Data Programming for Learning Discourse Structure

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

[1]IRIT, [2]Linagora, [3]Université Toulouse III & CNRS
{*sonia.badene,kate.thompson,nicholas.asher*}*@irit.fr*, {*sbadene,jplorre*}*@linagora.com*

Abstract

This paper investigates the advantages and limits of data programming for the task of learning discourse structure. The data programming paradigm implemented in 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 training candidates. These results are later generalized using a discriminative model. Snorkel's attractive promise to create a large amount of annotated data from a smaller set of training data by unifying the output of a set of heuristics has yet to be used for computationally difficult tasks, such as that of discourse attachment, in which one must decide where a given discourse unit attaches to other units in a text in order to form a coherent discourse structure. Although approaching this problem using Snorkel requires significant modifications to the structure of the heuristics, we show that weak supervision methods can be more than competitive with classical supervised learning approaches to the attachment problem.

1 Introduction

Discourse structures for texts represent relational semantic structures that convey causal, topical, argumentative relations *inter alia* or more generally *coherence relations*. Following (Muller et al., 2012; Li et al., 2014; Morey et al., 2018), we represent them as dependency structures or graphs containing a set of nodes that represent *discourse units* (DUs), or instances of propositional content, and a set of labelled arcs that represent coherent relations between DUs. For dialogues with multiple interlocutors, extraction of their discourse structures could provide useful semantic information 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 on computational discourse-analysis (Duverle and Prendinger, 2009; Joty et al., 2013; Ji and Eisenstein, 2014; Surdeanu et al., 2015; Yoshida et al., 2014; Li et al., 2016), we are still a long way from usable discourse models, especially for dialogue. The problem of extracting full discourse structures is difficult: standard supervised models struggle to capture the sparse long distance 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 a weak supervision approach, data programming (Ratner et al., 2016), in which we exploit expert linguistic knowledge in a more compact and consistent rule-based form.

In our study, we restrict the structure learning problem to predicting edges or attachments between DU pairs in the dependency graph. After training a supervised deep learning algorithm to predict attachments on the STAC 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, Labeling Functions (LFs), with reference to this development set. Although the whole of the STAC corpus is annotated, we treated the remainder of the corpus as unseen/unannotated data in order to simulate the conditions in which the snorkel framework is meant to be used, i.e. where there is a large amount of unlabeled data but where it is only feasible to hand label a relatively small portion of it. Accordingly, we applied the completed LFs to our "unseen" training set, 80% of the corpus, and used the final 10% as our test set.

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

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 640–645
Florence, Italy, July 28 - August 2, 2019.

After applying the LFs to the unannotated data and training the generative model, the F1 score for attachment was 4 points higher than that for the supervised method, showing that hybrid learning architectures combining expert linguistic conceptual knowledge with data-driven techniques can be highly competitive with standard learning approaches.

2 State of the Art

Given that our interest lies in the analysis of multiparty dialogue, we followed (Afantenos et al., 2015; Perret et al., 2016) and used the STAC corpus, in which dialogue structures are assumed to be directed acyclical graphs (DAG) as in 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; ℓ: $E_1 \rightarrow R$ 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 which are connected as an ensemble to other DUs in the graph. As learning this 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.

To build an SDRT discourse structure, we need to: (i) segment the text into DUs; (ii) predict the attachments between DUs, i.e. identify the elements in E_1; (iii) predict 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: 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 on the discourse parsing attachment problem for monologue. It also targets a restricted version of an SDRT graph. It trains a simple MaxEnt algorithm to produce probability distributions over pairs of elementary discourse units, a "local model", with a positive F1 attachment score of 63.5; global decoding constraints produce a slight improvement in attachment scores. (Afantenos et al., 2015) uses a similar strategy on an early version of the STAC corpus. (Perret et al., 2016) targets a more elaborate approximation of SDRT graphs on a later version of the STAC corpus and reports a local model F1 attachment of .483. It then uses Integer Linear Programming (ILP) to encode global decoding constraints to improve the F1 attachment score (0.689).

(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, the latest version of STAC is a multi-modal 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.

[2]Segmented Discourse Representation Theory

3.2 Data Preparation

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

1. Roughly 56% of the dialogues in the corpus contain only non-linguistic DUs. The discourse structure of these dialogues are 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.

4 Data Programming Experiments

4.1 Candidates and Labeling Functions

Our weak supervision approach follows the Snorkel implementation of the data programming paradigm. The first step is candidate extraction, followed by LF creation. Candidates are the units of data for which labels will be predicted: all pairs of DUs in a dialogue for attachment problem in discourse. 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"). The LFs should have maximal and if possible overlapping coverage of the candidates to optimize the results of the generative model.

To predict dialogue attachment, our LFs exploit information about candidates including whether they are linguistic or non-linguistic DUs, the dialogue acts they express, their speaker identities, lexical content and grammatical category, as well as the distance between DUs: all features also used in supervised learning methods (Perret et al., 2016; Afantenos et al., 2015; Muller et al., 2012). Furthermore, our LFs take into account the particular behavior of each relation type, information that expert annotators consider when deciding whether two DUs are attached. Thus the LFs were divided among the 9 relation types as well as the combination of DU endpoints for each type, e.g. linguistic/non-linguistic. We also 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 complete rule set, along with descriptions of each of the relation types, is available here: https://tizirinagh.github.io/acl2019/.

In Table 1 we list the rules and their performances on the portion of the development set to which they apply. For example, the LF for Question-answer-pair between two linguistic endpoints (QAP LL) has a coverage of 32% – which is the proportion of the development set containing relations between two linguistic endpoints– and has an accuracy of 89%.

4.2 The Generative Model

Once the LFs are applied to all the candidates, we have a matrix of labels ($\overline{\Lambda}$) given by each LF Λ for each candidate. The generative model 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:

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

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

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

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

The generative model thus 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 doesn't always hold: one LF might be a variation of another or they might depend on a common source of information (Mintz et al., 2009). We will look at dependencies between LFs in future work.

Individual LF Performances						
	Coverage	True Pos	True Neg	False Pos	False Neg	Accuracy
QAP LL	0.32	282	9397	239	150	0.8928
QAP NLNL	0.31	84	9476	4	0	0.9995
Result NLNL	0.31	758	8636	134	36	0.9822
Result LNL	0.16	13	4596	319	97	0.9117
Result LL	0.32	21	9371	617	41	0.9345
Result NLL	0.21	2	6535	0	2	0.9996
Continuation LL	0.32	16	9818	110	106	0.9785
Continuation NLNL	0.31	613	8867	83	1	0.9912
Sequence NLL	0.21	82	6351	84	22	0.9837
Sequence NLNL	0.31	236	8199	1053	76	0.8819
Comment LL	0.32	123	8632	1140	0	0.8847
Comment NLL	0.21	12	6369	57	101	0.9758
Conditional LL	0.32	9	10026	7	0	0.9993
Elaboration LL	0.32	67	9694	214	75	0.9712
Elaboration NLNL	0.31	48	9420	96	0	0.9899
Acknowledgement LL	0.32	50	9612	251	137	0.9613
Contrast LL	0.32	14	9978	11	47	0.9942

Table 1: Performances of each LF on the development set. "Coverage" describes the percentage of the development set to which the LF applies, and is determined by the types of endpoints of the relation.

	Generative Model			Discriminative Model on Test	
	Dev	Train	Test	with Marginals	with Gold annotations
Precision	0.45	0.50	0.40	0.28	0.33
Recall	0.70	0.74	0.72	0.59	0.80
F1 score	0.55	0.59	0.51	0.38	0.47
Accuracy	0.87	0.88	0.84	0.74	0.75

Table 2: Evaluations of attachment with the weakly supervised and supervised approaches.

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.

The discriminative model we use in our study is a single layer BI-LSTM with 300 neurons, which takes as input 100 dimensional-embeddings for the text of each EDU 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 (ReLU) (Hahnloser et al., 2000; Jarrett et al., 2009; Nair and Hinton, 2010) activation and optimized with Adam (Kingma and Ba, 2014). Given that our data is extremely unbalanced in favor of the "unattached" class ("attached" candidates make up roughly 13% of the development set), we also implement 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.

In order to use this method, we had to binarize the marginals before moving to the discriminative step using a threshold of $p > .5$ (the threshold that gave us the best F1 score on the development corpus). 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 rebalancing was used, albeit much lower than expected overall.

5 Results and Analysis

We first evaluated our LFs individually on the development corpus, which permitted us to measure their coverage and accuracy on a subset of the data[3]. We then evaluated the generative model and the generative + discriminative model with the Snorkel architecture on the test set with the results in Table 2.

While our supervised discriminative model gave results on a par with the local model of (Perret et al., 2016) (which had an F1 of 0.483), our generative model (using a threshold value of $p > .5$ for positive attachment) had significantly better results, competitive with those in the literature on the attachment problem. Our models show strong recall but weaker precision than (Perret et al., 2016), and we believe this is in part because our LFs were expressly written to broadly cover relations and we have written very few rules on non-attachments.

The Snorkel coupling of generative and discriminative models did not produce the anticipated improvement over the results of generative model. When we trained the discriminative model directly on the marginals, we got a score of 0.26 for F1. To improve these results (column 4 in the Table 2), we used the class re-balancing method above. However in order to do this, we had to binarize the outputs of the generative model before training the discriminative model, which also contributed to lower precision scores by effectively reducing the total information available to the model.

6 Conclusions and Future Work

We have compared a weak supervision approach, as implemented in Snorkel, with a standard supervised model on the difficult task of discursive attachment. The results of the model from Snorkel's generative step surpass those of a standard supervised learning approach, showing it to be a promising method for generating a lot of annotated data in a very short time relative to what is needed for a traditional approach: from (Asher et al., 2016) we infer that the STAC corpus took at least 4 years to build; we created and refined our label functions in 2 months. 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 a discriminative model is supposed to afford.

In future work, we will enrich our weak supervision 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 exogenous decoding constraints applied in (Muller et al., 2012; Perret et al., 2016). We hope such experiments with the weak supervision paradigm will eventually lead us to understand how weakly supervised methods might effectively capture the global structural constraints on discourse structures without decoding or more elaborate learning architectures.

[3] https://tizirinagh.github.io/acl2019/

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.

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.

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)*, pages 13–24, Baltimore, Maryland. Association for Computational Linguistics.

Shafiq R Joty, Giuseppe Carenini, Raymond T Ng, and Yashar Mehdad. 2013. Combining intra-and multi-sentential rhetorical parsing for document-level discourse analysis. In *ACL (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*.

Qi Li, Tianshi Li, and Baobao Chang. 2016. Discourse parsing with attention-based hierarchical neural networks. In *EMNLP*, pages 362–371.

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, Thomas Hicks, and Marco A Valenzuela-Escárcega. 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, Doha, Qatar. Association for Computational Linguistics.

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