Programmable Annotation with Diversed Heuristics and Data Denoising

†Ernie Chang, *Alex Marin, †Vera Demberg
†Dept. of Language Science and Technology, Saarland University
*Microsoft Corporation, Redmond, WA
cychang@coli.uni-saarland.de

Abstract

Neural natural language generation (NLG) and understanding (NLU) models are costly and require massive amounts of annotated data to be competitive. Recent data programming frameworks address this bottleneck by allowing human supervision to be provided as a set of labeling functions to construct generative models that synthesize weak labels at scale. However, these labeling functions are difficult to build from scratch for NLG/NLU models, as they often require complex rule sets to be specified. To this end, we propose a novel data programming framework that can jointly construct labeled data for language generation and understanding tasks – by allowing the annotators to modify an automatically-inferred alignment rule set between sequence labels and text, instead of writing rules from scratch. Further, to mitigate the effect of poor quality labels, we propose a dually-regularized denoising mechanism for optimizing the NLU and NLG models. On two benchmarks we show that the framework can generate high-quality data that comes within a 1.48 BLEU and 6.42 slot F1 of the 100% human-labeled data (42k instances) with just 100 labeled data samples – outperforming benchmark annotation frameworks and other semi-supervised approaches.

1 Introduction

Modern machine learning systems require large amounts of labeled data. For many applications, such labeled data is created by getting humans to explicitly label each training example. However, the standard labeling process that involves Wizard-of-Oz (Kelley, 1984) and other crowd-sourcing approaches (e.g. (Wen et al., 2017; Coucke et al., 2018; Budzianowski et al., 2018)) is restricted to the level of individual examples, and so are slow and static (Ratner et al., 2019). As a result, they are not only costly but require relabeling for any fine-grained domain revisions.

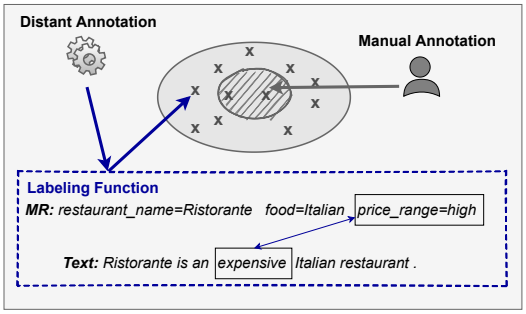

Figure 1: **Annotation scenario**: Each × represents a labeled data instance. The annotation framework allows to generalize from few human-labeled instances (inner) to large amounts of weakly labeled data (outer) by building rules (alignments) between sequence labels and text via labeling functions.

Data programming is a successful paradigm where humans provide low cost labeling rules written in programming languages to build imperfect training sets, which are then denoised for further improvements (Ratner et al., 2016a, 2017). However, two caveats exist: (1) Heuristic rules are costly to construct from scratch as exhaustive alignments between text and sequence labels have to be specified *manually* (Evensen et al., 2020), and especially strenuous for language generation that transforms meaning representations (MR) or slots into textual descriptions (e.g. (Reiter and Dale, 2000; Barzilay and Lapata, 2005)) where NLG is a one-to-many process (i.e.). (2) Labeling functions make decisions based on *discrete rules* that heavily limit the framework capability in making fine-grained decisions due to the lack of probabilistic information that guides the rule inference (Chatterjee et al., 2019).

To this end, we present a new data programming framework where language understanding and generation data can be *jointly labeled*. We argue that joint NLG/NLU annotation not only improves the overall data quality, but provide a greater degree of *compositionality* where semantic units such as slot-value pairs can be individually controlled. We tar-

Proceedings of the 29th International Conference on Computational Linguistics, pages 2681–2691
October 12–17, 2022.

get a weak supervision scenario (shown in Figure 1) consisting of small, high-quality expert-labeled data and a large set of unlabeled MR or text instances. In this framework, subject-matter experts are to use labeling rules to *modify* the automatically-inferred semantic alignments between MR and text, which are probabilistic rules that can be used for joint NLU/NLG labeling. The rules help to construct weak data via iterative denoising, before the dually-regularized NLU and NLG models can learn from the clean seed data to generate high-quality data. This work makes the following contributions:

- We introduce a novel annotation framework based on data programming that allows for joint labeling of language understanding and generation data. We validate the framework on two benchmarks and demonstrates its ability to create high quality data by expanding rules and then denoising the noisy data with dual regularization.

- We present a preliminary study to demonstrate the *compositionality* of the framework by showing that it can perform automatic domain revisions of MR slots without any relabeling efforts. This is especially beneficial in use of annotation tools where frequent data revisions are needed.

2 Related Work

Distant Supervision in Language Understanding. Learning with weak supervision is a well-studied area that is popularized by the rise of data-driven neural approaches (Ratner et al., 2017; Safranchik et al., 2020; Bach et al., 2017; Wu et al., 2018; Jiang et al., 2018). In particular, recent literature explore *knowledge distillation* from rules by either guiding the individual layers (Li and Srikumar, 2019) or training the model weights within constraints of the rule based system using a student and teacher model (Hu et al., 2016). Similarly, Snorkel and other techniques (Ratner et al., 2016b; Bach et al., 2017; Varma et al., 2019) rely on domain experts manually developing heuristics for noisy labels. However, these approaches are largely limited to NLU tasks and focus on providing discrete heuristics for tasks such as relations. Thus, our work serves to bridge this gap by (1) providing a way to more readily create texts, and (2) including probabilistic scores for labels.

Weak Supervision for Language Generation. Past works on semi-supervised learning consider settings with a large set of unlabeled data as in machine translation (Artetxe et al., 2017; Lample et al., 2017), or more relevantly the joint learning framework for training NLU and NLG (Tseng et al., 2020; Su et al., 2020; Schmitt and Schütze, 2019) that also considers a small labeled data. In particular, unsupervised statistical machine translation (e.g. (Artetxe et al., 2018; Lample et al., 2018)) utilizes statistical alignment models (Brown et al., 1993) that automatically infer explicit alignments between phrases in source and target sentences. Our work exploits this explicit alignment by treating them as a modifiable rule set[1], then using it to noisily synthesize weak data, allowing for annotation of NLU and NLG labels.

3 Annotation Framework Summary

Here we formally describe the joint annotation framework. Let X denote the set of meaning representation[2] (MR) instances and Y denote the text sequences. In our setting, we have (1) a seed dataset S which consists of k labeled pairs, and (2) a large unlabeled MR or text set U where $|U| \gg k > 0$. The annotation framework targets the creation of labeled samples $L = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ where $x_t \in X$ is an MR instance and $y_t \in Y$ is its corresponding text. To do so, we construct a probabilistic rule set based on the seed data S (see §4) and outline this process in Figure 2. In particular, we draw the connection between the rule set *compositionality* and *coverage* to data denoising and highlight the framework advantages. The resulting rule set allows to create a large set of noisily-labeled data, where the framework then learns from the mixture of clean (seed) and noisy data in the process of data denoising (see §5) to create a higher quality set of data.

4 Rule Set Construction

We use the rule set to define the semantic alignment between MR and text. For instance, in a "cuisine" domain, the *fast food* slot can be aligned with slot value "Macdonald" and other related terms, and likewise for each value that might be associated with more than one slot. This relationship can

[1] This is similar to phrase table pruning (Zens et al., 2012; Galbrun, 2009).

[2] They can be seen as sequence labels.

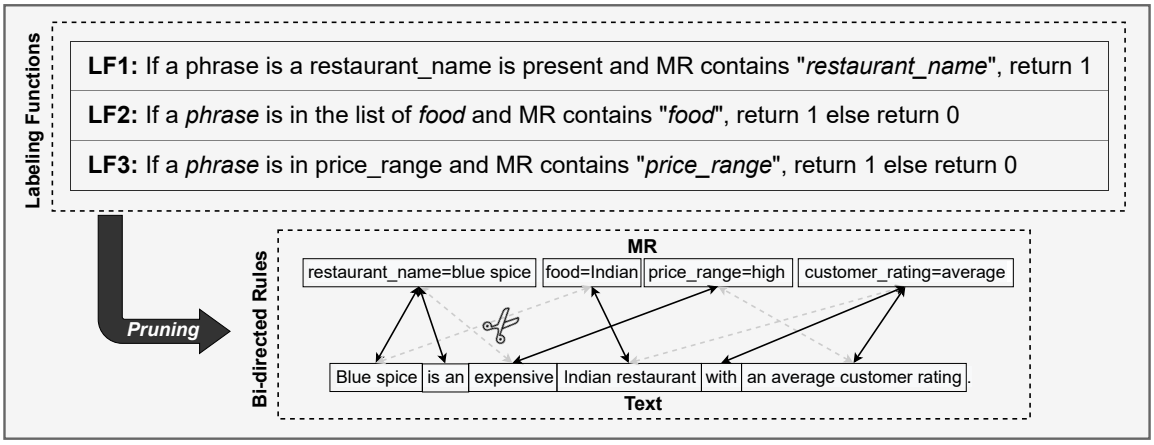

Figure 2: Depiction of the labeling process, where $\hat{x}$, $\hat{y}$ denote the noisy labels. Labeling functions are shown as textual descriptions, which are used to filter bi-directed rules below. In *bi-directed rules*, rules not removed by labeling functions are shown in **bold**. For more labeling functions see Appendix **??**.

be captured by the explicit many-to-many[3] rules derived from the MR-text pairs, which connects NLU and NLG as they are then explicitly linked with the alignments.

To build the rule set, we initialize the rule set in three steps: (1) We first generate new in-domain data using the pretrained language model (GPT-2) – this helps to increase data size and diversify the text distribution. (2) From this augmented data, we automatically initialize the rule set (phrase table) using a statistical alignment model. (3) Finally, we prompt the subject-matter experts to explicitly modify the alignments between phrases in MR and text with labeling functions, as these alignments help to refine the semantic relationships. Specifically, the aligned pairs are provided to experts where they are asked to write labeling functions to *prune* the non-matching phrases that contain incorrect semantic alignments (See Figure 2).

4.1 Diversifying Rule Set Diversity

To increase the diversity (and thus coverage) of the eventual annotation rule set generated, we perform an initial augmentation of the seed labeled data set S with additional weakly-labeled data, thus producing a larger but more diverse weakly-labeled dataset (L). We first generate additional MR via value swapping for each MR slot as in Chang et al. (2021a), then use GPT-2 (Radford et al., 2019) to perform conditional data-to-text generation as in Harkous et al. (2020); Mager et al. (2020). These works showed that fine-tuning GPT-2 on the joint distribution of MR and text for text-only generation yields decent performance. Given the sequential MR representation and a sentence in the seed labeled data, we maximize the joint probability $p_{\text{GPT-2}}(X, Y)$, where each sequence is concatenated into "[MR] $x_1 \cdots x_M$ [TEXT] $y_1 \cdots y_N$". The fine-tuned LM conditions on the augmented MR sample set to generate the in-domain text[4], and thus produces the augmented dataset with noisy texts. Similarly, for conditional MR generation with $p_{\text{GPT-2}}(Y, X)$, we apply the same process with text and MR flipped in the concatenation.

While the weak labels expands the seed data, *it creates noisy data by introducing false correlations between MR and text*. In what follows we discuss the creation of the rule set and the use of labeling functions that help to mitigate this noise.

4.2 Rule Set Initialization.

We extend the idea of a *phrase table* in statistical machine translation to be the rule set in our context: from the noisy augmented data L, we derive a rule set that is constructed based on the fertility-based[5] alignment model (GIZA++) (Och and Ney, 2003)[6] optimized using the EM algorithm (Dempster et al., 1977). This allows to obtain the semantic correspondences (or probabilistic rule set) $R_{i \to j} \subseteq \{(i, j, P_{i \to j}) : i = 1 \cdots |x|; j = 1 \cdots |y|\}$ where i and j refer to positions in flattened MR x and text sequence y, and $P_{i \to \hat{j}}$ is the probability of

[3]Relatedly, NLU is many-to-one and NLG is one-to-many (Tseng et al., 2020).

[4]We adopt the top-k random sampling with both $k = 2$ and $k = 15$ to encourage diversity and ensuring correct outputs (Radford et al., 2019)

[5]Fertility is defined as the number of words that correspond to a semantic unit.

[6]http://www.statmt.org/moses/giza/GIZA++.html

aligning the i^{th} semantic unit in x to the j^{th} unit in y. Each semantic correspondence can be seen as a bi-directed edge of a *rule* that connects semantic units (or phrases) in MR to phrases in text, where the granularity of each semantic unit is determined by the feature-based phrase-based alignment model (Brown et al., 1993). Thus, we can likewise induce $R_{j\rightarrow i} \subseteq \{(j, i, P_{j\rightarrow i})\}$ as the rule set that can be *derived for either NLU or NLG* inferences. We discuss the resolution of potential conflicts between rules in §5.

4.3 Building Labeling Functions

We ask a group of annotators to write Python code snippets within the time limit of an hour each. We denote these code snippets as labeling functions (LFs). Each labeling function follows basic rule relations, as shown in figure 2, and returns one of the possible values: $1, -1, 0$ ("valid"/"not valid"/"undetermined"). To prune the rule set, we apply the labeling functions to each rule, and judge whether the given rule is *strictly invalid* or not. Two types of LFs are designed: (1) *slot-specific LFs*: one LF is written for each slot identified in the MR, as all the decisions related to the slot can be grouped together. For certain slots, such as "restaurant_name", a basic dictionary of correct slot values is collected as to verify if mappings are correct. (2) *general LFs* help to eliminate false rules across all slots. One example is the rule that links a conjunction of text with a specific slot of the MR. Using the general LFs, such incorrect rules can be removed altogether. Rules marked as "undetermined" by the LFs are preserved and will be further evaluated by the denoising mechanism. For instance, the phrase "is a" may remain aligned with "restaurant_name" as no LF may have marked it as "not valid".

4.4 Rule Coverage and Compositionality

MRs are structured and compositional as they typically consist of attributes in the form of flat or tree-structured slot-value pairs (Balakrishnan et al., 2019). Here, we define the *compositionality* of a rule set as the average percentage of rules that correctly correlate MR and text over all slots. In our framework, rule sets can be manipulated to directly reflect high-level requirements in dialogue – such as the need to remove or add values to a slot in both MR and text for domain revisions. For slot removal, this can be done directly via the addition of labeling functions to remove specific attributes from MR and text altogether. An important trade-off exists between *rule coverage* and *compositionality* – higher rule coverage leads to lower compositionality, since a larger rule set tends to make more erroneous correlations between MR and text.

In what follows we describe the process of *data denoising*, in which the framework learns to utilize both the noisy and clean data.

5 Data Denoising

Label bias is a well-studied problem (Lafferty et al., 2001) where the frequency of some transitions will far outweigh the others even when they have roughly the same probability mass. Beneficially, the joint use of both NLU/NLG models helps to mutually shape the probability masses in token-label pairs. Figure 3 represents a simple finite-state model designed to map the two words "good" and "review" to their respective labels. Suppose that the observation sequence is "good review". From starting states 0 and 1, "good" matches both potential label transitions "price" and "rating", so the probability mass gets distributed rather equal among those two transitions.

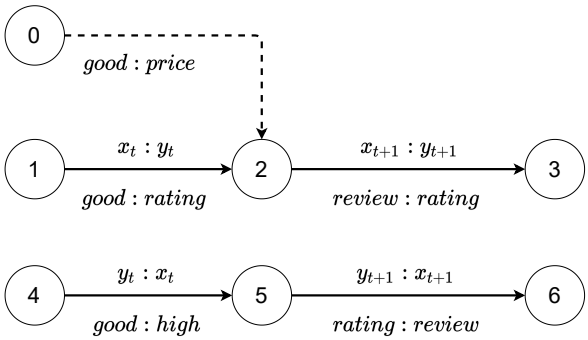

Figure 3: **Example of resolving label bias**: we place the token-label pairs (x_t:y_t) on transitions rather than states.

While the two paths will be roughly equally likely, but if one of the two labels is slightly more common in the training set (i.e. price), the transitions out of the start state will slightly prefer this corresponding transition, and yielding incorrect correlations. However, having both NLU and NLG models alleviate this bias since we now have two "versions" of the sequence in states $\{1, 2, 3\}$ and $\{4, 5, 6\}$ that can help to break the ties and shape the probability mass according to the sequential information.

Thus, we initialize noisy data with rules using the statistical *base models* and then training the *denoising models* to distill from the mixture of

noisy and clean data. This is done by iteratively creating noisy labels with rules and then refining these labeled data for higher quality ones. Further, we maintain both statistical NLU and NLG models for iterative joint labeling in the process.

5.1 Overview of the Denoising Mechanism

The denoising mechanism is achieved by first formulating the NLG model $p(y|x)$ as Brown et al. (1993):

$$p(y|x) \approx \sum_{j=1}^{m=|y|} \prod_{i=0}^{l=|x|} t(y_j|x_i)a(i|j,m,l) \quad (1)$$

where $t(\cdot)$ and $a(\cdot)$ are the translation and alignment probabilities learned from the seed data for x and y of lengths m and l respectively. Note that for the NLU model, we simply flip y and x in Eq. 1. Thus the formulation allows to estimate both statistical NLU and NLG models by adjusting the weights of log-linear combination of features[7] to optimize the evaluation metric (i.e. BLEU-4 (Papineni et al., 2002)) on the validation corpus via the minimum error rate training (MERT) (Och, 2003), which maximizes the BLEU-4 scores based on given inputs and their noisy labels.

In what follows we describe the training objectives to distill knowledge from the rule set into the base models (**Step-1**), then introduce clean data in **Step-2** to improve upon the data quality.

Step-1: Distillation from Noisy Rules. We optimize the parameters of the base version of the NLU and NLG models (**base models**) by alternately fixing the parameters of one model and optimizing the other model until convergence[8]: (1) MERT computes the optimal value for each model parameter and greedily selects data based on the generated candidate labels that leads to the largest gain in BLEU-4. (2) Then it noisily labels the MR samples with text or text with MR via the updated parameters at each iteration so as to obtain a better approximation of label candidates. The process of data denoising is functionally beneficial in reducing the label biases present in the imperfect labels.

Step-2: Adding Clean Data. The **denoising models** are trained with the expert-labeled seed data and the set of noisy data that were generated. However, the performance on the seed data is better than the pseudo-labels in the early rounds. This is anecdotally observed in both NLU and NLG models on various datasets and leads to potentially suboptimal performance (Shen and Sanghavi, 2019a,b). This motivates our proposal to reduce the total loss by using NLU and NLG models to select only a subset of data to train on – we filter out samples with large cross-entropy losses in early iterations *with replacement*, and train the models on the samples left after filtering. This serves to learn more effectively from both clean and corrupted data.

Specifically, we propose the use of *dual regularization (DR)* where $s_1, \ldots, s_n$ are the samples, θ are the model parameters:

$$\hat{\theta}^{(\text{DR})} = \arg\min_{\theta} \Big[\min_{\tilde{X} \subset X : |\tilde{X}| = \lfloor \beta * n \rfloor} \sum_{i \in \tilde{X}} L_{NLG_\theta(x_i)} \Big] + \arg\min_{\theta} \Big[\min_{\tilde{Y} \subset Y : |\tilde{Y}| = \lfloor \beta * n \rfloor} \sum_{i \in \tilde{Y}} L_{NLU_\theta(y_i)} \Big].$$

To find $\hat{\theta}^{(\text{DR})}$, we minimize over both the (a) sample subsets $\tilde{X}, \tilde{Y}$ given the ratio β and (b) the model parameters θ. In (a), the MR-text sample size is $\lfloor \beta * n \rfloor$, where $\beta = 0.1$[9] is the ratio of training samples to train on. $\tilde{X}, \tilde{Y}$ are the subsets of X, Y selected for the NLU and NLG models.

In this formulation, NLU and NLG models jointly select samples that are deemed to be less challenging for each other, before proceeding to learn from more challenging samples[10]. As such, the NLU and NLG models are kept to be similar in inference capabilities; this allows to select more suitable samples for each other due to the smaller degree of semantic misalignment.

6 Experiment Setting

We conduct experiments on the Weather (Balakrishnan et al., 2019) and E2E (Novikova et al., 2017b) datasets. Weather contains 25k instances of tree-structure annotations. E2E is a crowd-sourced dataset containing 50k instances in the restaurant domain. The NLU and NLG models are implemented in PyTorch (Paszke et al., 2019) with 2 Bi-LSTM layers and 100-dimensional token embeddings and Adam optimizer (Kingma and Ba, 2014) with an initial learning rate of 0.0002. The

[7] This includes the bidirectional *rule*, *lexical probabilities*, the *language model*, the *reordering model*, the *word penalty* and the *phrase penalty*.

[8] Following the training procedures in Artetxe et al. (2018).

[9] Following (Shen and Sanghavi, 2019b), we set β to be 1%, as it was shown to work well in generative models.

[10] It is related to *competence-based curriculum learning* (Platanios et al., 2019) where samples are selected based on their difficulties and the model competence.

	MR → Text (NLG)							Text → MR (NLU)						
Text-only	100	150	200	250	300	SR	VE	100	150	200	250	300	SR	VE
seq2seq	35.55	41.79	41.63	43.50	46.62	39.20	40.61	47.61	51.33	51.94	52.53	53.83	52.38	56.84
JUG	37.62	42.32	42.01	45.42	48.27	40.88	42.64	48.60	53.40	52.78	52.44	54.23	54.38	56.56
stats (iter-10)	47.49	48.10	48.68	49.71	50.09	47.72	55.48	51.87	54.30	56.43	55.67	56.52	55.88	59.43
stats w/ GPT2 (iter-10)	52.81	53.77	54.11	54.24	55.27	50.21	57.31	53.91	56.62	59.15	56.97	58.40	59.00	60.36
stats (iter-20)	54.09	56.72	56.25	57.87	57.46	53.24	57.30	54.78	59.83	60.16	58.39	61.96	61.50	62.81
+ step-1	54.33	56.28	57.39	57.62	58.51	57.49	57.28	59.11	60.03	61.43	61.00	62.93	63.89	64.79
+ seed data	56.71	58.68	**59.42**	60.63	61.98	60.75	60.48	60.25	63.75	**64.55**	63.21	63.97	66.95	66.59
+ seed data & step-2 (**Ours**)	**60.35**	**61.44**	59.10	**62.32**	**63.71**	**62.55**	**61.33**	**62.63**	**64.33**	64.34	**65.33**	**67.17**	**68.89**	**68.27**

	MR → Text (NLG)							Text → MR (NLU)						
MR-only	100	150	200	250	300	SR	VE	100	150	200	250	300	SR	VE
seq2seq	31.05	32.92	33.53	36.27	35.09	31.43	30.04	48.61	52.33	52.94	53.53	54.83	53.38	57.84
JUG	35.05	36.84	36.66	39.61	38.5	35.96	33.17	51.02	55.26	55.74	55.07	55.92	55.68	59.14
stats (iter-10)	38.70	40.08	40.42	43.18	41.76	42.36	40.35	53.74	56.95	58.43	57.28	58.78	57.77	60.61
stats w/ GPT2 (iter-10)	41.84	43.20	43.83	46.27	45.67	45.64	43.60	56.24	59.06	61.15	58.29	61.16	60.69	62.59
stats (iter-20)	45.69	46.40	46.84	49.82	49.52	49.17	46.87	57.46	61.26	62.21	60.36	63.98	62.58	64.88
+ step-1	49.66	50.40	50.74	53.09	53.26	52.40	49.93	60.14	62.69	63.38	62.71	65.53	65.15	66.19
+ seed data	53.60	**55.19**	54.32	56.24	57.03	55.92	53.15	62.16	**65.74**	65.65	64.93	66.86	67.99	68.41
+ seed data & step-2 (**Ours**)	**54.77**	54.92	**57.78**	**58.41**	**61.32**	**56.68**	**57.39**	**64.47**	65.38	**67.10**	**67.51**	**69.12**	**70.04**	**69.42**

Table 1: Ablation studies for text generation/NLG (BLEU-4) and slot filling/NLU (F1) on the E2E corpus with increasing amounts of manually-annotated data (100-300 samples). We show the performance increase to the base model initialized from the rule set (**stats**) as **GPT2** augmentation, statistical NLG/NLU models with distillation from **stats** (**step-1**), and dually-regularized sample selection (**step-2**) are added. Domain revisions are performed with 300 data instances. We train the following on the seed data for comparison: (1) a semi-supervised baseline, **JUG** (Tseng et al., 2020) and (2) a LSTM-based baseline (**seq2seq**).

		Slot Filling		Text Generation			
		F1(%)	Wrong	BLEU4	Naturalness	Wrong	Diversity
E2E	reference	-	-	-	4.51	0	53.89
	SLUG+100%	-	-	55.30	4.37	7	46.72
	JUG+100%	73.7	29	57.72	4.49	38	46.21
	seq2seq+100%	73.19	31	56.1	4.32	35	43.09
	GPT2	54.83	55	40.84	4.23	65	**44.55**
	Heuristic	62.81	39	53.08	3.82	**19**	31.37
	COACH	48.35	49	-	-	-	-
	seq2seq+Snorkel	60.71	43	-	-	-	-
	seq2seq+**Ours**	**66.42**	**36**	**54.62**	**4.39**	22	40.65
Weather	reference	-	-	-	4.30	0	40.97
	JUG+100%	67.09	11	51.43	3.30	14	33.61
	seq2seq+100%	66.43	8	46.29	4.10	9	35.74
	GPT2	36.51	46	34.01	**3.95**	35	**40.28**
	Heuristic	50.33	29	38.83	3.40	**16**	31.45
	COACH	46.21	32	-	-	-	-
	seq2seq+Snorkel	47.61	27	-	-	-	-
	seq2seq+**Ours**	**54.71**	**22**	**44.63**	3.80	23	36.76

Table 2: **Performance** and **human evaluation** comparing **Ours** with the benchmarks in the **text-only** scenario with 300 training samples evaluated on the test samples. We count the number of *wrong* slot-value pairs; and *naturalness* is based on average of 15 human ratings on a scale of 5. *Diversity* is the mean segmental type-token ratio (size=25) (Covington and McFall, 2010). **100%** indicates models trained on 100% manual annotation (also **highlighted in gray**). **A+B** indicates training the **A** model on the data generated by approach **B**.

scores are averaged over 10 random initialization runs. Two *subject-matter experts* are employed to construct labeling functions given labeled instances for 1-hour of labeling time. The labeling functions obtained were used for all subsequent scenarios.

Experimental Scenarios. We conduct experiments on two few-shot scenarios (see Table 1): **Text-only** consists of only unlabeled text; **MR-only** is given unlabeled MR alone. Both scenarios are given a small amount of clean, manual-annotated data consisting of MR-text pairs.

To demonstrate the framework's ability to perform *domain revisions* without relabeling, we explore two situations under the few-shot settings: (1) in slot removal (**SR**), we remove the "*customer rating*" slot by selecting from the original seed/dev/test sets. (2) in slot value enhancement (**VE**), we introduce additional restaurant names through relabeling the data. We release the data alongside our code. Note that we selected up to 300 training samples so as to simulate a low resource scenario. We display some examples of **SR** and **VE** in Figure 3.

We compare the performance of our framework with additional benchmark systems on both E2E and Weather datasets in Table 2. Note that our framework is only given the seed data and the additional unlabeled *text* or *MR* samples, while some models are trained with up to 100% of the data. The NLU benchmark systems include a baseline sequence-to-sequence model (**seq2seq**), and **COACH** (Liu et al., 2020) and a baseline data programming framework (**Snorkel**) (Ratner et al., 2017), both state-of-the-art systems on few-shot settings. For NLG, we included a **heuristic** labeler, a **GPT2** labeler (Harkous et al., 2020), and the high-performing SLUG (Juraska et al., 2018) on the E2E data. The **Heuristic** labeler was built on top of the labeling functions, but was extended to be a complete generative system. To compare with models capable of performing both NLG

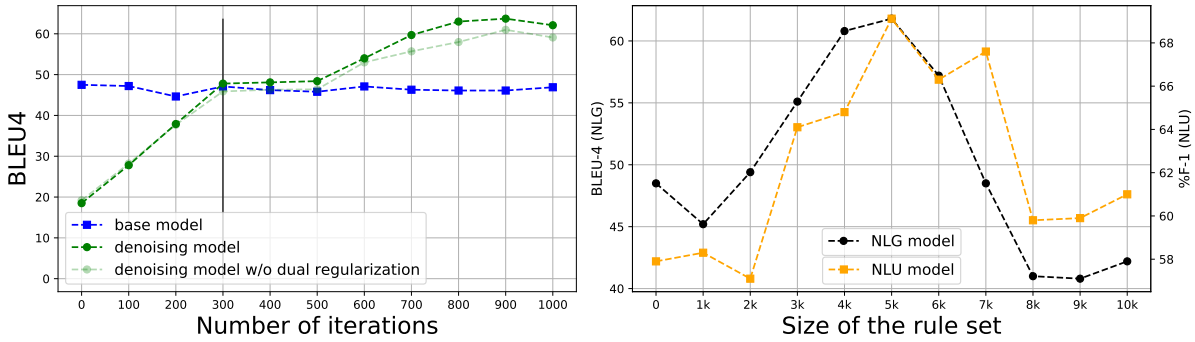

Figure 4: **Left:** This plot indicates the process of *knowledge distillation* (or gain in performance in BLEU-4) from the statistical *base models* to the *NLG model* as a function of the number of iterations. **Vertical line** indicates the addition of clean data. **Right:** we display the *modeling advantage* which showcase the improvement in performance as a function of the number of labeling functions. Both plots are based on 300 clean data.

MR & Text
Slot Removal (SR): **[MR]** name[the rice boat], food[italian], priceRange[cheap], area[riverside], familyFriendly[no], customerRating[average], near[express by holiday inn] **[Text]** the rice boat is located near express by holiday inn in riverside that serves italian food at a low price range .
Slot Value Enhancement (VE): **[MR]** name[the rice boat], food[italian], priceRange[cheap], area[riverside], familyFriendly[no], customerRating[average], near[cobalt lane 32092] **[Text]** the rice boat is located at the cobalt lane 32092, which is next to the riverside that serves italian food at a low price range .

Table 3: Samples of heuristically-based annotation for the revised domain for **SR** and **VE**.

and NLU, we include JUG (Tseng et al., 2020), which is a semi-supervised multi-task framework that allows to perform inference on both NLU and NLG. Table 2 contrasts the performance between **seq2seq** trained on 100% human annotation (**seq2seq+**100%) and of the data generated by our framework (**seq2seq+Ours**).

7 Results and Analysis

The results shown in tables 1 and 2 demonstrate the flexibility of our framework to perform annotation in both *text-only* and *MR-only* scenarios. Moreover, we see in Table 2 that, with as little as 300 data points, the framework is able to produce quality data[11] that allow the baseline **seq2seq** model to come close to the performance of the *same model* trained on full manual annotations; the combination reaches within 1.48 BLEU on NLG and 6.42% F1 score for NLU. Moreover, the framework produces high quality data that effectively mitigate the noise induced by automatic weak annotation, and manages to generate *natural* and *diverse* text for NLG purposes.

[11]The train-dev-test samples are 30-100-100 for the slot manipulation.

In Table 1, we first observe that adding GPT-2 augmentation does diversify the text and improves performance on both datasets maximally by 4.53 BLEU. The augmented system is used to initialize the base model's next iteration, and thus observe that base models can be iteratively enhanced. This is reflected across different sizes of seed data; the effect of iterative denoising is most prominent with seed data size$=$ 300. Next, we see that the use of denoising helps to further improve the models, as it allows to learn from both the base model's initialization and from the effect of noisy labeling. The base model, in this case, serves as the *teacher model* that guides the denoising models to iteratively improve. As the knowledge is completely distilled, we see that the denoised data performs slightly better than the base model (see left of Figure 4), having learned to search through the space of decoding for more optimal paths. We find that, after 20 iterations, the improvements become marginal for both datasets. Thus, we end our experiments at *iter=20*. We also include the expert-labeled (seed) data during denoising as an addition to the large set of pseudo-labeled (noisy) data. This brings about maximally a 3.47 BLEU (for NLG)

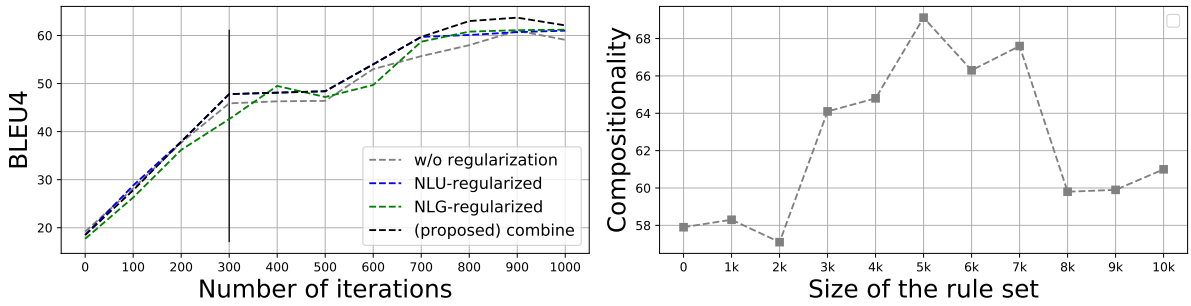

Figure 5: **Left:** This plot indicates the effect of dually-regularized sample selection. **Right:** The plot displays the trade-off between rule coverage and compositionality, where *compositionality* is defined as the average percentage of rules that correctly correlate MR and text over all slots.

and 3.72% F1 score improvement (for NLU) in **Text-only**, and 3.77 BLEU (for NLG) and 2.84% F1 score increase (for NLU) on the **MR-only** scenario. With the proposed dually-regularized sample selection, we further boost the performance by up to 4.29 BLEU and 2.58% F1 and an average of 2.37 BLEU improvements for NLG. This shows the efficacy of the proposed approach in modulating the effect of noisy and clean data.

Since the **Text-only** scenario consists of high-quality manually-labeled text, it generally performs better in NLG; similarly, in the **MR-only** scenario, NLU performance is generally better as MR samples are ground-truth labels. However, this effect is less prominent in the **MR-only** scenario as the difference between ground-truth MR samples and the weakly-labeled ones are often negligible.

Error Analysis. Word-level overlapping scores (BLEU-4) usually correlate rather poorly with human judgements on fluency and information accuracy (Reiter and Belz, 2009; Novikova et al., 2017a). Thus, we perform human evaluation on the E2E corpus on 100 sampled generation outputs, as seen in Table 2. We show that, with 100 instances, the denoising models yield significantly fewer *wrong* slot errors, while having more *natural* and *diverse* outputs. Moreover, we observe that benchmark systems (e.g. COACH) fail to generalize from the small seed data, and suffers heavily in terms of using the *wrong* facts (or slots).

8 Further Analysis

Analysis of Modeling Advantage. We further explore the relationship between performance and the number of labeling functions in the right plot of Figure 4. At one extreme, very few number of labeling functions will result in a very noisy set of rules, which leads to poorly labeled data. We find that, as the number of labeling functions grow, the *capability of the framework to denoise* the initial inferred rules improves. This continues until eventually the framework's denoising capability reaches its peak and starts to deteriorate – as some labeling functions eliminate a useful subset of alignments, as represented by the rule set.

Analysis of Dual Regularization. To analyze the process of sample selection during the training of denoising models, we experiment with selecting samples based on the cross-entropy loss from (1) NLU model, (2) NLG model or (3) the combined use of NLU and NLG models for sample selection, which is the proposed approach. We also compare them with performance without sample selection to show the contrast. In Figure 5 we show this comparison. We observe that selection based on either NLU or NLG model is not sufficient to match the performance of selection using both models. This shows that it is crucial to ensure that the NLG and NLU models learn at approximately the same rate, thereby allowing the semantic alignments induced from the base models to be preserved.

On *Rule Coverage* and *Compositionality* Trade-Offs. As discussed in section 4.4, we evaluate the framework limitations in composing semantic alignments (*compositionality*) as the number of rules becomes high. In Figure 5, we show that the number of rules influence the proportion of rules that correctly align MR and text, as indicated by the percentage of compositionality. In particular, with no labeling functions, the entire rule set is used and this leads to poor performance as most slots are being incorrectly aligned. As more labeling functions are introduced to reduce this rule set to its useful subset, the models begin to construct

better data with the right alignments. The labeling functions thus serve to remove the incorrect, low-impact rules, so that the high impact rules can play a greater role in constructing the necessary semantic alignments for both NLU and NLG – until too little rules remain to construct the base model.

9 Limitations

Overall, while we observe effectiveness in the proposed approach to recreate data, it remains to say that the constructed texts suffer from two main drawbacks: First, the diversity of the text is rather limited by the original seed set, which in turn constrains the data augmentation process that intend to enrich the text diversity. Second, the process of creating programmable labeling functions can indeed be a cumbersome process – it relies heavily on the adequate skill sets of the annotators who need to understand the target domain and basic scripting in order to proceed. It is then vital to ease the process of programming script writing, and reuse functions as much as possible to avoid overheads.

10 Conclusion and Future Work

In this paper, we show the efficacy of the framework where both NLU and NLG data can be jointly and automatically labeled to construct high quality data. We also demonstrate that the framework is receptive to the changes in MR slots, allowing for automatic domain revisions of MR and text data. Importantly, we observe that the success of the framework depends on finding the right balance between the number of labeling functions and the inherent level of compositionality of the data to be labeled. Thus, for future work we intend to focus on identifying the level of compositionality and predicting the threshold number of labeling functions necessary for decent performance, potentially manipulating the inherent graphical relationships (Hong et al., 2019). Moreover, the initial seed set in our experiments are assumed to be present, it is therefore necessary to first sample unlabeled data based on difficulty to annotate and the performance considerations (Chang et al., 2021b,c), before fine-tuning with pretrained language models which have strong priors for better quality data (Chang et al., 2022b,a).

Acknowledgements

This research was funded in part by the German Research Foundation (DFG) as part of SFB 248 "Foundations of Perspicuous Software Systems". We sincerely thank the anonymous reviewers for their insightful comments that helped us to improve this paper.

References

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. Unsupervised statistical machine translation. *arXiv preprint arXiv:1809.01272*.

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2017. Unsupervised neural machine translation. *arXiv preprint arXiv:1710.11041*.

Stephen H Bach, Bryan He, Alexander Ratner, and Christopher Ré. 2017. Learning the structure of generative models without labeled data. *Proceedings of machine learning research*, 70:273.

Anusha Balakrishnan, Jinfeng Rao, Kartikeya Upasani, Michael White, and Rajen Subba. 2019. Constrained decoding for neural nlg from compositional representations in task-oriented dialogue. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 831–844.

Regina Barzilay and Mirella Lapata. 2005. Modeling local coherence: An entity-based approach. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 141–148, Ann Arbor, Michigan. Association for Computational Linguistics.

Peter F Brown, Stephen A Della Pietra, Vincent J Della Pietra, and Robert L Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2):263–311.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. 2018. Multiwoz-a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026.

Ernie Chang, Jesujoba Alabi, David Adelani, and Vera Demberg. 2022a. Few-shot pidgin text adaptation via contrastive fine-tuning. In *Proceedings of COLING 2022*.

Ernie Chang, Vera Demberg, and Alex Marin. 2021a. Jointly improving language understanding and generation with quality-weighted weak supervision of automatic labeling. In *EACL 2021*.

Ernie Chang, Alex Marin, and Vera Demberg. 2022b. Improving zero-shot multilingual text generation via iterative distillation. *Proceedings of COLING 2022*.

Ernie Chang, Xiaoyu Shen, Alex Marin, and Vera Demberg. 2021b. The selectgen challenge: Finding the best training samples for few-shot neural text generation. In *Proceedings of the 14th INLG*, pages https–aclanthology.

Ernie Chang, Xiaoyu Shen, Hui-Syuan Yeh, and Vera Demberg. 2021c. On training instance selection for few-shot neural text generation. In *Proceedings of ACL 2021*.

Oishik Chatterjee, Ganesh Ramakrishnan, and Sunita Sarawagi. 2019. Data programming using continuous and quality-guided labeling functions. *arXiv preprint arXiv:1911.09860*.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, et al. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv:1805.10190*.

Michael A Covington and Joe D McFall. 2010. Cutting the gordian knot: The moving-average type–token ratio (mattr). *Journal of quantitative linguistics*, 17(2):94–100.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22.

Sara Evensen, Chang Ge, Dongjin Choi, and Çağatay Demiralp. 2020. Data programming by demonstration: A framework for interactively learning labeling functions. *arXiv preprint arXiv:2009.01444*.

Esther Galbrun. 2009. *Phrase table pruning for statistical machine translation*. Ph.D. thesis, University of Helsinki.

Hamza Harkous, Isabel Groves, and Amir Saffari. 2020. Have your text and use it too! end-to-end neural data-to-text generation with semantic fidelity. *arXiv preprint arXiv:2004.06577*.

Xudong Hong, Ernie Chang, and Vera Demberg. 2019. Improving language generation from feature-rich tree-structured data with relational graph convolutional encoders. In *Proceedings of the 2nd Workshop on Multilingual Surface Realisation (MSR 2019)*, pages 75–80.

Zhiting Hu, Xuezhe Ma, Zhengzhong Liu, Eduard Hovy, and Eric Xing. 2016. Harnessing deep neural networks with logic rules. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2410–2420.

Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. 2018. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International Conference on Machine Learning*, pages 2304–2313.

Juraj Juraska, Panagiotis Karagiannis, Kevin Bowden, and Marilyn Walker. 2018. A deep ensemble model with slot alignment for sequence-to-sequence natural language generation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 152–162.

John F Kelley. 1984. An iterative design methodology for user-friendly natural language office information applications. *ACM Transactions on Information Systems (TOIS)*, 2(1):26–41.

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

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2017. Unsupervised machine translation using monolingual corpora only. *arXiv preprint arXiv:1711.00043*.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Phrase-based & neural unsupervised machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5039–5049.

Tao Li and Vivek Srikumar. 2019. Augmenting neural networks with first-order logic. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 292–302.

Zihan Liu, Genta Indra Winata, Peng Xu, and Pascale Fung. 2020. Coach: A coarse-to-fine approach for cross-domain slot filling. *arXiv preprint arXiv:2004.11727*.

Manuel Mager, Ramón Fernandez Astudillo, Tahira Naseem, Md Arafat Sultan, Young-Suk Lee, Radu Florian, and Salim Roukos. 2020. Gpt-too: A language-model-first approach for amr-to-text generation. *arXiv preprint arXiv:2005.09123*.

Jekaterina Novikova, Ondřej Dušek, Amanda Cercas Curry, and Verena Rieser. 2017a. Why we need new evaluation metrics for nlg. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2241–2252.

Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. 2017b. The e2e dataset: New challenges for end-to-end generation. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 201–206.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for*

Computational Linguistics, pages 160–167, Sapporo, Japan. Association for Computational Linguistics.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational linguistics*, 29(1):19–51.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035.

Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom Mitchell. 2019. Competence-based curriculum learning for neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1162–1172.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8).

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 Ratner, Braden Hancock, Jared Dunnmon, Frederic Sala, Shreyash Pandey, and Christopher Ré. 2019. Training complex models with multi-task weak supervision. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4763–4771.

Alexander J Ratner, Christopher M De Sa, Sen Wu, Daniel Selsam, and Christopher Ré. 2016a. Data programming: Creating large training sets, quickly. In *Advances in Neural Information Processing Systems 29*.

Alexander J Ratner, Christopher M De Sa, Sen Wu, Daniel Selsam, and Christopher Ré. 2016b. Data programming: Creating large training sets, quickly. *Advances in neural information processing systems*, 29:3567–3575.

Ehud Reiter and Anja Belz. 2009. An investigation into the validity of some metrics for automatically evaluating natural language generation systems. *Computational Linguistics*, 35(4):529–558.

Ehud Reiter and Robert Dale. 2000. *Building natural language generation systems*. Cambridge university press.

Esteban Safranchik, Shiying Luo, Stephen H Bach, Elaheh Raisi, Stephen H Bach, Stephen H Bach, Daniel Rodriguez, Yintao Liu, Chong Luo, Haidong Shao, et al. 2020. Weakly supervised sequence tagging from noisy rules. In *AAAI*, pages 5570–5578.

Martin Schmitt and Hinrich Schütze. 2019. Unsupervised text generation from structured data. *arXiv preprint arXiv:1904.09447*.

Yanyao Shen and Sujay Sanghavi. 2019a. Iterative least trimmed squares for mixed linear regression. In *Advances in Neural Information Processing Systems*, volume 32, pages 6078–6088. Curran Associates, Inc.

Yanyao Shen and Sujay Sanghavi. 2019b. Learning with bad training data via iterative trimmed loss minimization. In *Proceedings of the 36th International Conference on Machine Learning*, pages 5739–5748.

Shang-Yu Su, Chao-Wei Huang, and Yun-Nung Chen. 2020. Towards unsupervised language understanding and generation by joint dual learning. *ACL*.

Bo-Hsiang Tseng, Jianpeng Cheng, Yimai Fang, and David Vandyke. 2020. A generative model for joint natural language understanding and generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1795–1807.

Paroma Varma, Frederic Sala, Ann He, Alexander Ratner, and Christopher Ré. 2019. Learning dependency structures for weak supervision models. *arXiv preprint arXiv:1903.05844*.

Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gasic, Lina M Rojas Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2017. A network-based end-to-end trainable task-oriented dialogue system. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 438–449.

Yu Wu, Wei Wu, Zhoujun Li, and Ming Zhou. 2018. Learning matching models with weak supervision for response selection in retrieval-based chatbots. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 420–425.

Richard Zens, Daisy Stanton, and Peng Xu. 2012. A systematic comparison of phrase table pruning techniques. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 972–983.