an additional feature to the neural network.

Graph Construction Vertices in the graph correspond to tokens, and edges are distance between token features which capture semantic similarity. The total size of the graph is equal to the number of tokens in both labeled data V_l and unlabeled data V_u. The tokens are modelled with a concatenation of pre-trained word embeddings (with dimension d) of 5-gram centered by the current token, the word embedding of the closest verb, and a set of discrete features including part-of-speech tags and capitalization (43 and 4 dimension one-hot features). The resulting feature vector with dimension of $5d + d + 43 + 4$ is then projected down to 100 dimensions using PCA. We define the weight w_{uv} of the edge between nodes u and v as follows: $w_{uv} = d_e(u, v)$ if $v \in \mathcal{K}(u)$ or $u \in \mathcal{K}(v)$, where $\mathcal{K}(u)$ is the set of k-nearest neighbors of u and $d_e(u, v)$ is the Euclidean distance between any two nodes u and v in the graph. An example of our graph is in Fig. 2.

For every node i in the graph, we compute the marginal probabilities $\{\boldsymbol{q}_i\}$ using the forward-backward algorithm. Let θ^i represent the estimate of the CRF parameters after the n-th iteration, we compute the marginal probabilities $\tilde{\boldsymbol{p}}_{(j,t)} = p(y_t^j|\boldsymbol{x}; \theta^i)$ over IOBES tags for every token position t in sentence j in labeled and unlabeled data.

Label Propagation We use prior-regularized measure propagation (Liu and Kirchhoff, 2014; Subramanya and Bilmes, 2011) to propagate labels from the annotated data to their neighbors in the graph. The algorithm aims for the label distribution between neighboring nodes to be as similar to each other as possible by optimizing an objective function that minimizes the Kullback-Leibler distances between: i) the empirical distribution $\boldsymbol{r}_u$ of labeled data and the predicted label distribution $\boldsymbol{q}_u$ for all labeled nodes in the graph; ii) the distributions $\boldsymbol{q}_u$ and $\boldsymbol{q}_v$ for all nodes u in the graph and their neighbors v; iii) the distributions $\boldsymbol{q}_u$ and the CRF marginals $\tilde{\boldsymbol{p}}_u$ for all nodes. The third term regularizes the predicted distribution toward the

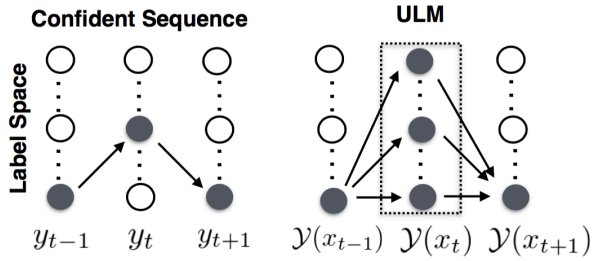

Figure 3: Lattice representation of ULM. Dashed box is the uncertain token which is going to be marginalized over. Arrows and grey nodes are paths to be summed over during training. When all tokens are confident, the score of only one path is calculated.

CRF prediction if the node is not connected to a labeled vertex, ensuring the algorithm performs at least as well as standard self-training.

Posterior Estimates We develop two strategies to estimate the new posteriors $\hat{p}(y_t|\boldsymbol{x};\theta)$, which can then be used in the CRF training.

The first strategy (called GRAPHINTERP) is the commonly used approach (Subramanya et al., 2010; Aliannejadi et al., 2014) that interpolates the smoothed posterior $\{\boldsymbol{q}\}$ with CRF marginals p:

$$\hat{p}(y_t|\boldsymbol{x};\theta) = \alpha p(y_t|\boldsymbol{x};\theta) + (1-\alpha)q(y) \quad (3)$$

where α is a mixing coefficient.

A second strategy introduced here (called GRAPHFEAT) uses the smoothed posterior $\{\boldsymbol{q}\}$ as features and learns it with other parameters in the neural network. Given a sentence $\{x_1,\ldots,x_n\}$, let $Q = \{\boldsymbol{q}_1,\ldots,\boldsymbol{q}_n\}$ be the predicted label distribution from the graph. We then use Q as a feature input to neural network as $\tilde{P} = P + MQ$ where P is the $n \times m$ matrix output by the bidirectional LSTM network as in Eq. 1, and M is $m \times m$ matrix and is learned together with other parameters of neural network. We modify Eq. 1 by replacing P_{t,y_t} with $\tilde{P}_{t,y_t}$. Note that GRAPHFEAT can only be done in a transductive way since it requires output Q from the graph at test time.

5.2 CRF training with Uncertain Labels

A standard approach to self-training is to make hard decisions for labeling tokens based on the estimated posteriors and retrain the model. However, the estimated posteriors in our task are noisy due to the difficulty and variety of the ScienceIE task. Instead, we extend the CRF training to leverage the confidence of the estimated posteriors. The new CRF training (called Uncertain Label Marginalizing (ULM)) treats low confidence tokens as missing labels and computes the sentence-level score by marginalizing over them. A similar idea has been previously used in treating partially labeled data (Kim et al., 2015).

Specifically, given a sentence $\boldsymbol{x}$ we define a constrained *lattice* $\mathcal{Y}(\boldsymbol{x})$, where at each position t the allowed label types $\mathcal{Y}(x_t)$ are:

$$\mathcal{Y}(x_t) = \begin{cases} \{y_t\}, & \text{if} \quad p(y_t|\boldsymbol{x};\theta) > \eta \\ \text{All label types}, & \text{otherwise} \end{cases} \quad (4)$$

where η is the confidence threshold, y_t is the prediction of posterior decoding and $p(y_t|\boldsymbol{x};\theta)$ is its CRF token marginal. The new neural network parameters θ are estimated by maximizing the log-likelihood of $p_\theta(\mathcal{Y}(\boldsymbol{x}^k)|\boldsymbol{x}^k)$ for every input sentence $\boldsymbol{x}^k$, where

$$p_\theta(\mathcal{Y}(\boldsymbol{x}^k)|\boldsymbol{x}^k) = \frac{\sum_{\boldsymbol{y}^k \in \mathcal{Y}(\boldsymbol{x}^k)} \exp(\phi(\boldsymbol{y}^k;\boldsymbol{x}^k,\theta))}{\sum_{\boldsymbol{y}' \in Y} \exp(\phi(\boldsymbol{y}';\boldsymbol{x},\theta))}$$

where $\boldsymbol{y}^k$ is an instance sequence of lattice $\mathcal{Y}(\boldsymbol{x})$, and k is the sentence index in the training set. Extreme cases are when all tokens are uncertain then the likelihood would be equal to 1, when all tokens of a sequence are confident, it would be equal to Eq. 2 where only one possible sequence, as in Fig. 3.

Inductive and Transductive Learning The semi-supervised training process is summarized as follow: It first computes marginals over the unlabeled data given a set of CRF parameters. It then uses the marginals as a regularization term for label propagation. The smoothed posteriors from the graph are then interpolated with the CRF marginal in GRAPHINTERP or used as an additional feature in GRAPHFEAT. It then uses the estimated labels for the unlabeled data combined with the labeled data to retrain the CRF using either the hard decision CRF training objective as Eq. 2 or the ULM data selection objective.

In the inductive setting, we only use the unlabeled data from the development set for the semi-supervision. In the transductive setting we also use the unlabeled data of the test set to construct the graph. In both cases, the parameters are tuned only on the dev set.

6 Experimental Setup

Data The SemEval ScienceIE (SE) corpus consists of 500 journal articles; one paragraph of each

Span Level	Classification (dev)	Classification (test)	Identification
Gupta et.al.(unsupervised)	-	9.8	6.4
Tsai et.al. (unsupervised)	-	11.9	8.0
MULTITASK	45.5	-	-
Best Non-Neural SemEval^{+}	-	38	51
Best Neural SemEval^{+}	-	44	56
NN-CRF(supervised)	48.1	40.2	52.1
NN-CRF(semi)	51.9	45.3	56.9
NN-CRF(semi)*	**52.1**	**46.6**	**57.6**

Table 1: Overall span-level F1 results for keyphrase identification (SemEval Subtask A) and classification (SemEval Subtask B). * indicates tranductive setting. $^{+}$ indicates not documented as either transductive or inductive. - indicates score not reported or not applied.

Model	P	R	F1
NN-CRF(supervised)	46.2	48.2	47.2
No features	44.2	46.1	45.1
No bi-LSTM	45.2	44.7	44.9
No CRF	36.7	38.2	37.4
No char	45.7	46.2	45.9

Table 2: Ablation study showing impact of neural network configurations of our NN-CRF(supervised) model on the dev set.

article is randomly selected and annotated. The complete unlabeled articles and their metadata are provided together with the labeled data. The training data consists of 350 documents; 50 are kept for development and 100 for testing. The 500 articles come from 82 different journals evenly distributed in three domains. We manually labeled 82 journal names in the dataset into the three domains and do analysis based on the domain partitions. The 500 full articles contains 2M words and is 30 times the size of the annotated data.

Additionally, we use two external resources for pretraining word embeddings: i) WIKI, as for Wikipedia articles, specifically a full Wikipedia dump from 2012 containing 46M words, and ii) ACM, a collection of CS papers, containing 108M words.

Comparisons We compare our system with two template matching baselines and the state-of-the-art on the SemEval Science IE task. The first baseline (Gupta and Manning, 2011) is an unsupervised method to extract keyphrases by initially using seed patterns in a dependency tree, and then adding to seed patterns through bootstrapping. The second baseline (Tsai et al., 2013) improves the work of Gupta and Manning (2011) by adding Named Entity Features and use different set of seed patterns.

Implementation details All parameters are tuned on the dev set performance, the best parameters are selected and fixed for model switching and semi-supervised systems. The word embedding dimension is 250; the token-level hidden dimension is 100; the character-level hidden dimension is 25; and the optimization algorithm is SGD with a learning rate of 0.05. For building the graph, the best pre-trained embeddings for the supervised system (Sec. 7.2) are used in each domain. Two special tokens *BOS* and *EOS* are added when pre-training, indicating the begin and end of a sentence. The number of the graph vertices is 2M in tranductive setting and 1.4M in inductive setting. The ULM parameter η in Eq. 4 is tuned from 0.1 to 0.9, the best η is 0.4. The best parameters of label propagation are $\mu = 10^{-6}$ and $\nu = 10^{-5}$. The interpolation parameter α in Eq. 3 is tuned from 0.1 to 0.9, the best α is 0.3. We do iteration of semi-supervised learning until we obtain the best result on the dev set, which is mostly achieved in the second round.

We use Stanford CoreNLP (Manning et al., 2014) tokenizer to tokenize words. The tokenizer is augmented with a few hand-designed rules to handle equations (e.g. "fs(B,t)=Spel(t)S" is a single token) and other non-standard word phenomena (Cu40Zn, 20MW/m2) in scientific literature. We use Approximate Nearest Neighbor Searching (ANN)[4] to calculate the k-nearest neighbors. For all experiments in this paper, $k = 10$.

Setup We evaluate our system in both inductive and transductive settings. The systems with a * superscript in the table are transductive. The inductive setting uses 400 full articles in ScienceIE training and dev sets, while the transductive setting uses 500 full articles including the test set. In both settings parameters are tuned over the dev set.

[4] https://www.cs.umd.edu/~mount/ANN/

7 Experimental Results

We evaluate our NN-CRF model in both supervised and semi-supervised settings. We also perform ablations and try different variants to best understand our model.

7.1 Best Case System Performance

Table 1 reports the results of our neural sequence tagging model NN-CRF in both supervised and semi-supervised learning (ULM and graph-based), and compares them with the baselines and the state-of-the-art (best SemEval System (Augenstein et al., 2017)).

Augenstein and Søgaard (2017) use a multi-task learning strategy to improve the performance of supervised keyphrase classification, but they only report dev set performance on SemEval Task 10, we also include their result here and refer it as MULTITASK. We report results for both span identification (SemEval SubTask A) and span classification into TASK, PROCESS and MATERIAL (SemEval Subtask B).[5]

The results show that our neural sequence tagging models significantly outperforms the state of the art and both baselines. It confirms that our neural tagging model outperforms other non-neural and neural models for the SemEval ScienceIE challenge[6]. It further shows that our system achieves significant boost from semi-supervised learning using unlabeled data. Table 5 shows the detailed analysis of the system across different categories.

7.2 Supervised Learning

Impact of Neural Model Components Table 2 provides the results of an ablation study on the dev set showing the impact of different components of our NN-CRF on the Scientific IE task. For the basic model, the word embeddings are initialized by word2vec trained on the 350 full journal articles in the SE training set together with Wikipedia and ScienceIE data. The feature layer, character layer, and bi-LSTM word layers all improves the performance. Moreover, we observe a large improvement (20.6% relative) in the scientific IE task by adding the CRF layer.

Initialization Table 3 reports our NN-CRF performance when pretrained on different domains. We explore different word embedding pretraining with ScienceIE training set alone (SE), and adding other external resources including Wikipedia (wiki) and Computer Science articles (ACM). All alternatives use word2vec. Compared with using SE alone, introduction of all external data sources improve performance. Moreover, we observe that with the introduction of the ACM dataset, the performance on the CS domain is increased significantly in both the dev and test sets. Adding Wikipedia data benefits all three domains, with more significant improvement on the MS and Physics domains.

	Dev			Test		
Initialization	MS	Phy	CS	MS	Phy	CS
SE	49.4	39.4	45.0	42.9	33.0	30.5
+wiki	**52.9**	**40.5**	47.9	**46.1**	**39.2**	31.0
+ACM	50.3	39.8	**49.5**	42.2	37.8	34.2
+wiki+ACM	50.5	40.3	48.9	43.1	37.9	**34.4**

Table 3: F1 score on the dev and test sets for using different sources of data for pretraining.

Based on these observations, we select the best model on each domain according to the dev set and use the combined result as our best suprevised system (called NN-CRF(supervised)). The F1 score improves from 39.4 to 40.2 when applying model switching strategy. The best model on the dev set is used for each domain: for MS and physics domain, we pretrain word embeddings with the SE and Wiki, and for the CS domain, we pretrain with the SE and ACM.

7.3 Semi-Supervision Learning

Table 4 reports the results of the semi-supervised learning algorithms in different settings. In particular we ablate incorporating the graph-based methods of computing the posterior and CRF training (ULM vs. hard decision). The table shows incorporating graph-based methods for computing posterior and ULM for CRF training outperforms their counterparts.

For computing the posterior, we explore two different strategies of the graph-based methods: i) GRAPHINTERP that interpolates the smoothed posterior from label propagation with CRF marginals; For inductive setting, GRAPHINTERP only uses un-annotated data from the dev set and uses the best model for decoding at test time. For transductive setting, GRAPHINTERP* uses unannoated data from test set to build the graph as

[5]The evaluation script is provided by the challenge, with a modification to report 3 decimal precision results.

[6]Best SemEval Numbers from https://scienceie.github.io/

Posterior	Training	Dev	Test
-	-	50.2	42.9
-	ULM	51.3	44.4
GRAPHINTERP	-	50.9	43.3
GRAPHINTERP	ULM	**51.9**	**45.3**
GRAPHINTERP*	-	50.7	44.0
GRAPHINTERP*	ULM	51.8	45.7
GRAPHFEAT*	-	51.4	44.9
GRAPHFEAT*	ULM	**52.1**	**46.6**

Table 4: F1 scores of semi-supervised Learning approaches; * shows transductive models.

Span Level	T	P	M	K
Best SemEval	19	44	48	55
supervised	13.3	40.5	43.7	52.1
ULM+GRAPHINTERP	17.0	45.4	49.4	56.9
ULM+GRAPHFEAT*	17.2	46.5	50.7	57.6
Token Level	T	P	M	K
supervised	29.6	56.0	59.3	70.8
ULM+GRAPHINTERP	40.0	60.7	61.2	77.0
ULM+GRAPHFEAT*	40.1	62.8	63.4	78.1

Table 5: F1 score results on the test set for different categories: T indicates TASK, P indicates PROCESS, M is MATERIAL and K is Keyword identification (SubTask A). * is transductive model.

well, and tune the parameters on the dev set. ii) GRAPHFEAT uses the smoothed posterior from label propagation as additional feature to neural network and only has transductive setting.

As expected, the transductive approaches consistently outperform inductive approaches on the test set. With around the same performance on dev set, GRAPHINTERP* seems to generalize better on test set with 1.6% relative improvement over GRAPHINTERP. We observe higher improvement with GRAPHFEAT* compared to GRAPHINTERP. This is mainly because automatically learning the weight matrix M between neural network scores and graph outputs adds more flexibility compared to tuning an interpolation weight α. The performance is further improved by applying data selection through modifying the objective to ULM. The best inductive system is ULM+GRAPHINTERP with 5.6% relative improvement over pure Self-Training that makes hard decisions, and the best transductive system is ULM+GRAPHFEAT* with 8.6% relative improvement.

7.4 Category and Span Analysis

Table 5 details the performance of our method on the three categories at the span and token level. We observe significant improvement by using ULM+GRAPHINTERP and ULM+GRAPHFEAT over best SemEval and our best supervised system on all three categories at both token and span levels. We further observe that systems' performance on TASK classification is much lower than PROCESS and MATERIAL. This is in part because TASK is much less frequent than the other types. In addition, TASK keyphrases often include verb phrases while the other two domains mainly consists of noun phrases. An analysis of confusion patterns show that the most frequent type confusions are between PROCESS and MATERIAL. However, we observe that ULM+GRAPHFEAT* can greatly reduce the confusion, with 3.5% relative improvement of PROCESS and 3.6% relative improvement of PROCESS over ULM+GRAPHINTERP on token level.

7.5 Error Analysis

We provide examples of typical errors that our system makes in Table 6. As described in the previous subsection, TASK is the hardest type to identify with our system. Row 1 shows a failure to detect the verb phrase following 'to' as part of the TASK, but detect 'enantiopure products' as MATERIAL. The system prefers to predict PROCESS or MATERIAL since those classes have more samples than TASK. Row 2 illustrates the problem of identifying general terms as keyphrases due to similar context, such as 'receptors' and 'drug action'. A third common error involves incorrectly labeling adjectives, such as 'neighbouring' in Row 3, which leads to span errors. Another common cause of error is insufficient context: in the last example, a larger context is needed to determine whether 'SWE' is a PROCESS or MATERIAL.

8 Conclusion

This paper casts the scientific information extraction task as a sequence tagging problem and introduces a hierarchical LSTM-CRF neural tagging model for this task, building on recent results in NER. We introduced a semi-supervised learning algorithm that incorporates graph-based label propagation and confidence-aware data selection. We show the introduction of semi-supervision significantly outperforms the performance of the supervised LSTM-CRF tagging model. We additionally show that external resources are useful for initializing word embeddings. Both inductive and transductive semi-supervised strategies

Error types	Annotation and System Output
Verb phrases	A key requirement in aiming to [achieve [enantiopure products]$_{\text{Material}}$]$_{\text{Task}}$ is therefore a means to [quantitate [the enantiometric excess]$_{\text{Process}}$]$_{\text{Task}}$.
General terms	Since the [receptors]$_{\text{Material}}$ in human biology mostly consist of [chiral molecules]$_{\text{Material}}$, [drug action]$_{\text{Process}}$ mostly involves a specified enantiometric form.
Falsely predicted adjectives	It has been shown that the most efficient forms of energy transfer between the two occurs when there is a [neighbouring carotenoid species]$_{\text{Material}}$.
Lack of context	Other models use [SWEs]$^{\text{Material}}_{\text{Process}}$ but focus on the use of multi resolution grids or irregular mesh.

Table 6: Common errors, where blue means golden label our system misses, red means falsely predicted results, and green means correctly predicted spans.

achieve state-of-the-art performance in SemEval 2017 ScienceIE task. We also conducted a detailed analysis of the system and point out common error cases.

In our experiments, we observe that including in-domain data only for semi-supervised learning has slightly better performance than using cross-domain data. Reducing the amount of in-domain data hurts performance. Therefore, adding more in-domain unlabeled data may help when combined with selection schemes such as the ULM algorithms proposed here. It would be useful to assess the impact of matched unlabeled data for the physics and material science domain. Other future work includes leveraging global context, information of citation network.

9 Acknowledgments

This research was supported by the NSF (IIS 1616112), Allen Institute for AI (66-9175), Allen Distinguished Investigator Award, and gifts from Google, Samsung, and Bloomberg. We thank the anonymous reviewers for their helpful comments.

References

Amjad Abu-Jbara and Dragomir Radev. 2011. Coherent citation-based summarization of scientific papers. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, pages 500–509.

Mohammad Aliannejadi, Masoud Kiaeeha, Shahram Khadivi, and Saeed Shiry Ghidary. 2014. Graph-based semi-supervised conditional random fields for spoken language understanding using unaligned data. In *Australasian Language Technology Association Workshop*. page 98.

Ashton Anderson, Dan McFarland, and Dan Jurafsky. 2012. Towards a computational history of the ACL: 1980-2008. In *Proceedings of the ACL-2012 Special Workshop on Rediscovering 50 Years of Discoveries*. Association for Computational Linguistics, pages 13–21.

Awais Athar and Simone Teufel. 2012a. Context-enhanced citation sentiment detection. In *Proceedings of the 2012 conference of the North American chapter of the Association for Computational Linguistics: Human language technologies*. Association for Computational Linguistics, pages 597–601.

Awais Athar and Simone Teufel. 2012b. Detection of implicit citations for sentiment detection. In *Proceedings of the Workshop on Detecting Structure in Scholarly Discourse*. Association for Computational Linguistics, pages 18–26.

Isabelle Augenstein, Mrinal Das, Sebastian Riedel, Lakshmi Vikraman, and Andrew McCallum. 2017. Semeval 2017 task 10: ScienceIE - extracting keyphrases and relations from scientific publications. *Proceedings of SemEval* .

Isabelle Augenstein and Anders Søgaard. 2017. Multi-task learning of keyphrase boundary classification. In *arXiv preprint arXiv:1704.00514*.

Miguel Ballesteros, Chris Dyer, and Noah A Smith. 2015. Improved transition-based parsing by modeling characters instead of words with LSTMs. In *EMNLP*.

Steven Bird, Robert Dale, Bonnie J Dorr, Bryan R Gibson, Mark Thomas Joseph, Min-Yen Kan, Dongwon Lee, Brett Powley, Dragomir R Radev, Yee Fan Tan, et al. 2008. The ACL anthology reference corpus: A reference dataset for bibliographic research in computational linguistics. In *LREC*.

Jason PC Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional lstm-cnns. In *TACL*.

Michael Collins and Yoram Singer. 1999. Unsupervised models for named entity classification. In *Proceedings of the joint SIGDAT conference on empirical methods in natural language processing and very large corpora*. Citeseer, pages 100–110.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research* 12(Aug):2493–2537.

Andrew M Dai and Quoc V Le. 2015. Semi-supervised sequence learning. In *Advances in Neural Information Processing Systems*. pages 3079–3087.

Huy Hoang Nhat Do, Muthu Kumar Chandrasekaran, Philip S Cho, and Min Yen Kan. 2013. Extracting and matching authors and affiliations in scholarly documents. In *Proceedings of the 13th ACM/IEEE-CS joint conference on Digital libraries*. ACM, pages 219–228.

Kata Gabor, Haifa Zargayouna, Davide Buscaldi, Isabelle Tellier, and Thierry Charnois. 2016. Semantic annotation of the ACL anthology corpus for the automatic analysis of scientific literature. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. Paris, France.

Sonal Gupta and Christopher D Manning. 2011. Analyzing the dynamics of research by extracting key aspects of scientific papers. In *IJCNLP*. pages 1–9.

Bret Harsham, Shinji Watanabe, Alan Esenther, John Hershey, Jonathan Le Roux, Yi Luan, Daniel Nikovski, and Vamsi Potluru. 2015. Driver prediction to improve interaction with in-vehicle hmi. In *Proc. Workshop on Digital Signal Processing for In-Vehicle Systems (DSP)*.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. In *arXiv preprint arXiv:1508.01991*.

Kokil Jaidka, Muthu Kumar Chandrasekaran, Beatriz Fisas Elizalde, Rahul Jha, Christopher Jones, Min-Yen Kan, Ankur Khanna, Diego Molla-Aliod, Dragomir R Radev, Francesco Ronzano, et al. 2014. The computational linguistics summarization pilot task. *Proceedings of TAC* .

Miray Kas. 2011. Structures and statistics of citation networks. Technical report, DTIC Document.

Young-Bum Kim, Minwoo Jeong, Karl Stratos, and Ruhi Sarikaya. 2015. Weakly supervised slot tagging with partially labeled sequences from web search click logs. In *HLT-NAACL*. pages 84–92.

John Lafferty, Andrew McCallum, Fernando Pereira, et al. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the eighteenth international conference on machine learning, ICML*. volume 1, pages 282–289.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *NAACL*.

Gina-Anne Levow, Valerie Freeman, Alena Hrynkevich, Mari Ostendorf, Richard Wright, Julian Chan, Yi Luan, and Trang Tran. 2014. Recognition of stance strength and polarity in spontaneous speech. In *Spoken Language Technology Workshop (SLT), 2014 IEEE*. IEEE, pages 236–241.

Omer Levy and Yoav Goldberg. 2014. Dependency-based word embeddings. In *ACL*. pages 302–308.

Yuzong Liu and Katrin Kirchhoff. 2013. Graph-based semi-supervised learning for phone and segment classification. In *Proceedings of Annual Conference of the International Speech Communication Association (Interspeech)*.

Yuzong Liu and Katrin Kirchhoff. 2014. Graph-based semi-supervised acoustic modeling in DNN-based speech recognition. In *IEEE SLT*.

Yuzong Liu and Katrin Kirchhoff. 2015. Acoustic modeling with neural graph embeddings. In *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*.

Yuzong Liu and Katrin Kirchhoff. 2016a. Graph-based semisupervised learning for acoustic modeling in automatic speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 24(11):1946–1956.

Yuzong Liu and Katrin Kirchhoff. 2016b. Novel front-end features based on neural graph embeddings for DNN-HMM and LSTM-CTC acoustic modeling. In *Proceedings of Annual Conference of the International Speech Communication Association (Interspeech)*. ISCA.

Yi Luan, Yangfeng Ji, Hannaneh Hajishirzi, and Boyang Li. 2016a. Multiplicative representations for unsupervised semantic role induction. In *The 54th Annual Meeting of the Association for Computational Linguistics*. page 118.

Yi Luan, Yangfeng Ji, and Mari Ostendorf. 2016b. Lstm based conversation models. In *arXiv preprint arXiv:1603.09457*.

Yi Luan, Daisuke Saito, Yosuke Kashiwagi, Nobuaki Minematsu, and Keikichi Hirose. 2014a. Semi-supervised noise dictionary adaptation for exemplar-based noise robust speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, pages 1745–1748.

Yi Luan, Masayuki Suzuki, Yutaka Yamauchi, Nobuaki Minematsu, Shuhei Kato, and Keikichi Hirose. 2012. Performance improvement of automatic pronunciation assessment in a noisy classroom. In *Spoken Language Technology Workshop (SLT), 2012 IEEE*. IEEE, pages 428–431.

Yi Luan, Shinji Watanabe, and Bret Harsham. 2015. Efficient learning for spoken language understanding tasks with word embedding based pre-training. In *INTERSPEECH*. Citeseer, pages 1398–1402.

Yi Luan, Richard Wright, Mari Ostendorf, and Gina-Anne Levow. 2014b. Relating automatic vowel space estimates to talker intelligibility. In *Fifteenth Annual Conference of the International Speech Communication Association*.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *ACL*.

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The stanford CoreNLP natural language processing toolkit. In *ACL (System Demonstrations)*. pages 55–60.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *arXiv preprint arXiv:1301.3781*.

Nanyun Peng and Mark Dredze. 2015. Named entity recognition for chinese social media with jointly trained embeddings. In *EMNLP*. pages 548–554.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*. volume 14, pages 1532–1543.

Behrang QasemiZadeh and Anne-Kathrin Schumann. 2012. The acl rd-tec 2.0: A language resource for evaluating term extraction and entity recognition methods. In *LREC*.

Yanchuan Sim, Noah A Smith, and David A Smith. 2012. Discovering factions in the computational linguistics community. In *Proceedings of the ACL-2012 Special Workshop on Rediscovering 50 Years of Discoveries*. Association for Computational Linguistics, pages 22–32.

Amarnag Subramanya and Jeff Bilmes. 2011. Semi-supervised learning with measure propagation. *Journal of Machine Learning Research* 12(Nov):3311–3370.

Amarnag Subramanya, Slav Petrov, and Fernando Pereira. 2010. Efficient graph-based semi-supervised learning of structured tagging models. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 167–176.

Chen-Tse Tsai, Gourab Kundu, and Dan Roth. 2013. Concept-based analysis of scientific literature. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*. ACM, pages 1733–1738.

Adam Vogel and Dan Jurafsky. 2012. He said, she said: Gender in the ACL anthology. In *Proceedings of the ACL-2012 Special Workshop on Rediscovering 50 Years of Discoveries*. Association for Computational Linguistics, pages 33–41.