Weak Supervision for Learning Discourse Structure

Sonia Badene[1,2], Kate Thompson[1,2], Jean-Pierre Lorré[2], Nicholas Asher[1,3,4]

[1]IRIT, [2]Linagora, [3]CNRS, [4]ANITI

{*sonia.badene,kate.thompson,nicholas.asher*}@*irit.fr*, {*sbadene,jplorre*}@*linagora.com*

Abstract

This paper provides a detailed comparison of a data programming approach with (i) off-the-shelf, state-of-the-art deep learning architectures that optimize their representations (BERT) and (ii) handcrafted-feature approaches previously used in the discourse analysis literature. We compare these approaches on the task of learning discourse structure for multi-party dialogue. The data programming paradigm offered by the Snorkel framework allows a user to label training data using expert-composed heuristics, which are then transformed via the "generative step" into probability distributions of the class labels given the data. We show that on our task the generative model outperforms both deep learning architectures as well as more traditional ML approaches when learning discourse structure—it even outperforms the combination of deep learning methods and hand-crafted features. We also implement several strategies for "decoding" our generative model output in order to improve our results. We conclude that weak supervision methods hold great promise as a means for creating and improving data sets for discourse structure.

1 Introduction

In this paper, we investigate and demonstrate the potential of a weak supervision, data programming approach (Ratner et al., 2016) to the task of learning discourse structure for multi-party dialogue. We offer a detailed comparison of our data programming approach with (i) off-the-shelf, state-of-the-art deep learning architectures that optimize their representations (BERT) and (ii) handcrafted-feature approaches previously used in the discourse analysis literature. Our data programming paradigm exploits the Snorkel framework that allows a user to label training data using expert-composed heuristics, which are then transformed via the "generative step" into probability distributions of the class labels given the data. We show that the generative model produced from these heuristics outperforms both deep learning architectures as well as more traditional ML approaches when learning discourse structure by up to 20 points of F1 score; it even outperforms the combination of generative and discriminative approaches that are the foundation of the Snorkel framework. We also implement several strategies for "decoding" our generative model output that improve our results.

We assume discourse structures are dependency structures (Muller et al., 2012; Li et al., 2014) and restrict the structure learning problem to predicting edges or attachments between discourse unit (DU) pairs in the dependency graph. Although the problem of attachment is only a part of the overall task of discourse interpretation, it is a difficult problem that serves as a useful benchmark for various approaches to discourse parsing. After training a supervised deep learning algorithm to predict attachments on the STAC annotated corpus[1], we then constructed a weakly supervised learning system in which we used 10% of the corpus as a development set. Experts on discourse structure wrote a set of attachment rules, or labeling functions (LFs), and tested them against this development set. We treated the remainder of the corpus as raw/unannotated data to be automatically annotated using the data programming framework.

2 State of the Art

Discourse structures for texts represent causal, topical, argumentative information through what are called coherence relations. For dialogues with multiple interlocutors, extraction of their discourse structures provides useful semantic infor-

[1]https://www.irit.fr/STAC/

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 2296–2305, Hong Kong, China, November 3–7, 2019. ©2019 Association for Computational Linguistics

mation to the "downstream" models used, for example, in the production of intelligent meeting managers or the analysis of user interactions in online fora. However, despite considerable efforts to retrieve discourse structures automatically (Fisher and Roark, 2007; Duverle and Prendinger, 2009; Li et al., 2014; Joty et al., 2013; Ji and Eisenstein, 2014; Yoshida et al., 2014; Li et al., 2014; Surdeanu et al., 2015), we are still a long way from usable discourse models, especially for dialogue. Standard supervised models struggle to capture the sparse attachments, even when relatively large annotated corpora are available. In addition, the annotation process is time consuming and often fraught with errors and disagreements, even among expert annotators. This motivated us to explore the data programming approach that exploits expert linguistic knowledge in a more compact and consistent rule based form.

Given our interest in the analysis of multi-party dialogues, we used the STAC corpus of multi-party chats, an initial version of which is described in (Afantenos et al., 2015; Perret et al., 2016). In all versions of this corpus, dialogue structures are directed acyclical graphs (DAGs) formed according to SDRT[2] (Asher and Lascarides, 2003; Asher et al., 2016). An SDRT discourse structure is a graph, $\langle V, E_1, E_2, \ell, \text{Last}\rangle$, where: V is a set of nodes or Discourse Units (DUs); $E_1 \subseteq V^2$ is a set of edges between DUs representing coherence relations; $E_2 \subseteq V^2$ represents a dependency relation between DUs; $\ell\colon E_1 \rightarrow R$ is a labeling function that assigns a semantic type to an edge in E_1 from a set R of discourse relation types, and *Last* is a designated element of V giving the last DU relative to textual or temporal order. E_2 is used to represent Complex Discourse Units (CDUs), which are clusters of two or more DUs connected as an ensemble to other DUs in the graph. As learning this type of recursive structure presents difficulties beyond the scope of this paper, we followed a "flattening" strategy similar to (Muller et al., 2012) to remove CDUs. This process yields a set $V*$, which is V without CDUs, and a set $E*_1$, a flattened version of E_1.

Building these structures typically requires three steps: (i) segmenting the text into the basic units of the discourse, typically clauses - these are EDUs or *Elementary Discourse Units*; these, together with CDUs, form the set of nodes V in the graph; (ii) predicting the attachments between DUs, i.e. to identify the elements in E_1; (iii) predicting the semantic type of the edge in E_1. This paper focuses on step (ii). Our dialogue structures are thus of the form $\langle V*, E*_1, \text{Last}\rangle$. Step (ii) is a difficult problem for automatic processing because attachments are theoretically possible between any two DUs in a dialogue or text, and often graphs include long-distance relations. Muller et al. (2012) is the first paper we know of that focuses on the discourse parsing attachment problem, albeit for monologue. It targeted a restricted version of an SDRT graph and trains a simple MaxEnt algorithm to produce probability distributions over pairs of EDUs, what we call a "local model" with a positive F1 attachment score of 0.635. They further applied global decoding constraints to produce a slight improvement in attachment scores (these are discussed in more detail in Section 5). Afantenos et al. (2015) used a similar strategy for dialogue on an early version of the STAC corpus. Perret et al. (2016) targeted a more elaborate approximation of SDRT graphs on the same version of the STAC corpus and reported a local model F1 attachment of 0.483. They then used Integer Linear Programming (ILP) to encode global decoding constraints particular to SDRT to improve the F1 attachment score to 0.689.

Having sketched recent progress in discourse parsing, we briefly turn to the state of the art concerning data programming. Ratner et al. (2016) introduced the data programming paradigm, along with a framework, Snorkel (Ratner et al., 2017), which uses a weak supervision method (Zhou, 2017), to apply labels to large data sets by way of heuristic labeling functions that can access distant, disparate knowledge sources. These labels are then used to train classic data-hungry machine learning (ML) algorithms. The crucial step in the data programming process uses a generative model to unify the noisy labels by generating a probability distribution for all labels for each data point. This set of probabilities replaces the ground-truth labels in a standard discriminative model outfitted with a noise-aware loss function and trained on a sufficiently large data set.

3 The STAC Annotated Corpus

3.1 Overview

While earlier versions only included linguistic moves by players, STAC now contains in addi-

[2]Segmented Discourse Representation Theory

tion a multimodal corpus of multi-party chats between players of an online game (Asher et al., 2016; Hunter et al., 2018). It includes 2,593 dialogues (each with a weakly connected DAG discourse structure), 12,588 "linguistic" DUs, 31,811 "non-linguistic" DUs and 31,251 semantic relations. A dialogue begins at the beginning of a player's turn, and ends at the end of that player's turn. In the interim, players can bargain with each other or make spontaneous conversation. These player utterances are the "linguistic" turns. In addition the corpus contains information given visually in the game interface but transcribed in the corpus into Server or interface messages, "non-linguistic" turns (Hunter et al., 2018). All turns are segmented into DUs, and these units are then connected by semantic relations.

Each dialogue represents a complete conversation. There are typically many such conversations, each beginning with a non-linguistic turn in which a player is designated to begin negotiations (see Figure 1). The dialogues end when this player performs a non-linguistic action that signals the end of their turn. The dialogues are the units on which we build a complete discourse structure.

The STAC multimodal corpus is divided into a development, train and test set. The development and test sets are each 10% of the total size of the corpus.

To compare our approach to earlier efforts, we also used the corpus from (Perret et al., 2016). This corpus was also useful to check for over fitting of our Generative model developed on the multi-modal data. The corpus from (Perret et al., 2016) is an early version of a "linguistic only" version of the STAC corpus. It contains no non-linguistic DUs, unlike the STAC multimodal corpus.[3] It also contains quite a few errors; for example, about 60 stories in the (Perret et al., 2016) dataset have no discourse structure in them at all and consist of only one DU. We eliminated these from the Perret 2016 data set that we used in our comparative experiments below, as these stories were obviously not a correct representation of what was going on in the game at the relevant point.

3.2 Data Preparation

To concentrate on the attachment task, we implemented the following simplifying measures on the STAC corpus:

1. Roughly 56% of the dialogues in the corpus contain only non-linguistic DUs. The discourse structure of these dialogues is more regular and thus less challenging; so we ignore these dialogues for our prediction task.

2. 98% of the discourse relations in our development corpus span 10 DUs or less. To reduce class imbalance, we restricted the relations we consider to a distance of ≤ 10.

3. Following (Muller et al., 2012; Perret et al., 2016) we "flatten" CDUs by connecting all relations incoming or outgoing from a CDU to the "head" of the CDU, or its first DU.

The STAC corpus as we use it in our learning experiments thus includes 1,130 dialogues, 13,734 linguistic DUs, 18,767 non-linguistic DUs and 22,098 semantic relations.

We also performed these operations on our version of the linguistic only corpus used by (Perret et al., 2016).

4 Data Programming Experiments

4.1 Candidates and Labeling Functions

The Snorkel implementation of the data programming paradigm inspired our weak supervision approach. We first identified and extracted candidates from the data, and then wrote a set of labeling functions (LFs) to apply to the candidates while only consulting the development set, the 10% of the STAC corpus set aside to develop and test our LFs. We treated the training set (80% of the corpus) as unseen/unlabeled data to which we applied the finished LFs and the models. The last 10% of the STAC corpus was reserved as a final test set.

Candidates are the units of data for which labels are predicted. For this study, the candidates are all DU pairs which could possibly be connected by a semantic relation. We use our own method to create candidates from the DUs culled from the texts of the dialogues, making sure to limit the pairs to

[3] There is also on the STAC website an updated linguistic only version of the STAC corpus. It has 1,091 dialogues, 11,961 linguistic only DUs and 10,191 semantic relations. We have not reported results on that data set here. The dataset from (Perret et al., 2016) is similar to our linguistic only STAC corpus but is still substantially different and degraded in quality. Asher et al. (2016) report significant error rates in annotation on the earlier versions of the STAC corpus and that the current linguistic only corpus of STAC offers an improvement over the (Perret et al., 2016) corpus.

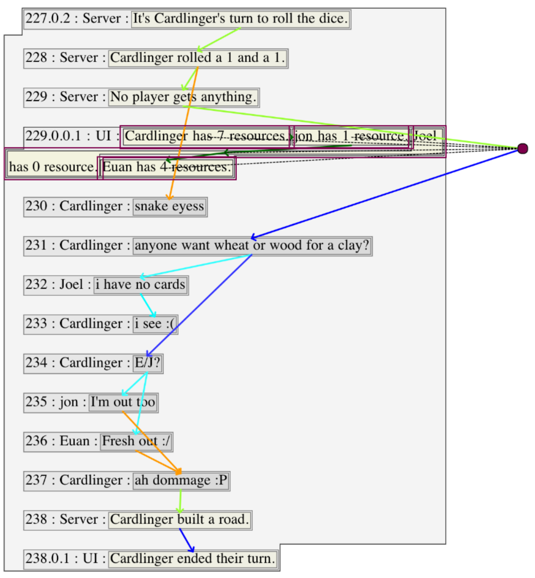
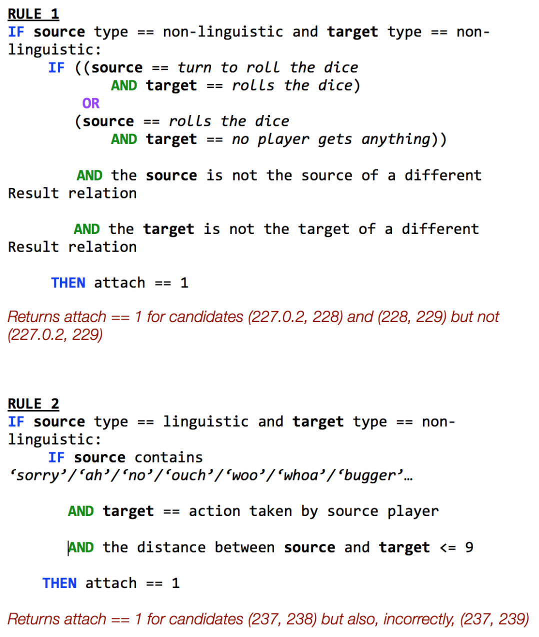

227.0.2 : Server : It's Cardlinger's turn to roll the dice.
228 : Server : Cardlinger rolled a 1 and a 1.
229 : Server : No player gets anything.
229.0.0.1 : UI : Cardlinger has 7 resources. jon has 1 resource. Joel has 0 resource. Euan has 4 resources.
230 : Cardlinger : snake eyess
231 : Cardlinger : anyone want wheat or wood for a clay?
232 : Joel : i have no cards
233 : Cardlinger : i see :(
234 : Cardlinger : E/J?
235 : jon : I'm out too
236 : Euan : Fresh out :/
237 : Cardlinger : ah dommage :P
238 : Server : Cardlinger built a road.
238.0.1 : UI : Cardlinger ended their turn.

Figure 1: A dialogue from the STAC multi-modal corpus featuring player ('linguistic') turns and Server/UI ('non-linguistic') turns. There are four instances of *Result* relations, shown in green. *Result* connects a cause to its effect, i.e., the main eventuality of the first argument is understood to cause the eventuality given by the second.



```
RULE 1
IF source type == non-linguistic and target type == non-
linguistic:
    IF ((source == turn to roll the dice
         AND target == rolls the dice)
       OR
       (source == rolls the dice
         AND target == no player gets anything))

        AND the source is not the source of a different
Result relation

        AND the target is not the target of a different
Result relation

     THEN attach == 1

Returns attach == 1 for candidates (227.0.2, 228) and (228, 229) but not
(227.0.2, 229)


RULE 2
IF source type == linguistic and target type == non-
linguistic:
    IF source contains
'sorry'/'ah'/'no'/'ouch'/'woo'/'whoa'/'bugger'…

        AND target == action taken by source player

        AND the distance between source and target <= 9

    THEN attach == 1

Returns attach == 1 for candidates (237, 238) but also, incorrectly, (237, 239)
```

Figure 2: Glosses of two rules that are meant to capture the *Result* relations when applied to the candidates as pulled from the corpus, exemplified in Figure 1. Although the rules are written to capture specific relation types between segments, they return 1/0 for attached/not attached

those that occur in the same dialogues. We also ruled out the possibility of backwards relations between two DUs which have different speakers: it is linguistically impossible for a speaker of, say, an assertion d_1 at time t_1 to answer a question d_2, asked by a different speaker at time $t_2 > t_1$, i.e. before d_2 was asked. That is, we include (d_1, d_2) in our candidates but rule out (d_2, d_1).

LFs are expert-composed functions that make an attachment prediction for a given candidate: each LF returns a 1, a 0 or a -1 ("attached"/"do not know"/"not attached") for each candidate. However, each of our LFs is written and evaluated with a specific relation type *Result*, *Question-answer-pair (QAP)*, *Continuation*, *Sequence*, *Acknowledgement*, *Conditional*, *Contrast*, *Elaboration* and *Comment* in mind. In this way, LFs leverage a kind of type-related information, which makes sense from an empirical perspective as well as an epistemological one. An attachment decision concerning two DUs is tightly linked to the type of relation relating the DUs: when an annotator decides that two DUs are attached, he or she does so with some knowledge of what type of relation attaches them. Figure 2 shows a sample LFs used for attachment prediction with the *Result* relation in mind.

LFs also exploit information about the DUs' linguistic or non-linguistic status, the dialogue acts they express, their lexical content, grammatical category and speaker, and the distance between them—features also used in supervised learning methods (Perret et al., 2016; Afantenos et al., 2015; Muller et al., 2012). Finally, we fix the order in which each LF "sees" the candidates such that it considers adjacent DUs before distant DUs. This allows LFs to exploit information about previously predicted attachments and dialogue history in new predictions. Our rule set and their description are available here: https://tizirinagh.github.io/acl2019/. Figure 2 gives an example of a labeling function that we used.

4.2 The Generative Model

Once the LFs are applied to all the candidates, we have a matrix of labels $(\bar{\Lambda})$ given by each LF Λ for each candidate. The generative model, GEN, as specified in (1), provides a general distribution of marginal probabilities relative to n accuracy dependencies $\phi_j(\Lambda_i, y_i)$ for an LF λ_j with respect inputs x_i, the LF's outputs on i Λ_{ij} and true labels

y_i that depend on parameters θ_j where:

$$\phi_j(\Lambda_i, y_i) := y_i \Lambda_{ij}$$

$$p_\theta(\Lambda, Y) \propto exp(\sum_{i=1}^{m}\sum_{j=1}^{n} \theta_j \phi_j(\Lambda_i, y_i)) \quad (1)$$

The parameters are estimated by minimizing the negative log marginal likelihood of the output of an observed matrix $\overline{\Lambda}$ as in (2).

$$argmin_\theta - log \sum_Y p_\theta(\overline{\Lambda}, Y) \quad (2)$$

GEN does not have access to the gold labels on the Training set but uses the Training set as unlabelled data. So in this model, the true class labels y_i are latent variables that generate the labeling function outputs, which are estimated via Gibbs sampling over the Training set (80% of the STAC corpus), after it has been labeled by the LFs. The objective in (2) is then optimized by interleaving stochastic gradient descent steps with Gibbs sampling ones. For each candidate, GEN thus uses the accuracy measures for the LFs in (1) to assign marginal probabilities that two DUs are attached.

GEN estimates the accuracy of each LF, a marginal probability for each label, and consequently a probability for positive attachment. In this model, the true class labels y_i are latent variables that generate the labeling function outputs. The model in (1) presupposes that the LFs are independent, but this assumption does not always hold: one LF might be a variation of another or they might depend on a common source of information (Mintz et al., 2009). If we don't take these dependencies into account, we risk assigning incorrect accuracies to the LFs. Snorkel provides a more complex model that automatically calculates the dependencies between LFs and marginal probabilities which we use for the generative step (Bach et al., 2017). The higher order dependencies significantly improved the generative model's results on the full STAC corpus (see Table 1).

When we obtain the results from the generative model GEN, we choose on the development corpus a threshold to apply to these marginals by calculating the threshold that gives us the best F1 score. The best threshold is 0.85 ($p > .85$ for positive attachment) in the STAC corpus. Figure 3 shows the probability distribution, on which even taking 0.8 as a threshold gives a lower F1 score because of false positive attachment. Binarizing these marginals allows us to pass these binarized probabilities to the discriminative model. This also allows us to evaluate GEN with respect to gold label "attachment" 0/"non attachment" 1 on the STAC test data.

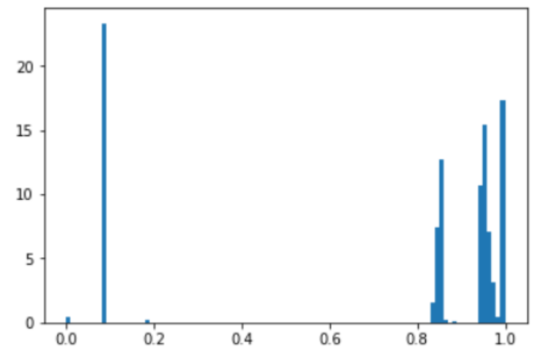


Figure 3: Probability distribution for positive attachment in the STAC development set.

The generative model GEN shares with other "local" models the feature that it considers pairs of DUs in isolation of the whole structure. However, unlike other local models, our LFs enable GEN to exploit prior decisions on pairs of DUs, and thus we exploit more contextual information about discourse structure in GEN than in our classical, supervised local models. In addition, GEN uses the Training set very differently from classical, supervised models.

4.3 Discriminative Model

The standard Snorkel approach inputs the marginal probabilities from the generative step directly into a discriminative model, which is trained on those probabilities using a *noise-aware loss function* (Ratner et al., 2016). Ideally, this step generalizes the LFs by augmenting the feature representation - from, say, dozens of LFs to a high dimensional feature space - and allows the model to predict labels for more new data. Thus the precision potentially lost in the generalization is offset by a larger increase in recall.

We tested three discriminative models in our study. Each one was trained on the gold labeled data in the Training set of the STAC corpus. First we tried a single layer BI-LSTM with 300 neurons, which takes as input 100 dimensional-embeddings for the text of each DU in the candidate pair. We concatenated the outputs of the BI-LSTM and fed them to a simple perceptron with one hidden layer and Rectified Linear Unit

Generative Model on dev set	Precision	Recall	F1 score	Accuracy
Without higher order dependencies	0.45	0.70	0.55	0.87
With higher order dependencies	0.68	0.67	0.67	0.92

Table 1: Evaluations of attachment on Dev set with and without higher order dependencies.

(ReLU) activation (Hahnloser et al., 2000; Jarrett et al., 2009; Nair and Hinton, 2010) and optimized with Adam (Kingma and Ba, 2014). Given that our data is extremely unbalanced in favor of the "unattached" class ("attached" candidates are roughly 13% of the candidates on the development set), we also implemented a class-balancing method inspired by (King and Zeng, 2001) which maps class indices to weight values used for weighting the loss function during training.

We also implemented BERT (Devlin et al., 2018)'s sequence classification model (source code on the link below[4]) with 10 training epochs and all default parameters otherwise. BERT, the Bidirectional Encoder Representations from Transformers, is a text encoder pre-trained using language models where the system has to guess a missing word or word piece removed at random from the text. Originally designed for automatic translation tasks, BERT uses bi-directional self-attention to produce the encodings and performs at the state of the art on many textual classification tasks.

In order to use these methods, we had to binarize the marginal probabilities before moving to the discriminative step, using a threshold of $p > .85$ as explained in Section 4.2. Though this marks a departure from the standard Snorkel approach, we found that our discriminative model results were higher when the marginals were binarized and when the class re-balancing was used, albeit much lower than expected overall.

Finally, to facilitate comparison with earlier work, we also implemented a local model, LogReg* as mentioned in the following of the paper, that used marginal probabilities together with handcrafted features (the feature set used in (Afantenos et al., 2015) listed in their Table 2) and a Logistic Regression classifier.

[4]Link to BERT sequence classification model code: https://github.com/huggingface/pytorch-pretrained-BERT/blob/master/examples/run_classifier.py

5 Decoding

A set of highly accurate predictions for individual candidates does not necessarily lead to accurate discourse structures; for instance, without global structural constraints, GEN and local models may not yield the directed acyclic graphs (DAGs), required by SDRT. As in previous work (Muller et al., 2012; Afantenos et al., 2015; Perret et al., 2016), we use the Maximum Spanning Tree (MST) algorithm, and a variation thereof, to ensure that the dialogue structures predicted conform to some more general structural principle. We implemented the Chu-Liu-Edmonds algorithm (Chu, 1965; Edmonds, 1967), an efficient method of finding the highest-scoring non-projective tree in a directed graph, as described in Jurafsky and Martin[5]. The algorithm greedily selects the relations with the highest probabilities from the dependency graphs produced by the local model, then removes any cycles. The result is a tree structure with one incoming relation per node. In cases of nodes with multiple equiprobable incoming relations, the algorithm takes whichever relation it sees first.

Since SDRT structures can contain nodes with multiple incoming relations, i.e. are not always tree-like, we altered the MST algorithm in the manner of (Muller et al., 2012; Afantenos et al., 2015; Perret et al., 2016), forcing the MST to include all high-probability incoming relations which do not create cycles. This produces MS-DAG structures which are in principle more faithful to SDRT. In addition, since discourse attachments in general follow an inverse power law (many short-distance attachments and fewer long-distance attachments), we implemented two MST/MS-DAG variants that always choose the shortest relation among multiple high-probability relations (MST/short and MS-DAG/short).

6 Results and Analysis

We set out to test the performance of combinations of generative and discriminative models along the

[5]https://web.stanford.edu/ jurafsky/slp3/13.pdf

lines of the data programming paradigm on the task of dialogue structure prediction in order to automatically generate SDRT corpus data. While our results were consistent with what data programming promises—more data with accuracy comparable if not slightly below that of hand-labeled data—our most surprising and interesting result was the performance of the generative model on its own. As seen in Table 2 on STAC test data, GEN dramatically outperformed our deep learning baselines—BiLSTM, BERT, and BERT + LogReg* architectures on gold labels—as well as the LAST baseline, which attaches every DU in a dialogue to the DU directly preceding it. In addition, stand alone GEN also outperformed all the coupled Snorkel models, in which GEN is combined with an added discriminative step, by up to a 30 point improvement in F1 score (GEN vs. GEN+BiLSTM). We did not expect this, given that adding a discriminative model in Snorkel is meant to generalize, and hence improve, what GEN learns.

Critical to this success was the inclusion of higher order dependencies in GEN and the fact that our LFs exploited contextual information about the DUs that was unavailable to the deep learning models or even the handcrafted feature model. GEN beats all competitors in terms of F1 score while taking a fraction of the annotated data to develop and train the model, showing the power and promise of the generative model.

One might wonder whether GEN and discriminative models are directly comparable. Generative machine learning algorithms learn the joint probability of X and Y, whereas discriminative algorithms learn the conditional probability of Y given X. Nevertheless, when we exploit the generative model we are trying to find the Y for which $P(X \wedge Y)$ is maximized. In effect we are producing, *though not learning*, a conditional probability. So it makes sense to compare our generative model's output with that of other, discriminative machine learning approaches.

We also got surprising results concerning the supervised model benchmarks. Table 2 shows that LogReg* was the best supervised learning method on the STAC data in terms of producing local models. This is evidence that hand crafted features capturing non local information about a DU's contexts do better than all purpose contextual encodings from neural nets at least on this task. We also implemented BERT+LogReg*, a learning algorithm that uses BERT's encodings together with a Logistic Regression classifier trained on STAC's gold data with handcrafted features from (Afantenos et al., 2015) and used in (Perret et al., 2016). BERT+LogReg* outputs a local model that improves upon BERT's local model, but it did not do as well as LogReg* on its own (let alone GEN), suggesting that BERT's encodings actually interfered with the correct predictions.

We also investigated GEN coupled with various discriminative models to test the standard Snorkel. As we remarked above, we found that binarizing GEN's output improved the performance of the coupled discriminative model, so Table 2 only reports scores for various GEN-coupled discriminative models that take the binarized GEN predictions as input. In keeping with our analysis of the supervised benchmarks, we found that the best discriminative model to couple with GEN in the Snorkel architecture was LogReg*, far outperforming GEN with either a BiLSTM or BERT on the STAC test set. Its results were only slightly less good than those of stand alone GEN.

To further investigate comparisons between different architectures for solving the attachment problem, we compared various local models extended with the MS-DAG decoding algorithm discussed in Section 5, giving the global results shown in the righthand columns of Tables 3 and 4. With MS-DAG added, GEN continued to outperform all other approaches on STAC data. In Table 5, we experimented with adding all decoding algorithms to the local GEN result. This gave a boost in F1 score—2 points with classic MST and MS-DAG and 4 points with the variants favoring relation instances with shorter attachments.

It is not surprising that MST improves the GEN results, since it eliminates some of the false positive relations that pass the generative threshold and includes some of the false negative relations that fall below the threshold. The general inverse power law distribution of discourse attachments explains the good performance of the MST shortest link variant. GEN + "MST-short" has the highest attachment score of all approaches to the problem of attachment in the literature (Morey et al., 2018), though we are cautious in comparing scores for systems applied to different corpora.

Finally, we wanted to see how GEN and our other models fared on our version of the (Perret

	Precision	Recall	F1 score	Accuracy
SUPERVISED BASELINES				
LAST	0.54	0.55	0.55	0.84
BiLSTM on Gold labels	0.33	0.80	0.47	0.75
BERT on Gold labels	0.56	0.48	0.52	0.88
LogReg* on Gold labels	0.73	0.52	0.61	0.91
BERT+LogReg* on Gold labels	0.59	0.49	0.53	0.89
SNORKEL PIPELINE				
GEN + Disc (BiLSTM)	0.28	0.59	0.38	0.74
GEN + Disc (BERT)	0.49	0.40	0.44	0.86
GEN + Disc (LogReg*)	0.68	0.65	0.67	0.91
GENERATIVE STAND ALONE				
GEN	**0.69**	**0.66**	**0.68**	**0.92**
GEN + MST-short	**0.73**	**0.71**	**0.72**	**0.93**

Table 2: Evaluations of weakly supervised (Snorkel and stand alone GEN) and supervised approaches on STAC data.

	Local Model			**Global Model**		
	Precision	Recall	F1	Precision	Recall	F1
GEN	**0.69**	**0.66**	**0.68**	**0.71**	**0.69**	**0.70**
LogReg* on STAC	0.73	0.52	0.61	0.65	0.64	0.65

Table 3: Comparison of local and global models on STAC data.

et al., 2016) data set. Comparing GEN to LogReg* on our version of the (Perret et al., 2016) data set, GEN has higher scores than LogReg*'s local model; but with a decoding mechanism similar to that reported in (Perret et al., 2016), LogReg*'s global model significantly improves over the GEN's. We see a 6 point loss in F1 score on GEN's global model relative to LogReg*'s, even though both used identical MST decoding mechanisms. This is what one would expect from a Snorkel based architecture, although it's not the rule that we observed for GEN. The only reason GEN did not beat LogReg* is that it did not get a sufficient boost from decoding. We think that this happened because our LFs already contain a lot of global information about the discourse structure, which meant that MST had less of an effect.

Note, however, that even on the (Perret et al., 2016) data set, the MST decoding mechanism provided LogReg* only a boost of 12 F1 points, as seen in Table 4, which is significantly lower than what is reported in (Perret et al., 2016). This 12% boost is the upper limit for boosts with MST that we were able to reproduce. This could be a result of our eliminating the degraded one EDU stories from the data set.

7 Conclusions and Future Work

We have compared a weak supervision approach, inspired by Snorkel, with a standard supervised model on the difficult task of discourse attachment. The results of the model from Snorkel's generative step surpass those of a standard supervised learning approach, proving it more than competitive with standard approaches. It can also generate a lot of annotated data in a very short time relative to what is needed for a traditional approach: Asher et al. (2016) state that the STAC corpus took at least 4 years to build; we created and refined our labeling functions in two months. In addition, a big advantage of the generative model from the learning point of view is that we don't have class balancing or drowning problems, which plague problems like discourse attachment. This is because the generative model's predictions are generated in a very different way from those of a discriminative model, which are based on inductive generalizations. Still it is clear that we must further investigate the interaction of the generative and discriminative models in order to eventually leverage the power of generalization that a discriminative model is supposed to afford.

In future work, we will enrich our weak supervi-

	Local Model			**Global Model**		
	Precision	Recall	F1	Precision	Recall	F1
Perret et al. 2016	0.66	0.38	0.48	**0.68**	**0.65**	**0.67**
LogReg*	0.64	0.39	0.48	0.59	0.61	0.60
BERT	0.43	0.31	0.36	0.46	0.47	0.46
GEN	**0.46**	**0.65**	**0.54**	0.53	0.54	0.54
GEN + LogReg*	0.46	0.65	0.54	0.53	0.54	0.53

Table 4: Comparison of Perret 2016 and results of our methods on Perret 2016 data.

	F1 scores of MST variants using GEN inputs			
	MS-DAG	MST	MST/short	MS-DAG/short
F1 Score:	0.70	0.70	0.72	0.72

Table 5: F1 scores for different decoding mechanisms with GEN on STAC data.

sion system by giving the LFs access to more sophisticated contexts that take into account global structuring constraints in order to see how they compare to the simple, exogenous decoding constraints like MST. We think we can put much more sophisticated decoding constraints that don't just work off local probabilities of arcs but on the full information about the arc in its discourse context. This will lead us to understand how weakly supervised methods can effectively capture the global structural constraints on discourse structures directly without decoding or elaborate learning architectures.

References

Stergos Afantenos, Eric Kow, Nicholas Asher, and Jérémy Perret. 2015. Discourse parsing for multiparty chat dialogues. In *Association for Computational Linguistics (ACL)*.

Nicholas Asher, Julie Hunter, Mathieu Morey, Farah Benamara, and Stergos D Afantenos. 2016. Discourse structure and dialogue acts in multiparty dialogue: the stac corpus. In *LREC*.

Nicholas Asher and Alex Lascarides. 2003. *Logics of conversation*. Cambridge University Press.

Stephen H Bach, Bryan He, Alexander Ratner, and Christopher Ré. 2017. Learning the structure of generative models without labeled data. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 273–282. JMLR. org.

Yoeng-Jin Chu. 1965. On the shortest arborescence of a directed graph. *Scientia Sinica*, 14:1396–1400.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

David A Duverle and Helmut Prendinger. 2009. A novel discourse parser based on support vector machine classification. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 665–673. Association for Computational Linguistics.

Jack Edmonds. 1967. Optimum branchings. *Journal of Research of the national Bureau of Standards B*, 71(4):233–240.

Seeger Fisher and Brian Roark. 2007. The utility of parse-derived features for automatic discourse segmentation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 488–495.

Richard HR Hahnloser, Rahul Sarpeshkar, Misha A Mahowald, Rodney J Douglas, and H Sebastian Seung. 2000. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature*, 405(6789):947.

Julie Hunter, Nicholas Asher, and Alex Lascarides. 2018. A formal semantics for situated conversation. *Semantics and Pragmatics*, 11. DOI: http://dx.doi.org/10.3765/sp.11.10.

Kevin Jarrett, Koray Kavukcuoglu, Yann LeCun, et al. 2009. What is the best multi-stage architecture for object recognition? In *2009 IEEE 12th International Conference on Computer Vision (ICCV)*, pages 2146–2153. IEEE.

Yangfeng Ji and Jacob Eisenstein. 2014. Representation learning for text-level discourse parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 13–24.

Shafiq Joty, Giuseppe Carenini, Raymond Ng, and Yashar Mehdad. 2013. Combining intra-and multi-sentential rhetorical parsing for document-level discourse analysis. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 486–496.

Gary King and Langche Zeng. 2001. Logistic regression in rare events data. *Political analysis*, 9(2):137–163.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Sujian Li, Liang Wang, Ziqiang Cao, and Wenjie Li. 2014. Text-level discourse dependency parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 25–35. Association for Computational Linguistics.

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

Mathieu Morey, Philippe Muller, and Nicholas Asher. 2018. A dependency perspective on rst discourse parsing and evaluation. *Computational Linguistics*, pages 198–235.

Philippe Muller, Stergos Afantenos, Pascal Denis, and Nicholas Asher. 2012. Constrained decoding for text-level discourse parsing. *Proceedings of COLING 2012*, pages 1883–1900.

Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814.

Jérémy Perret, Stergos Afantenos, Nicholas Asher, and Mathieu Morey. 2016. Integer linear programming for discourse parsing. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 99–109.

Alexander Ratner, Stephen H Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. 2017. Snorkel: Rapid training data creation with weak supervision. *Proceedings of the VLDB Endowment*, 11(3):269–282.

Alexander J Ratner, Christopher M De Sa, Sen Wu, Daniel Selsam, and Christopher Ré. 2016. Data programming: Creating large training sets, quickly. In *Advances in neural information processing systems*, pages 3567–3575.

Mihai Surdeanu, Tom Hicks, and Marco Antonio Valenzuela-Escarcega. 2015. Two practical rhetorical structure theory parsers. In *Proceedings of the 2015 conference of the North American chapter of the association for computational linguistics: Demonstrations*, pages 1–5.

Yasuhisa Yoshida, Jun Suzuki, Tsutomu Hirao, and Masaaki Nagata. 2014. Dependency-based discourse parser for single-document summarization. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1834–1839.

Zhi-Hua Zhou. 2017. A brief introduction to weakly supervised learning. *National Science Review*, 5(1):44–53.