Sentiment Tagging with Partial Labels using Modular Architectures

Xiao Zhang
Purdue University
zhang923@purdue.edu

Dan Goldwasser
Purdue University
dgoldwas@purdue.edu

Abstract

Many NLP learning tasks can be decomposed into several distinct sub-tasks, each associated with a *partial* label. In this paper we focus on a popular class of learning problems, sequence prediction applied to several sentiment analysis tasks, and suggest a modular learning approach in which different sub-tasks are learned using separate functional modules, combined to perform the final task while sharing information. Our experiments show this approach helps constrain the learning process and can alleviate some of the supervision efforts.

1 Introduction

Many natural language processing tasks attempt to replicate complex human-level judgments, which often rely on a composition of several sub-tasks into a unified judgment. For example, consider the Targeted-Sentiment task (Mitchell et al., 2013), assigning a sentiment polarity score to entities depending on the context that they appear in. Given the sentence *"according to a CNN poll, Green Book will win the best movie award"*, the system has to identify both entities, and associate the relevant sentiment value with each one (neutral with *CNN*, and positive with *Green Book*). This task can be viewed as a combination of two tasks, entity identification, locating contiguous spans of words corresponding to relevant entities, and sentiment prediction, specific to each entity based on the context it appears in. Despite the fact that this form of functional task decomposition is natural for many learning tasks, it is typically ignored and learning is defined as a monolithic process, combining the tasks into a single learning problem.

Our goal in this paper is to take a step towards modular learning architectures that exploit the learning tasks' inner structure, and as a result simplify the learning process and reduce the annotation effort. We introduce a novel task decomposition approach, *learning with partial labels*, in which the task output labels decompose hierarchically, into partial labels capturing different aspects, or sub-tasks, of the final task. We show that learning with partial labels can help support weakly-supervised learning when only some of the partial labels are available.

Given the popularity of sequence labeling tasks in NLP, we demonstrate the strength of this approach over several sentiment analysis tasks, adapted for sequence prediction. These include target-sentiment prediction (Mitchell et al., 2013), aspect-sentiment prediction (Pontiki et al., 2016) and subjective text span identification and polarity prediction (Wilson et al., 2013). To ensure the broad applicability of our approach to other problems, we extend the popular LSTM-CRF (Lample et al., 2016) model that was applied to many sequence labeling tasks[1].

The modular learning process corresponds to a task decomposition, in which the prediction label, y, is deconstructed into a set of partial labels $\{y^0, .., y^k\}$, each defining a sub-task, capturing a different aspect of the original task. Intuitively, the individual sub-tasks are significantly easier to learn, suggesting that if their dependencies are modeled correctly when learning the final task, they can constrain the learning problem, leading to faster convergence and a better overall learning outcome. In addition, the modular approach helps alleviate the supervision problem, as often providing full supervision for the overall task is costly, while providing additional partial labels is significantly easier. For example, annotating entity segments syntactically is considerably easier than determining their associated sentiment, which requires understanding the nuances of the

[1] We also provide analysis for NER in the apendix

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 579–590
Florence, Italy, July 28 - August 2, 2019.

context they appear in semantically. By exploiting modularity, the entity segmentation partial labels can be used to help improve that specific aspect of the overall task.

Our modular task decomposition approach is partially inspired by findings in cognitive neuroscience, namely the *two-streams hypothesis*, a widely accepted model for neural processing of cognitive information in vision and hearing (Eysenck and Keane, 2005), suggesting the brain processes information in a modular way, split between a "where" (dorsal) pathway, specialized for locating objects and a "what" (ventral) pathway, associated with object representation and recognition (Mishkin et al., 1983; Geschwind and Galaburda, 1987; Kosslyn, 1987; Rueckl et al., 1989). Jacobs et al. (1991) provided a computational perspective, investigating the "what" and "where" decomposition on a computer vision task. We observe that this task decomposition naturally fits many NLP tasks and borrow the notation. In the target-sentiment tasks we address in this paper, the segmentation tagging task can be considered as a "where"-task (i.e., the location of entities), and the sentiment recognition as the "what"-task.

Our approach is related to multi-task learning (Caruana, 1997), which has been extensively applied in NLP (Toshniwal et al., 2017; Eriguchi et al., 2017; Collobert et al., 2011; Luong, 2016; Liu et al., 2018). However, instead of simply aggregating the objective functions of several *different* tasks, we suggest to *decompose a single task into multiple inter-connected sub-tasks* and then integrate the representation learned into a single module for the final decision. We study several modular neural architectures, which differ in the way information is shared between tasks, the learning representation associated with each task and the way the dependency between decisions is modeled.

Our experiments were designed to answer two questions. *First*, can the task structure be exploited to simplify a complex learning task by using a modular approach? *Second*, can partial labels be used effectively to reduce the annotation effort?

To answer the first question, we conduct experiments over several sequence prediction tasks, and compare our approach to several recent models for deep structured prediction (Lample et al., 2016; Ma and Hovy, 2016; Liu et al., 2018), and when available, previously published results (Mitchell et al., 2013; Zhang et al., 2015; Li and Lu, 2017; Ma et al., 2018) We show that modular learning indeed helps simplify the learning task compared to traditional monolithic approaches. To answer the second question, we evaluate our model's ability to leverage partial labels in two ways. First, by restricting the amount of full labels, and observing the improvement when providing increasing amounts of partial labels for only one of the sub-tasks. Second, we learn the sub-tasks using completely disjoint datasets of partial labels, and show that the knowledge learned by the sub-task modules can be integrated into the final decision module using a small amount of full labels.

Our contributions: (1) We provide a general modular framework for sequence learning tasks. While we focus on sentiment analysis task, the framework is broadly applicable to many other tagging tasks, for example, NER (Carreras et al., 2002; Lample et al., 2016) and SRL (Zhou and Xu, 2015), to name a few. (2) We introduce a novel weakly supervised learning approach, *learning with partial labels*, that exploits the modular structure to reduce the supervision effort. (3) We evaluated our proposed model, in both the fully-supervised and weakly supervised scenarios, over several sentiment analysis tasks.

2 Related Works

From a technical perspective, our task decomposition approach is related to multi-task learning (Caruana, 1997), specifically, when the tasks share information using a shared deep representation (Collobert et al., 2011; Luong, 2016). However, most prior works aggregate multiple losses on either different pre-defined tasks at the final layer (Collobert et al., 2011; Luong, 2016), or on a language model at the bottom level (Liu et al., 2018). This work suggests to decompose a given task into sub-tasks whose integration comprise the original task. To the best of our knowledge, Ma et al. (2018), focusing on targeted sentiment is most similar to our approach. They suggest a joint learning approach, modeling a sequential relationship between two tasks, entity identification and target sentiment. We take a different approach viewing each of the model components as a separate module, predicted independently and then integrated into the final decision module. As we demonstrate in our experiments, this approach leads to better performance and increased flexibil-

ity, as it allows us to decouple the learning process and learn the tasks independently. Other modular neural architectures were recently studied for tasks combining vision and language analysis (Andreas et al., 2016; Hu et al., 2017; Yu et al., 2018), and were tailored for the grounded language setting. To help ensure the broad applicability of our framework, we provide a general modular network formulation for sequence labeling tasks by adapting a neural-CRF to capture the task structure. This family of models, combining structured prediction with deep learning showed promising results (Gillick et al., 2015; Lample et al., 2016; Ma and Hovy, 2016; Zhang et al., 2015; Li and Lu, 2017), by using rich representations through neural models to generate decision candidates, while utilizing an inference procedure to ensure coherent decisions. Our main observation is that modular learning can help alleviate some of the difficulty involved in training these powerful models.

3 Architectures for Sequence Prediction

Using neural networks to generate emission potentials in CRFs was applied successfully in several sequence prediction tasks, such as word segmentation (Chen et al., 2017), NER (Ma and Hovy, 2016; Lample et al., 2016), chunking and PoS tagging (Liu et al., 2018; Zhang et al., 2017). A sequence is represented as a sequence of L tokens: $\boldsymbol{x} = [x_1, x_2, \ldots, x_L]$, each token corresponds to a label $y \in \mathcal{Y}$, where $\mathcal{Y}$ is the set of all possible tags. An inference procedure is designed to find the most probable sequence $\boldsymbol{y}^* = [y_1, y_2, \ldots, y_L]$ by solving, either exactly or approximately, the following optimization problem:

$$\boldsymbol{y}^* = \arg\max_{\boldsymbol{y}} P(\boldsymbol{y}|\boldsymbol{x}).$$

Despite the difference in tasks, these models follow a similar general architecture: (1) Character-level information, such as prefix, suffix and capitalization, is represented through a character embedding layer learned using a bi-directional LSTM (BiLSTM). (2) Word-level information is obtained through a word embedding layer. (3) The two representations are concatenated to represent an input token, used as input to a word-level BiLSTM which generates the emission potentials for a succeeding CRF. (4) The CRF is used as an inference layer to generate the globally-normalized probability of possible tag sequences.

3.1 CRF Layer

A CRF model describes the probability of predicted labels $\boldsymbol{y}$, given a sequence $\boldsymbol{x}$ as input, as

$$P_{\boldsymbol{\Lambda}}(\boldsymbol{y}|\boldsymbol{x}) = \frac{e^{\Phi(\boldsymbol{x},\boldsymbol{y})}}{Z},$$

where $Z = \sum_{\tilde{\boldsymbol{y}}} e^{\Phi(\boldsymbol{x},\tilde{\boldsymbol{y}})}$ is the partition function that marginalize over all possible assignments to the predicted labels of the sequence, and $\Phi(\boldsymbol{x}, \boldsymbol{y})$ is the scoring function, which is defined as:

$$\Phi(\boldsymbol{x}, \boldsymbol{y}) = \sum_t \phi(\boldsymbol{x}, y_t) + \psi(y_{t-1}, y_t).$$

The partition function Z can be computed efficiently via the forward-backward algorithm. The term $\phi(\boldsymbol{x}, y_t)$ corresponds to the score of a particular tag y_t at position t in the sequence, and $\psi(y_{t-1}, y_t)$ represents the score of transition from the tag at position $t-1$ to the tag at position t. In the Neural CRF model, $\phi(\boldsymbol{x}, y_t)$ is generated by the aforementioned Bi-LSTM while $\psi(y_{t-1}, y_t)$ by a transition matrix.

4 Functional Decomposition of Composite Tasks

To accommodate our task decomposition approach, we first define the notion of partial labels, and then discuss different neural architectures capturing the dependencies between the modules trained over the different partial labels.

Partial Labels and Task Decomposition: Given a learning task, defined over an output space $y \in \mathcal{Y}$, where $\mathcal{Y}$ is the set of all possible tags, each specific label y is decomposed into a set of partial labels, $\{y^0, .., y^k\}$. We refer to y as the *full* label. According to this definition, a specific assignment to all k partial labels defines a single full label. Note the difference between *partially labeled data* (Cour et al., 2011), in which instances can have more than a single full label, and our setup in which the labels are partial.

In all our experiments, the partial labels refer to two sub-tasks, (1) a segmentation task, identifying *Beginning*, *Inside* and *Outside* of an entity or aspect. (2) one or more type recognition tasks, recognizing the aspect type and/or the sentiment polarity associated with it. Hence, a tag y_t at location t is divided into y_t^{seg} and y_t^{typ}, corresponding to segmentation and type (sentiment type here) respectively. Fig. 1 provides an example of the

target-sentiment task. Note that the sentiment labels do not capture segmentation information.

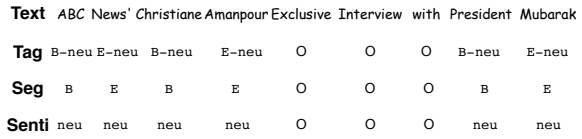


Figure 1: Target-sentiment decomposition example.

Modular Learning architectures: We propose three different models, in which information from the partial labels can be used. All the models have similar modules types, corresponding to the *segmentation* and *type* sub-tasks, and the decision module for predicting the final task. The modules are trained over the partial segmentation ($\boldsymbol{y}^{seg}$) and type ($\boldsymbol{y}^{typ}$) labels, and the full label $\boldsymbol{y}$ information, respectively.

These three models differ in the way they share information. **Model 1**, denoted *Twofold Modular, LSTM-CRF-T*, is similar in spirit to multi-task learning (Collobert et al., 2011) with three separate modules. **Model 2**, denoted *Twofold modular Infusion, (LSTM-CRF-TI*) and **Model 3**, denoted *Twofold modular Infusion with guided gating, (LSTM-CRF-TI(g))* both infuse information flow from two sub-task modules into the decision module. The difference is whether the infusion is direct or goes through a guided gating mechanism. The three models are depicted in Fig. 2 and described in details in the following paragraphs. In all of these models, underlying neural architecture are used for the emission potentials when CRF inference layers are applied on top.

4.1 Twofold Modular Model

The twofold modular model enhances the original monolithic model by using multi-task learning with shared underlying representations. The segmentation module and the type module are trained jointly with the decision module, and all the modules share information by using the same embedding level representation, as shown in Figure 2a. Since the information above the embedding level is independent, the LSTM layers in the different modules do not share information, so we refer to these layers of each module as *private*.

The segmentation module predicts the segmentation BIO labels at position t of the sequence by using the representations extracted from its private word level bi-directional LSTM (denoted as $\mathcal{H}^{seg}$) as emission for a individual CRF:

$$\boldsymbol{h}_t^{seg} = \mathcal{H}^{seg}(\boldsymbol{e}_t, \overrightarrow{\boldsymbol{h}}_{t-1}^{seg}, \overrightarrow{\boldsymbol{h}}_{t+1}^{seg}),$$
$$\phi(\boldsymbol{x}, y_t^{seg}) = \boldsymbol{W}^{seg\intercal}\boldsymbol{h}_t^{seg} + \boldsymbol{b}^{seg},$$

where $\boldsymbol{W}^{seg}$ and $\boldsymbol{b}^{seg}$ denote the parameters of the segmentation module emission layer, and $\mathcal{H}^{seg}$ denotes its private LSTM layer.

This formulation allows the model to forge the segmentation path privately through back-propagation by providing the segmentation information $\boldsymbol{y}^{seg}$ individually, in addition to the complete tag information $\boldsymbol{y}$.

The type module, using $\boldsymbol{y}^{typ}$, is constructed in a similar way. By using representations from the its own private LSTM layers, the type module predicts the sentiment (entity) type at position t of the sequence :

$$\boldsymbol{h}_t^{typ} = \mathcal{H}^{typ}(\boldsymbol{e}_t, \overrightarrow{\boldsymbol{h}}_{t-1}^{typ}, \overrightarrow{\boldsymbol{h}}_{t+1}^{typ}),$$
$$\phi(\boldsymbol{x}, y_t^{typ}) = \boldsymbol{W}^{typ\intercal}\boldsymbol{h}_t^{typ} + \boldsymbol{b}^{typ}.$$

Both the segmentation information $\boldsymbol{y}^{seg}$ and the type information $\boldsymbol{y}^{typ}$ are provided together with the complete tag sequence $\boldsymbol{y}$, enabling the model to learn segmentation and type recognition simultaneously using two different paths. Also, the decomposed tags naturally augment *more training data* to the model, avoiding over-fitting due to more complicated structure. The shared representation beneath the private LSTMs layers are updated via the back-propagated errors from all the three modules.

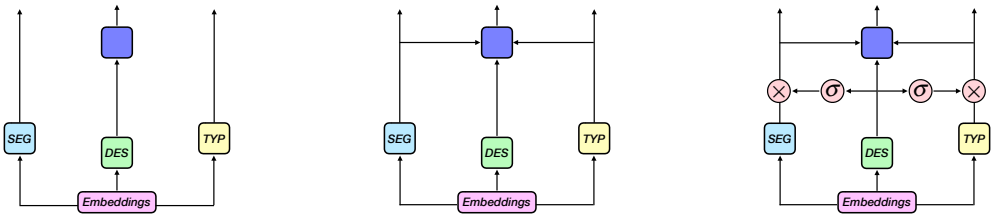
4.2 Two-fold Modular Infusion Model

The twofold modular infusion model provides a stronger connection between the functionalities of the two sub-tasks modules and the final decision module, differing from multi-task leaning.

In this model, instead of separating the pathways from the decision module as in the previous twofold modular model, the segmentation and the type representation are used as input to the final decision module. The model structure is shown in Figure 2b, and can be described formally as:

$$\boldsymbol{I}_t^{seg} = \boldsymbol{W}^{seg\intercal}\boldsymbol{h}_t^{seg} + \boldsymbol{b}^{seg},$$
$$\boldsymbol{I}_t^{typ} = \boldsymbol{W}^{typ\intercal}\boldsymbol{h}_t^{typ} + \boldsymbol{b}^{typ},$$
$$S_t = \boldsymbol{W}^{\intercal}[\boldsymbol{h}_t; \boldsymbol{I}_t^{seg}; \boldsymbol{I}_t^{typ}] + \boldsymbol{b},$$

where S_t is the shared final emission potential to the CRF layer in the decision module, and ; is the



(a) LSTM-CRF-T (b) LSTM-CRF-TI (c) LSTM-CRF-TI (G)

Figure 2: Three modular models for task decomposition. In them, blue blocks are *segmentation* modules, detecting entity location and segmentation, and yellow blocks are the *type* modules, recognizing the entity type or sentiment polarity. Green blocks are the final decision modules, integrating all the decisions. (G) refers to "Guided Gating"

concatenation operator, combining the representation from the decision module and that from the type module and the segmentation module.

The term *"Infusion"* used for naming this module is intended to indicate that both modules actively participate in the final decision process, rather than merely form two independent paths as in the twofold modular model. This formulation provides an alternative way of integrating the auxiliary sub-tasks back into the major task in the neural structure to help improve learning.

4.3 Guided Gating Infusion

In the previous section we described a way of infusing information from other modules naively by simply concatenating them. But intuitively, the hidden representation from the decision module plays an important role as it is directly related to the final task we are interested in. To effectively use the information from other modules forming sub-tasks, we design a gating mechanism to dynamically control the amount of information flowing from other modules by infusing the expedient part while excluding the irrelevant part, as shown in Figure 2c. This gating mechanism uses the information from the decision module to guide the information from other modules, thus we name it as guided gating infusion, which we describe formally as follows:

$$\begin{aligned} \boldsymbol{I}_t^{seg} &= \sigma(\boldsymbol{W_1} h_t + \boldsymbol{b_1}) \otimes (\boldsymbol{W}^{seg\intercal} \boldsymbol{h}_t^{seg} + \boldsymbol{b}^{seg}), \\ \boldsymbol{I}_t^{typ} &= \sigma(\boldsymbol{W_2} h_t + \boldsymbol{b_2}) \otimes (\boldsymbol{W}^{typ\intercal} \boldsymbol{h}_t^{typ} + \boldsymbol{b}^{typ}), \\ S_t &= \boldsymbol{W}^\intercal [\boldsymbol{h}_t; \boldsymbol{I}_t^{seg}; \boldsymbol{I}_t^{typ}] + \boldsymbol{b}, \end{aligned}$$

where σ is the logistic sigmoid function and $\otimes$ is the element-wise multiplication. The $\{W_1, W_2, b_1, b_2\}$ are the parameters of these guided gating, which are updated during the training to maximize the overall sequence labeling performance.

5 Learning using Full and Partial Labels

Our objective naturally rises from the model we described in the text. Furthermore, as our experiments show, it is easy to generalize this objective, to a "semi-supervised" setting, in which the learner has access to only a few fully labeled examples and additional partially labeled examples. E.g., if only segmentation is annotated but the type information is missing. The loss function is a linear combination of the negative log probability of each sub-tasks, together with the decision module:

$$\begin{aligned} \mathcal{J} = -\sum_i^N \log P(\boldsymbol{y}^i | \boldsymbol{x}^i) + \alpha \log P(\boldsymbol{y}^{seg(i)} | \boldsymbol{x}^{(i)}) \\ + \beta \log P(\boldsymbol{y}^{typ(i)} | \boldsymbol{x}^{(i)}), \end{aligned} \quad (1)$$

where N is the number of examples in the training set, $\boldsymbol{y}^{seg}$ and $\boldsymbol{y}^{typ}$ are the decomposed segmentation and type tags corresponding to the two sub-task modules, and α and β are the hyper-parameters controlling the importance of the two modules contributions respectively.

If the training example is fully labeled with both segmentation and type annotated, training is straightforward; if the training example is partially labeled, e.g., only with segmentation but without type, we can set the log probability of the type module and the decision module 0 and only train the segmentation module. This formulation provides extra flexibility of using partially annotated corpus together with fully annotated corpus to improve the overall performance.

6 Experimental Evaluation

Our experimental evaluation is designed to evaluate the two key aspects of our model:
(Q1) *Can the modular architecture alleviate the difficulty of learning the final task?* To answer

this question, we compare our modular architecture to the traditional neural-CRF model and several recent competitive models for sequence labeling combining inference and deep learning. The results are summarized in Tables 1-3.

(Q2) *Can partial labels be used effectively as a new form of weak-supervision?* To answer this question we compared the performance of the model when trained using disjoint sets of partial and full labels, and show that adding examples only associated with partial labels, can help boost performance on the final task. The results are summarized in Figures 3-5.

6.1 Experimental Settings

6.1.1 Datasets

We evaluated our models over three different sentiment analysis tasks adapted for sequence prediction. We included additional results for multilingual NER in the Appendix for reference.

Target Sentiment Datasets We evaluated our models on the targeted sentiment dataset released by Mitchell et al. (2013), which consists of entity and sentiment annotations on both English and Spanish tweets. Similar to previous studies (Mitchell et al., 2013; Zhang et al., 2015; Li and Lu, 2017), our task focuses on people and organizations (collapsed into *volitional named entities* tags) and the sentiment associated with their description in tweets. After this processing, the labels of each tweets are composed of both segmentation (entity spans) and types (sentiment tags).

We used the original 10-fold cross validation splits to calculate averaged F1 score, using 10% of the training set for development. We used the same metrics in Zhang et al. (2015) and Li and Lu (2017) for a fair comparison.

Aspect Based Sentiment Analysis Datasets We used the Restaurants dataset provided by SemEval 2016 Task 5 subtask 1, consisting of opinion target (aspect) expression segmentation, aspect classification and matching sentiment prediction. In the original task definition, the three tasks were designed as a pipeline, and assumed gold aspect labels when predicting the matching sentiment labels. Instead, our model deals with the challenging end-to-end setting by casting the problem as a sequence labeling task, labeling each aspect segment with the aspect label and sentiment polarity[2].

Subjective Polarity Disambiguation Datasets We adapted the SemEval 2013 Task 2 subtask A as another task to evaluate our model. In this task, the system is given a marked phrase inside a longer text, and is asked to label its polarity. Unlike the original task, we did not assume the sequence is known, resulting in two decisions, identifying subjective expressions (i.e., a segmentation task) and labeling their polarity, which can be modeled jointly as a sequence labeling task.

6.1.2 Input Representation and Model Architecture

Following previous studies (Ma and Hovy, 2016; Liu et al., 2018) showing that the word embedding choice can significantly influence performance, we used the pre-trained GloVe 100 dimension Twitter embeddings only for all tasks in the main text. All the words not contained in these embeddings (OOV, out-of-vocabulary words) are treated as an "unknown" word. Our models were deployed with minimal hyper parameters tuning, and can be briefly summarized as: the character embeddings has dimension 30, the hidden layer dimension of the character level LSTM is 25, and the hidden layer of the word level LSTM has dimension 300. Similar to Liu et al. (2018), we also applied highway networks (Srivastava et al., 2015) from the character level LSTM to the word level LSTM. In our pilot study, we shrank the number of parameters in our modular architectures to around one third such that the total number of parameter is similar as that in the LSTM-CRF model, but we did not observe a significant performance change so we kept them as denoted. The values of α and β in the objective function were always set to 1.0.

6.1.3 Learning

We used BIOES tagging scheme but only during the training and convert them back to BIO2 for evaluation for all tasks[3]. Our model was implemented using *pytorch* (Paszke et al., 2017). To help improve performance we parallelized the for-

[2] using only the subset of the data containing sequence information

[3] Using BIOES improves model complexity in Training, as suggested in previous studies. But to make a fair comparison to most previous work, who used BIO2 for evaluation, we converted labels to BIO2 system in the testing stage. (To be clear, using BIOES in the testing actually yields higher f1 scores in the testing stage, which some previous studies used unfairly)

ward algorithm and the Viterbi algorithm on the GPU. All the experiments were run on NVIDIA GPUs. We used the Stochastic Gradient Descent (SGD) optimization of batch size 10, with a momentum 0.9 to update the model parameters, with the learning rate 0.01, the decay rate 0.05; The learning rate decays over epochs by $\eta/(1+e*\rho)$, where η is the learning rate, e is the epoch number, and ρ is the decay rate. We used gradient clip to force the absolute value of the gradient to be less than 5.0. We used early-stop to prevent over-fitting, with a patience of 30 and at least 120 epochs. In addition to dropout, we used *Adversarial Training* (AT) (Goodfellow et al., 2014), to regularize our model as the parameter numbers increase with modules. AT improves robustness to small worst-case perturbations by computing the gradients of a loss function w.r.t. the input. In this study, α and β in Eq. 1 are both set to 1.0, and we leave other tuning choices for future investigation.

6.2 Q1: *Monolithic vs. Modular Learning*

Our first set of results are designed to compare our modular learning models, utilize partial labels decomposition, with traditional monolithic models, that learn directly over the full labels. In all three tasks, we compare with strong sequence prediction models, including LSTM-CRF (Lample et al., 2016), which is directly equivalent to our baseline model (i.e., final task decision without the modules), and LSTM-CNN-CRF (Ma and Hovy, 2016) and LSTM-CRF-LM (Liu et al., 2018) which use a richer latent representation for scoring the emission potentials.

Target Sentiment task The results are summarized in Tab. 1. We also compared our models with recently published state-of-the-art models on these datasets. To help ensure a fair comparison with Ma et al. which does not use inference, we also included the results of our model without the CRF layer (denoted LSTM-Ti(g)). All of our models beat the state-of-the-art results by a large margin. The source code and experimental setup are available online[4].

Aspect Based Sentiment We evaluated our models on two tasks: The first uses two modules, for identifying the position of the aspect in the text (i.e., chunking) and the aspect category prediction (denoted E+A). The second adds a third module that predicts the sentiment polarity associated with the aspect (denoted E+A+S). I.e., for a given sentence, label its entity span, the aspect category of the entity and the sentiment polarity of the entity at the same time. The results over four languages are summarized in Tab. 2. In all cases, our modular approach outperforms all monolithic approaches.

System	Architecture	Eng.	Spa.
Zhang et al. (2015)	Pipeline	40.06	43.04
	Joint	39.67	43.02
	Collapsed	38.36	40.00
Li and Lu (2017)	SS	40.11	42.75
	+embeddings	43.55	44.13
	+POS tags	42.21	42.89
	+semiMarkov	40.94	42.14
Ma et al. (2018)	HMBi-GRU	42.87	45.61
baseline	LSTM-CRF	49.89	48.84
This work	LSTM-Ti(g)	45.84	46.59
	LSTM-CRF-T	51.34	49.47
	LSTM-CRF-Ti	51.64	49.74
	LSTM-CRF-Ti(g)	**52.15**	**50.50**

Table 1: Comparing our models with the competing models on the target sentiment task. The results are on the full prediction of both segmentation and sentiment.

Subjective Phrase Identification and Classification This dataset contains tweets annotated with sentiment phrases, used for training the models. As in the original SemEval task, it is tested in two settings, in-domain, where the test data also consists of tweets, and out-of-domain, where the test set consists of SMS text messages. We present the results of experiments on these data set in Table 3.

6.3 Q2: *Partial Labels as Weak Supervision*

Our modular architecture is a natural fit for learning with *partial labels*. Since the modular architecture decomposes the final task into sub-tasks, the absence of certain partial labels is permitted. In this case, only the module corresponding to the available partial labels will be updated while the other parts of the model stay fixed.

This property can be exploited to reduce the supervision effort by defining semi-supervised learning protocols that use partial-labels when the full labels are not available, or too costly to annotate. E.g., in the target sentiment task, segmentation labels are significantly easier to annotate.

To demonstrate this property we conducted two sets of experiments. The first investigates how the decision module can effectively *integrate* the knowledge independently learned by sub-tasks

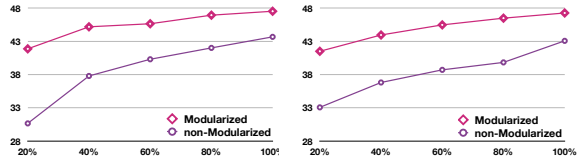
[4] https://github.com/cosmozhang/Modular_Neural_CRF

Models	English		Spanish		Dutch		Russian	
	E+A	E+A+S	E+A	E+A+S	E+A	E+A+S	E+A	E+A+S
LSTM-CNN-CRF(Ma and Hovy, 2016)	58.73	44.20	64.32	50.34	51.62	36.88	58.88	38.13
LSTM-CRF-LM(Liu et al., 2018)	62.27	45.04	63.63	50.15	51.78	34.77	62.18	38.80
LSTM-CRF	59.11	48.67	62.98	52.10	51.35	37.30	63.41	42.47
LSTM-CRF-T	60.87	49.59	64.24	52.33	52.79	37.61	64.72	43.01
LSTM-CRF-TI	63.11	50.19	64.40	52.85	53.05	38.07	64.98	44.03
LSTM-CRF-TI(g)	**64.74**	**51.24**	**66.13**	**53.47**	**53.63**	**38.65**	**65.64**	**45.65**

Table 2: Comparing our models with recent results on the Aspect Sentiment datasets.

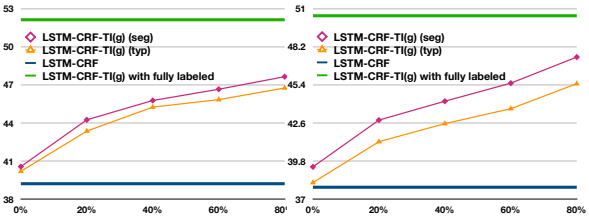
Models	Tweets	SMS
LSTM-CNN-CRF	35.82	23.23
LSTM-CRF-LM	35.67	23.25
LSTM-CRF	34.15	26.28
LSTM-CRF-T	35.37	27.11
LSTM-CRF-Ti	36.52	28.05
LSTM-CRF-Ti(g)	**37.71**	**29.24**

Table 3: Comparing our models with competing models on the subjective sentiment task.



(a) Spanish (b) English

Figure 3: Modular knowledge integration results on the Target Sentiment Datasets. The x-axis is the amount of percentage of the third fold of full labels. The "non-modularized" means we only provide fully labeled data from the third fold.



(a) Spanish (b) English

Figure 4: The fully labeled data was fixed to 20% of the whole training set, and gradually adding data with only segmentation information (Magenta), or with only type information (Orange), and test our model on the full prediction test. The LSTM-CRF model can only use fully labeled data as it does not decompose the task.

modules using different partial labels. We quantify this ability by providing varying amounts of full labels to support the integration process. The second set studies the traditional semi-supervised settings, where we have a handful of full labels, but we have a larger amount of partial labels.

Modular Knowledge Integration The modular architecture allows us to train each model using data obtained separately for each task, and only use a handful of examples annotated for the final task in order to integrate the knowledge learned by each module into a unified decision. We simulated these settings by dividing the training data into three folds. We associated each one of the first two folds with the two sub-task modules. Each one of the these folds only included the partial labels relevant for that sub-task. We then used gradually increasing amounts of the third fold, consisting of the full labels, for training the decision module.

Fig. 3 describes the outcome for target-sentiment, comparing a non-modular model using only the full labels, with the modular approach, which uses the full labels for knowledge integration. Results show that even when very little full data is available results significantly improve. Additional results show the same pattern for subjective phrase identification and classification are included in the Appendix.

Learning with Partially Labeled Data Partially-labeled data can be cheaper and easier to obtain, especially for low-resource languages. In this set of experiments, we model these settings over the target-sentiment task. The results are summarized in Fig. 4. We fixed the amount of full labels to 20% of the training set, and gradually increased the amount of partially labeled data. We studied adding segmentation and type separately. After the model is trained in this routine, it was tested on predicting the full labels jointly on the test set.

Domain Transfer with Partially Labeled Data In our final analysis we considered a novel domain-adaptation settings, where we have a small amount of fully labeled in-domain data from aspect sentiment and more out-of-domain data

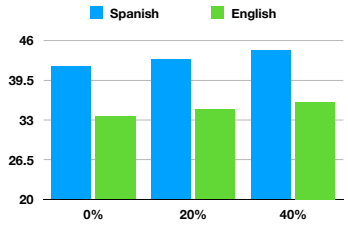


Figure 5: Domain Transfer experiments results with fixed 20% in-domain data from aspect sentiment and varying amounts of out-of-domain data from target sentiment, shown on the x-axis.

from target sentiment. However unlike the traditional domain-adaptation settings, the out-of-domain data is labeled for a different task, and only shares one module with the original task.

In our experiments we fixed 20% of the fully labeled data for the *aspect* sentiment task, and gradually added out-of-domain data, consisting of partial sentiment labels from the *target* sentiment task. Our model successfully utilized the out-of-domain data and improved performance on the in-domain task. The results are shown on Fig 5.

7 Conclusions

We present and study several modular neural architectures designed for a novel learning scenario: learning from partial labels. We experiment with several sentiment analysis tasks. Our models, inspired by cognitive neuroscience findings (Jacobs et al., 1991; Eysenck and Keane, 2005) and multi-task learning, suggest a functional decomposition of the original task into two simpler sub-tasks. We evaluate different methods for sharing information and integrating the modules into the final decision, such that a better model can be learned, while converging faster[5]. As our experiments show, modular learning can be used with weak supervision, using examples annotated with partial labels only.

The modular approach also provides interesting directions for future research, focusing on alleviating the supervision bottleneck by using large amount of partially labeled data that are cheaper and easy to obtain, together with only a handful amount of annotated data, a scenario especially suitable for low-resource languages.

[5]Convergence results are provided in the Appendix

Acknowledgements

We thank the reviewers for their insightful comments. We thank the NVIDIA Corporation for their GPU donation, used in this work. This work was partially funded by a Google Gift.

References

Rodrigo Agerri and German Rigau. 2016. Robust multilingual named entity recognition with shallow semi-supervised features. *Artificial Intelligence*.

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016. Learning to compose neural networks for question answering. In *Proc. of the Annual Meeting of the North American Association of Computational Linguistics (NAACL)*.

Xavier Carreras, Lluís Màrquez, and Lluís Padró. 2002. Named entity extraction using adaboost. In *Proc. of the Annual Conference on Computational Natural Language Learning (CoNLL)*.

Rich Caruana. 1997. Multitask Learning. *Machine Learning*, 28(1):41–75.

Chen, Shi, Qiu, and Huang. 2017. Adversarial multi-criteria learning for chinese word segmentation. In *Proc. of the Annual Meeting of the Association Computational Linguistics (ACL)*.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural Language Processing (Almost) from Scratch. *J. Mach. Learn. Res.*, 12.

Timothee Cour, Ben Sapp, and Ben Taskar. 2011. Learning from partial labels. *Journal of Machine Learning Research*, 12(May).

Eriguchi, Tsuruoka, and Cho. 2017. Learning to parse and translate improves neural machine translation. In *Proc. of the Annual Meeting of the Association Computational Linguistics (ACL)*.

M.W. Eysenck and M.T. Keane. 2005. *Cognitive Psychology: A Student's Handbook*. Psychology Press.

Norman. Geschwind and Albert M. Galaburda. 1987. *Cerebral lateralization : biological mechanisms, associations, and pathology*. MIT Press.

Gillick, Brunk, Vinyals, and Subramanya. 2015. Multilingual Language Processing From Bytes. *ArXiv*.

I. J. Goodfellow, J. Shlens, and C. Szegedy. 2014. Explaining and Harnessing Adversarial Examples. *ArXiv e-prints*.

Ronghang Hu, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Kate Saenko. 2017. Learning to reason: End-to-end module networks for visual question answering. In *Proc. of the International Conference on Computer Vision (ICCV)*.

Jacobs, Jordan, and Barto. 1991. Task decomposition through competition in a modular connectionist architecture: The what and where vision tasks. *Cognitive Science*, 15(2).

Stephen M. Kosslyn. 1987. Seeing and Imagining in the Cerebral Hemispheres: A Computational Approach. *Psychological Review*, 94(2):148–175.

Guillaume Lample, Miguel Ballesteros, Kazuya Kawakami, Sandeep Subramanian, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proc. of the Annual Meeting of the North American Association of Computational Linguistics (NAACL)*.

Hao Li and Wei Lu. 2017. Learning latent sentiment scopes for entity-level sentiment analysis. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*.

Liyuan Liu, Jingbo Shang, Frank F. Xu, Xiang Ren, Huan Gui, Jian Peng, and Jiawei Han. 2018. Empower sequence labeling with task-aware neural language model. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*.

Minh-Thang Luong. 2016. Multi-Task Sequence To Sequence Learning. In *Proc. International Conference on Learning Representation (ICLR)*.

Dehong Ma, Sujian Li, and Houfeng Wang. 2018. Joint learning for targeted sentiment analysis. In *Proc. of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proc. of the Annual Meeting of the Association Computational Linguistics (ACL)*.

Mortimer Mishkin, Leslie G. Ungerleider, and Kathleen A. Macko. 1983. Object vision and spatial vision: two cortical pathways.

Margaret Mitchell, Jacqui Aguilar, Theresa Wilson, and Benjamin Van Durme. 2013. Open domain targeted sentiment. In *Proc. of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*.

Joel Nothman, Nicky Ringland, Will Radford, Tara Murphy, and James R. Curran. 2013. Learning multilingual named entity recognition from wikipedia. *Artif. Intell.*, 194:151–175.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. In *NIPS-W*.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, AL-Smadi Mohammad, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, et al. 2016. Semeval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016)*.

L. Ratinov and D. Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proc. of the Annual Conference on Computational Natural Language Learning (CoNLL)*.

Rueckl, Cave, and Kosslyn. 1989. Why are "What" and "Where" Processed by Separate Cortical Visual Systems? A Computational Investigation. *cognitive neuroscience*.

Tjong Kim Sang and Erik F. 2002. Introduction to the conll-2002 shared task: Language-independent named entity recognition. In *Proc. of the Annual Conference on Computational Natural Language Learning (CoNLL)*.

Tjong Kim Sang, Erik F., and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proc. of the Annual Conference on Computational Natural Language Learning (CoNLL)*.

dos Santos and Guimarães. 2015. Boosting named entity recognition with neural character embeddings. In *Proc. of the Annual Meeting of the Association Computational Linguistics (ACL)*.

Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015. Highway Networks. *ArXiv e-prints*.

Shubham Toshniwal, Hao Tang, Liang Lu, and Karen Livescu. 2017. Multitask learning with low-level auxiliary tasks for encoder-decoder based speech recognition. In *INTERSPEECH*.

Theresa Wilson, Zornitsa Kozareva, Preslav Nakov, Alan Ritter, Sara Rosenthal, and Stoyanov Veselin. 2013. Semeval-2013 task 2: Sentiment analysis in twitter.

Licheng Yu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, Mohit Bansal, and Tamara L Berg. 2018. Mattnet: Modular attention network for referring expression comprehension. *arXiv*.

Meishan Zhang, Yue Zhang, and Duy Tin Vo. 2015. Neural networks for open domain targeted sentiment. In *Proc. of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*.

Xiao Zhang, Yong Jiang, Hao Peng, Kewei Tu, and Dan Goldwasser. 2017. Semi-supervised structured prediction with neural crf autoencoder. In *Proc. of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*.

Jie Zhou and Wei Xu. 2015. End-to-end learning of semantic role labeling using recurrent neural networks. In *Proc. of the Annual Meeting of the Association Computational Linguistics (ACL)*.

A Examples of Task Decomposition

In Figure 6, we show an example of task decomposition for standard NER.

Text	Brush	Wellman	comments	on	beryllium	lawsuits	.
Tag	B-ORG	I-ORG	O	O	O	O	O
Seg	B	I	O	O	O	O	O
Ent	ORG	ORG	O	O	O	O	O

Figure 6: An example of NER decomposition.

In Figure 7, we show another example of task decomposition for target sentiment, in addition to the one in the main text.

Text	KC	Concepcion	Rogue	Magazine	Photos	Continue	to	Get	Praised	by	Fans	on	Twitter
Tag	B-pos	B-pos	B-neu	E-neu	O	O	O	O	O	O	O	O	S-neu
Seg	B	E	B	E	O	O	O	O	O	O	O	O	S
Senti	pos	pos	neu	neu	O	O	O	O	O	O	O	O	neu

Figure 7: An extra example of target sentiment decomposition.

B Full Experimental Results on Target Sentiment

The complete results of our experiments on the target sentiment task are summarized in Tab. 4. Our LSTM-CRF-TI(g) model outperforms all the other competing models in Precision, Recall and the F1 score.

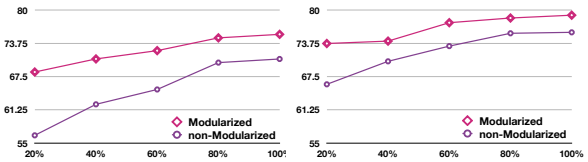
C Experiments on Named Entity Recognition

NER datasets We evaluated our models on three NER datasets, the English, Dutch and Spanish parts of the 2002 and 2003 CoNLL shared tasks (Sang and F., 2002; Sang et al., 2003). We used the original division of training, validation and test sets. The task is defined over four different entity types: *PERSON*, *LOCATION*, *ORGANIZATION*, *MISC*. We used the BIOES tagging scheme during the training, and convert them back to original tagging scheme in testing as previous studies show that using this tagging scheme instead of BIO2 can help improve performance (Ratinov and Roth, 2009; Lample et al., 2016; Ma and Hovy, 2016; Liu et al., 2018). As a result, the segmentation module had 5 output labels, and the entity module had 4. The final decision task, consisted of the Cartesian product of the segmentation set (BIES) and the entity set, plus the "O" tag, resulting in 17 labels.

Results on NER We compared our models with the state-of-the-art systems on English[6], Dutch and Spanish. For Dutch and Spanish, we used cross-lingual embedding as a way to exploit lexical information. The results are shown in Tab. 5 and Tab. 6[7]. Our best-performing model outperform all the competing systems.

D Additional Experiments on Knowledge Integration

We conducted additional experiments on knowledge integration in the same setting as in the main text to investigate the properties of the modules. Figure 8 shows the results for Dutch and Spanish NER datasets, while Figure 9 shows the results for the Subjective Polarity Disambiguation Datasets using the in-domain data.



(a) Dutch NER (b) Spanish NER

Figure 8: Experimental results on modular knowledge integration on the Dutch and Spanish NER datasets.

E Convergence Analysis

The proposed twofold modular infusion model (with guided gating as an option) breaks the complex learning problem into several sub-problems and then integrate them using joint training. The process defined by this formulation has more parameters and requires learning multiple objectives jointly. Our convergence analysis intends to evaluate whether the added complexity leads to a harder learning problem (i.e., slower to converge) or whether the tasks constrain each other and as a result can be efficiently learned.

[6] Liu et al.'s results are different since their implementation did not convert the predicted BIOES tags back to BIO2 during evaluation. For fair comparison, we only report the results of the standard evaluation.

[7] We thank reviewers for pointing out a paper (Agerri and Rigau, 2016) obtains the new state-of-the-art result on Dutch with comparable results on Spanish.

System	Architecture	English			Spanish		
		Pre	Rec	F1	Pre	Rec	F1
Zhang, Zhang and Vo (2015)	Pipeline	43.71	37.12	40.06	45.99	40.57	43.04
	Joint	44.62	35.84	39.67	46.67	39.99	43.02
	Collapsed	46.32	32.84	38.36	47.69	34.53	40.00
Li and Lu (2017)	SS	44.57	36.48	40.11	46.06	39.89	42.75
	+embeddings	47.30	40.36	43.55	47.14	41.48	44.13
	+POS tags	45.96	39.04	42.21	45.92	40.25	42.89
	+semiMarkov	44.49	37.93	40.94	44.12	40.34	42.14
Base Line	LSTM-CRF	53.29	46.90	49.89	51.17	46.71	48.84
This work	LSTM-CRF-T	54.21	48.77	51.34	51.77	47.37	49.47
	LSTM-CRF-Ti	54.58	49.01	51.64	52.14	47.56	49.74
	LSTM-CRF-Ti(g)	**55.31**	**49.36**	**52.15**	**52.82**	**48.41**	**50.50**

Table 4: Performance on the target sentiment task

Model	English
LSTM-CRF (Lample et al., 2016)	90.94
LSTM-CNN-CRF (Ma and Hovy, 2016)	91.21
LM-LSTM-CRF (Liu et al., 2018)	91.06
LSTM-CRF-T	90.8
LSTM-CRF-TI	91.16
LSTM-CRF-TI(g)	**91.68**

Table 5: Comparing our models with several state-of-the-art systems on the CoNLL 2003 English NER dataset.

Model	Dutch	Spanish
Carreras et al. (2002)	77.05	81.39
Nothman et al. (2013)	78.60	N/A
dos Santos and Guimarães (2015)	N/A	82.21
Gillick et al. (2015)	82.84	82.95
Lample et al. (2016)	81.74	85.75
LSTM-CRF-T	83.91	84.89
LSTM-CRF-TI	84.12	85.28
LSTM-CRF-TI(g)	**84.51**	**85.92**

Table 6: Comparing our models with recent results on the 2002 CoNLL Dutch and Spanish NER datasets.

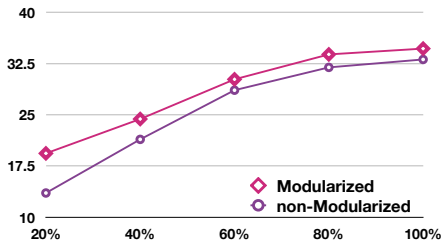


Figure 9: Experimental results on modular knowledge integration on the Subjective Polarity Disambiguation Datasets.

We compare between our LSTM-CRF-TI(g) model and recent published top models on the English NER dataset in Figure 10 and on the subjective polarity disambiguation datasets in Figure 11. The curve compares convergence speed in terms of learning epochs. Our LSTM-CRF-TI(g) model has a much faster convergence rate compared to the other models.

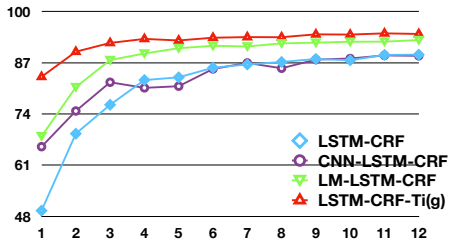


Figure 10: Comparing convergence over the development set on the English NER dataset. The x-axis is number of epochs and the y-axis is the F1-score.

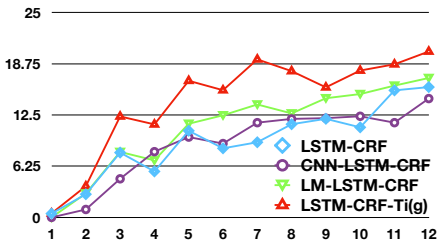


Figure 11: Comparing convergence over the development set on the subjective polarity disambiguation datasets. The x-axis is number of epochs and the y-axis is the F1-score.