Advancing Seq2seq with Joint Paraphrase Learning

So Yeon Min [*]
MIT CSAIL
MIT-IBM Watson AI Lab
symin95@alum.mit.edu

Preethi Raghavan
IBM Research, Cambridge
MIT-IBM Watson AI Lab
praghav@us.ibm.com

Peter Szolovits
MIT CSAIL
MIT-IBM Watson AI Lab
psz@mit.edu

Abstract

We address the problem of model generalization for sequence to sequence (seq2seq) architectures. We propose going beyond data augmentation via paraphrase-optimized multi-task learning and observe that it is useful in correctly handling unseen sentential paraphrases as inputs. Our models greatly outperform SOTA seq2seq models for semantic parsing on diverse domains (Overnight - up to 3.2% and emrQA - 7%) and Nematus (Sennrich et al., 2017), the winning solution for WMT 2017, for Czech to English translation (CzENG 1.6 - 1.5 BLEU).

1 Introduction

Natural language provides a vast number of alternative ways to state something or to ask a question (Bhagat et al., 2009). This poses a daunting challenge to natural language processing methods because there is no possible way to enumerate all these alternatives. As a result, many popular machine learning systems trained on benchmark datasets are surprisingly fragile to such previously unobserved variations of the training input at test time (Jia and Liang, 2017; Belinkov and Bisk, 2017; Goodfellow et al., 2014; Iyyer et al., 2018)[1].

An attempt to ameliorate this is to augment the training data with paraphrases. Regardless of the magnitude of data augmentation, unseen instances may still break the model; data augmentation alone is an insufficient remedy for model brittleness.

We propose to go above and beyond data augmentation in handling model generalization for sequence-to-sequence (seq2seq) architectures and improve model generalization to test sets that entirely consist of unseen paraphrases of the training set. Assuming that data augmentation already took place in the training set, either by annotation or off-the-shelf paraphrase generation, we propose new models that actively employ the properties of paraphrase-augmented data as part of the training objective.

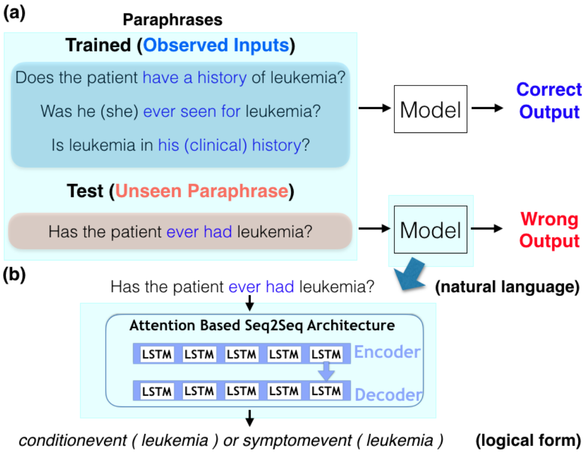
(a)
Paraphrases
Trained (Observed Inputs)
Does the patient have a history of leukemia?
Was he (she) ever seen for leukemia?
Is leukemia in his (clinical) history?
Model
Correct Output
Test (Unseen Paraphrase)
Has the patient ever had leukemia?
Model
Wrong Output
(b)
Has the patient ever had leukemia?
(natural language)
Attention Based Seq2Seq Architecture
LSTM
Encoder
Decoder
conditionevent (leukemia) or symptomevent (leukemia)
(logical form)

Figure 1: An overview of our work. (a) The objective is to train a seq2seq paraphrase model that is capable of accurately generalizing to unseen sentential paraphrases only observed at test time (red). Phrases highlighted in blue are synonymous when accompanied by a clinical condition, such as leukemia. (b) Example inputs and outputs for semantic parsing with emrQA.

We examine sequence models for diverse tasks across diverse domains, consider state of the art models for each of those tasks and incorporate multi-task paraphrase detection and generation learning. We show that our models compare over and above other popular generalization schemes, such as feature-based or fine-tuned word embeddings (Mikolov et al., 2013; Devlin et al., 2018) or paraphrase-based methods such as paraphrase embeddings (Wieting and Gimpel, 2017). The proposed models outperform state-of-the-art models (Pampari et al., 2018; Jia and Liang, 2016; Sennrich et al., 2017) when evaluated across a variety of settings on emrQA (Pampari et al., 2018)

[*] Work done while So Yeon Min was a student at MIT.

[1] AllenNLP's competitive BiDAF (Seo et al., 2017) reading comprehension model is not always capable at handling this: https://demo.allennlp.org/reading-comprehension/ODE5ODc2

Proceedings of the 3rd Clinical Natural Language Processing Workshop, pages 269–279
November 19, 2020. ©2020 Association for Computational Linguistics

and Overnight for semantic parsing and CzENG 1.6 (Bojar et al., 2016) with ParaNMT-50M (Wieting and Gimpel, 2018) for machine translation. Moreover, even when paraphrase augmentation is not available, we demonstrate that the proposed multi-task models improve model generalization with only synthetic, noisy paraphrases from off-the-shelf models.

The main contributions of our work are as follows: (1) We propose novel multi-task learning seq2seq models that significantly improve model generalization to unseen paraphrases at test time, in diverse domains (clinical text, 7 domains of Overnight, English subtitles). (2) Proposed models bring additional major performance boost on top of paraphrase-augmentation, but also work when the training set does not come with paraphrases at all. (3) We introduce new methods of splitting data into train/ test sets that more realistically evaluates model generalization to paraphrases. (4) We present the first competitive baseline for semantic parsing on the emrQA dataset.

2 Related Work

Dealing with unseen paraphrastic variants of the input has been a fundamental problem (Mitchell et al., 2018; Ettinger et al., 2017). Recently, multiple works have shown that models easily "break" when evaluated on adversarial examples, which are noisy variants of the training inputs (Goodfellow et al., 2014; Iyyer et al., 2018). However, there is relatively little work that go beyond augmentation and actively optimizes paraphrastic generalization along with learning the main NLP task at hand, in neural settings.

In non-neural settings, the idea that leveraging paraphrases facilitates modeling sentential semantics has been repeatedly verified across various NLP tasks. In semantic parsing, Berant and Liang (2014) deal with understanding the myriad paraphrastic variants in which knowledge base relations can be expressed in human language. They use a paraphrase of the original input utterance as an intermediary, which is used as an ancillary factor in ranking the likelihood of each candidate logical form. In machine translation, Callison-Burch et al. (2006) handles unseen source language phrases by substituting paraphrases of those phrases and then translating the paraphrases.

In neural settings, the most widespread approach is to simply generate paraphrases for data augmentation, as used by Fader et al. (2013a) in question answering and Wang et al. (2015) in semantic parsing. There are relatively few approaches that explicitly incorporate pairwise paraphrastic equivalence of inputs as part of the model. In semantic parsing, Dong et al. (2017) applies CNN to learn paraphrase detection in a multi-task manner; Su and Yan (2017) generate the simplest paraphrases for input utterances and use them as intermediaries for mapping input to ouput.

In question answering, several multi-task learning works learn paraphrase detection along with the main task; Bordes et al. (2014) optimizes a multi-task objective (negative cosine similarity) that encourages embeddings of paraphrases to have small angular distance in every other iteration of training. Additionally, Dong et al. (2015) uses an auxiliary multi-task learning objective for paraphrase detection in training multi-column convolutional neural networks for structured question answering. Both of these works leverage the paraphrase clusters of the WIKIANSWERS (Fader et al., 2013b) dataset. However, Dong et al. (2015) found that their multi-task learning method gives almost no advantage. Moreover, both works did not analyze which domains or types of validation inputs benefited from paraphrase learning. Most importantly, these works are fundamentally and methodologically different from ours, in that they leveraged the paraphrases from WIKIANSWERS not as inputs to the main model, but only for learning paraphrase detection. On the other hand, our work uses paraphrase instances for both multi-task paraphrase learning and the main task, which is the driving factor behind the significant performance boost by our models.

Unlike past methods applicable to single tasks, our work shows improvements across several different problems and domains. Also, we introduce the first framework in a neural-sequence-to-sequence setting, unlike past works that apply CNN's or non-neural settings. Furthermore, our approach can be generally applied to any state-of-the-art variants of seq2seq, such as Nematus (Sennrich et al., 2017).

3 Paraphrases & Datasets

Paraphrases are sentences or phrases that convey the same meaning using different wording (Bhagat and Hovy, 2013). Methods to construct paraphrases are largely divided into syntactic variation and substitution (Bhagat and Hovy, 2013). *"Does the patient have a history of leukemia?"* and *"Is*

Para. Types	**emrQA**
Syntactic Para's	what medication has the patient used for |problem| what medications have been previously used for the treatment of |problem|
Substitution Para's	is there any mention of |problem| in the patients record has been the patient ever been considered for |problem|
Para. Types t	**Overnight**
Syntactic Para's	find an additional author to an efron article who is the other author for the article written by efron
Substitution Para's	article that at least two article cites articles cited by two or more articles
Para. Types	**Paraphrase Augmented CzEng 1.6**
Syntactic Para's	It was good in spite of the taste Despite the flavor, it felt good
Substitution Para's	I took a stool sample from his heart. I took the stool sample after his lungs failed.

Table 1: Examples of annotated/ synthetically generated paraphrases in diverse domains. Syntactic variation paraphrases and synonymous substitution paraphrases are respective abbreviated as Syntactic Para's and Substitution Para's.

there leukemia in the patient's history?" are syntactic paraphrases, with overlapping words reordered. Most paraphrases are not fully syntactic, and involve substitutions with synonymous phrases by matching general semantics to that of a domain sublanguage.

Table 1 shows examples of annotated and synthetic paraphrases that are of syntactic variant/ synonymous substitution types. Some of emrQA and Overnight's paraphrases respectively assume knowledge of clinical ("considered for" ≡ "seen for, diagnosed with" when collocated with a |clinical problem|) and quantitative sublanguage ("at least two" ≡ "one or two"). CzEng 1.6 (Bojar et al., 2016), unlike the two others, do not come with annotated paraphrases. Thus, the paraphrases shown in Table 1 are those generated by ParaNMT-50M (Wieting and Gimpel, 2018). Noticeably, its synthetic paraphrases are quite noisy; "I took a stool sample from his heart" and "I took the stool sample after his lung failed" are not equivalent, but are identified so by the paraphrase generation model.

4 Problem Statement

Our setup assumes (1) a paraphrase-augmented dataset, either by annotation or simple off-the-shelf paraphrase generation models, and (2) a baseline seq2seq model (Sutskever et al., 2014) which maps an input sequence to an output sequence. We propose methods to endow additional improvement in model generalization given this setup.

Naive & Strict Splitting Schemes Not all paraphrases are created equal; some paraphrases are much less challenging than others in evaluating model performance. A common yet undesirable scenario in NLP datasets is that the form of input utterances can be repeated across training/ test splits. For example, in Overnight's *recipe* domain, there are several questions in the form of "*how many x are there*" where x is some recipe-related entity, such as "*recipes*", "*ingredients*", "*meals*", etc. With such datasets, the test set often contains too many repeating forms of the training set. Such a train/ test split is an unrealistic evaluation of model generalization. There are numerous ways a user can phrase one's information needs, and all possible forms cannot be seen at training even in the most well-augmented datasets.

We propose a new, more realistic way to split paraphrase-augmented data, with the emrQA dataset as an example (Figure 2). emrQA consists of paraphrase groups of inputs. Within a single group, "templates" are filled with clinical entities to produce actual input instances (Fig 2 purple box). 2(a) shows a naive splitting scheme where the input instances are split at random. On the other hand, 2(b) is a more realistic scenario where *a form that was seen during training **never** appears at test time*. For example, all instances of "Has the pat. ever been exposed to |problem|?" belong to training and never when the model is evaluated. On the other hand, all instances of "Does this patient have a history of |problem|?" never appear during training yet do so at test time; thus, the model is tested whether it can infer the meaning of this form only from its paraphrased forms seen during training (such as "Has the pat. ever been exposed to |problem|?"). While this split is more challenging than the naive one, test instances are still semantically equivalent to some training instance, so the model is expected to catch this and generalize to unseen form. While we used a paraphrase-annotated dataset as an example, 2(b)'s splitting scheme works with non-annotated data, via automatic augmentation with off-the-shelf models.

Thus, given a seq2seq task where we have a training and test set of input-output pairs and several unseen observations in the test set that are paraphrases of the training observations, we want to learn a model that can generalize accurately to these unseen observations, and preferrably, even to unseen forms in the strict split of Fig 2(b).

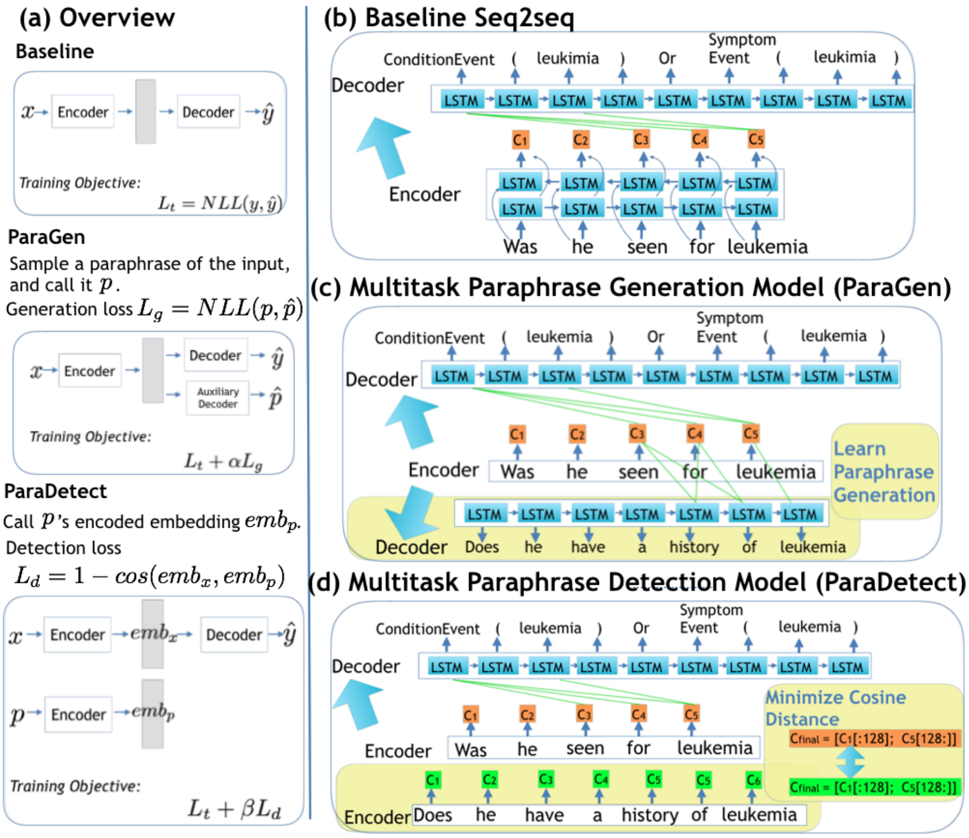
(a) Overview
Baseline
x → Encoder → Decoder → $\hat{y}$
Training Objective:
$L_t = NLL(y, \hat{y})$
ParaGen
Sample a paraphrase of the input, and call it p.
Generation loss $L_g = NLL(p, \hat{p})$
x → Encoder → Decoder → $\hat{y}$
Auxiliary Decoder → $\hat{p}$
Training Objective:
$L_t + \alpha L_g$
ParaDetect
Call p's encoded embedding emb_p.
Detection loss
$L_d = 1 - cos(emb_x, emb_p)$
x → Encoder → emb_x → Decoder → $\hat{y}$
p → Encoder → emb_p
Training Objective:
$L_t + \beta L_d$
(b) Baseline Seq2seq
Decoder
ConditionEvent (leukimia) Or Symptom Event (leukimia)
LSTM
C1 C2 C3 C4 C5
Encoder
Was he seen for leukemia
(c) Multitask Paraphrase Generation Model (ParaGen)
ConditionEvent (leukemia) Or Symptom Event (leukemia)
Decoder
Encoder
Was he seen for leukemia
Learn Paraphrase Generation
Decoder
Does he have a history of leukemia
(d) Multitask Paraphrase Detection Model (ParaDetect)
ConditionEvent (leukemia) Or Symptom Event (leukemia)
Decoder
Encoder
Was he seen for leukemia
Minimize Cosine Distance
Cfinal = [C1[:128]; C5[128:]]
C1 C2 C3 C4 C5 C5 C6
Cfinal = [C1[:128]; C5[128:]]
Encoder
Does he have a history of leukemia

Figure 2: Proposed paraphrase models. (a): Overview of all models; the encoder embeddings of inputs are depicted as gray boxes. (b): Simple Seq2seq (baseline) (c): Multitask Paraphrase Generation Model (d): Multitask Paraphrase Detection Model. Green lines represent attention weights, in (b), (c), (d). Detailed view of the multitask paraphrase generation and detection model is omitted for simplicity.

5 Methods: Seq2seq with Joint Paraphrase Learning

We incorporate auxiliary multi-task learning to the main seq2seq task - learning paraphrase generation (*ParaGen*), paraphrase detection (*ParaDetect*), and a combination of both tasks (*ParaGen* + *ParaDetect*). These methods work with any task whose inputs and outputs are sequences, on paraphrase-augmented data. The goal of our models is to inject paraphrastic inductive bias to the encoder hidden state - so that when it is passed to the decoder, paraphrase inputs have the same end results.

We achieve this by actively employing natural properties that rise from data augmentation as part of the training objective. First, *ParaGen, ParaDetect, ParaGen* + *ParaDetect* sample a paraphrase of the input and leverage it to reduce intra-class variance of paraphrases in the representation space. The sampled paraphrase is a term inside each model's respective multi-task objective, which affects the encoded input embedding in directions that reward paraphrastic homogeneity when back-propagated. However, paraphrase sampling is only required during training; at test time, the multi-task portion of the model is discarded, and the input is passed through the seq2seq model only. This is a realistic test scenario that does not require paraphrase identification among test inputs; the expectation is that the multi-task training has optimized the backbone seq2seq model's parameters for generalization at test time.

We introduce notations, with semantic parsing on emrQA as a running example (Fig 1). x is an input utterance (e.g., "Does the patient have a history of leukemia") and p is a paraphrase of it sampled from the training set (e.g., "Is leukemia in his clinical history?"). y is the desired output sequence (e.g. "ConditionEvent (Leukemia) or SymptomEvent (Leukemia)"); mapping from x to y is the *main task*, and L_t is the neagtive-log-likelihood (NLL) loss for this main task. $\hat{y}, \hat{p}$ are output sequence and paraphrase generated by the models. Finally, we note that we regard an attention-based (Luong et al., 2015) seq2seq with a bidirectional LSTM encoder and a LSTM decoder, with the dropout probability set to 0.1 (Srivastava et al., 2014), as the backbone baseline model (Figure 3b).

5.1 ParaGen: Multitask Paraphrase Generation Model

Given an input utterance x, we sample from the training set one of x's paraphrases, p, and learn paraphrase generation from x to p along with the main task. More specifically, from a shared encoder that accepts x as an input, we keep two separately parametrized decoders which respectively produce $\hat{y}$ (main task decoder) and $\hat{p}$ (paraphrase generation decoder) as desired outputs (Figure 3c). The resulting objective is a weighted sum of L_g, the loss for paraphrase generation, and L_t (main task objective), defined below:

$$L_{total} = L_t + \alpha L_g \qquad (1)$$

where L_g is the NLL loss between p and $\hat{p}$, and α is a hyperparameter for the weighted sum.

5.2 ParaDetect: Multitask Paraphrase Detection Model

In this model, we again sample a paraphrase p but learn as the auxiliary task, paraphrase detection - to identify whether x and p are paraphrases by looking at their embeddings emb_x and emb_p. We keep the same model structure as the baseline, but we pass p into the same encoder used for the input utterance x, to generate emb_p, a fixed-length vector representation of p. Then, we force emb_x and emb_p, vector representations of the two paraphrases, to have high cosine similarity - a criterion popularly used for paraphrase detection methods with input vector similarity (Mihalcea et al., 2006; Milajevs et al., 2014; Fernando and Stevenson, 2008). The resulting objective is a weighted sum of L_d, loss for paraphrase detection, and L_t, loss for the target task:

$$L_{total} = L_t + \beta L_d \qquad (2)$$

where

$$L_d = 1 - cos(emb_x, emb_p) = 1 - \frac{emb_x \cdot emb_p}{||emb_x||||emb_p||}$$

and β is a hyperparameter for the weighted sum.

5.3 Multitask Paraphrase Generation and Detection Model

We propose a combination of both models where we learn both paraphrase generation and detection as ancillary tasks. The resulting objective is a weighted sum of L_t, L_g, L_d:

$$L_{total} = L_t + \alpha(L_g + \beta L_d) \qquad (3)$$

where α, β are hyperparameters for the weighted sums.We hope to gain both advantages of ParaGen and ParaDetect by summing their objectives.

6 Experiments

We evaluate the proposed models over state-of-the-art methods and above existing methods of generalization, on two paraphrase-annotated datasets (emrQA, Overnight) and one that is automatically augmented with noisy paraphrases (CzEng 1.6).

6.1 Experiments with Paraphrase Annotated Datasets

We evaluate the proposed models on emrQA and Overnight, with the target task being semantic parsing - mapping English utterances to logical forms (structured representations that uniquely and exactly capture natural language meanings (Fig 1)). We train/test split emrQA with both "naive" and "realistic" (Section 4) schemes, and create four distinct splits for each scheme for fair model evaluation; Overnight has officially released train/test sets (unlike emrQA) so we use the official splits (that are "naive") for comparison with previous work.

Accuracy Metric

Our accuracy metric is "exact match" - which only considers model outputs that are identical to the labeled ones as correct. We mention this because "denotation accuracy" - which considers logical forms that return the label answer from the database as correct- has been used in several works on the Overnight dataset. However, there exist many instances in Overnight that can fool denotation accuracy.

In Overnight, there are many quantity-related questions; denotation accuracy can often consider model outputs of quantity-related questions right by chance (Table 2). For example, models often wrongly interpret "less than or equal to x" as "$< x$", but this could be considered correct if the database does not contain entries that are exactly x in time, amount, etc (Table 2). For example, in Table 2, given a question "venue of at most two article", the correct gold logical form should contain "$<=$". However, if the database does not contain any venue with exactly two articles, a false positivel logical form that contains "$<$" instead of "$<=$" can

Domain	Example Question	Gold Logical Form	False Positive Logical Form*
Publications	venue of at most two article	(listValue (countComparative (getProperty (singleton en.venue) (string !type)) (reverse (string venue))(string <=)(number 2) (getProperty (en.article) (string !type))))	(listValue (countComparative (getProperty (singleton en.venue) (string !type)) (reverse (string venue))(string <)(number 2) (getProperty (en.article) (string !type))))
Recipes	show me recipes that could be used for one or two meals	(listValue (countComparative (getProperty (singleton en.recipe) (string !type)) (string meal) (string <=) (number 2) (getProperty (en.meal) (string !type))))	(listValue (countComparative (getProperty (singleton en.recipe) (string !type)) (string meal) (string <=) (number 2) (getProperty (en.meal) (string !type))))

Table 2: Example quantitative questions in overnight prone to false positive logical forms under denotation match.

count correct under denotation accuracy. While hard to quantify, many questions in overnight are quantitative, including words such as "at most", "less than", "more than". [2]

Because of such a property, we consider exact match accuracy to be fairer than denotation match accuracy, and used it as our metric; the false positives pointed above cannot happen under exact match accuracy, since it only counts output exactly same as the gold logical form as correct. We also note that there is little worry of such an issue in emrQA, since questions rarely ask for quantitative information.

6.1.1 Methods for Comparison

To adequately judge the effect of joint paraphrase learning, we use seq2seq methods that have been established as State-of-the-Art for each dataset as the backbone baseline; proposed joint paraphrase learning is added on top of these backbones. **The same paraphrase augmented dataset is used for the baselines and our proposed models**, since the purpose is to evaluate the effectiveness of our proposed auxiliary objectives.

Seq2Seq SOTA's No previous work exists on semantic parsing for emrQA; thus, we establish the first competitive baseline with the copy mechanism (Gu et al., 2016) added on the backbone seq2seq described in Section 5, for copying of medical entities (e.g. *"leukemia"*). For Overnight, we implemented Jia and Liang (2016)'s model as baseline.

Paraphrase-based Generalization Methods Our primary goal is to show that active leveraging of paraphrase augmented data in the model gives additional benefits. Thus, we compare our proposed models with Seq2Seq baselines (defined above) on paraphrase-augmented datasets (each input instance in emrQA and Overnight is a paraphrase of some other instance in the dataset). This comparison proves the effectiveness of the proposed models beyond data augmentation.

We also compare our models with existing paraphrase-based generalization methods that can be used under seq2seq like Gated Average Recurrent Networks (GRAN) (Wieting and Gimpel, 2017) - a GRU with an additional averaging gate - that learn paraphrastic sentence embeddings.

The authors reported that pre-training with their method resulted in performance boost in transfer learning on SemEval tasks. To compare our methods with pre-training via GRAN, we replace the encoder of our baseline seq2seq with a GRAN encoder pre-trained on our tasks' training set, with the GRAN encoder's parameters not frozen.

We also compare with BERT (Devlin et al., 2018) (shown to be powerful in many NLP tasks) fine-tuned on paraphrase detection, which we framed as a sentence pair classification task into paraphrase/ non-paraphrase, applying the procedure in Devlin et al. (2018). For fine-tuning, we constructed the training set with all the paraphrase pairs in the original corpus and added non-paraphrase pairs sampled by the same number. Respectively for emrQA and Overnight, Clinical-BERT (Alsentzer et al., 2019) and 12-layer base BERT (English Wikipedia) were used as base; on both datasets, BERT was fine-tuned well enough to identify paraphrase with around 85% accuracy. For comparison, we took sentence embeddings from the fine-tuned BERT and replaced the encoder with it. We could not compare with end-to-end BERT models, because to our knowledge, no such prior work on semantic parsing exists.

Pre-trained Word Embeddings. Since pre-trained word embeddings are known to help generalization, the idea is to evaluate the contributions of the proposed paraphrase model over using standard methods to ensure generalization. We hypothesize two scenarios: (1) when pre-trained embeddings are available for large-scale corpus beyond train-

[2] More examples of quantity-related questions that can fool denotation accuracy can be found here: `https://github.com/ysu1989/CrossSemparse/blob/master/data/overnight/recipes/recipes.paraphrases.test.examples`

Method	**emrQA *“naive”* split** **(random split)**	**emrQA *“realistic”* split** **(unseen paraphrases only in test set)**
Baseline: Seq2seq with copy	85.24%	54.65%
Paraphrase Generation (ParaGen)	85.87%	61.97
Paraphrase Detection (ParaDetect)	85.37%	62.04%
ParaGen + ParaDetect	**86.55%**	**63.75%**

Table 3: Results on semantic parsing for the emrQA dataset, averaged across four splits.

Method / Domain	**Basketball**	**Blocks**	**Calendar**	**Publications**	**Recipes**	**Restaurants**	**Housing**	**SocialNetwork**
Baseline: Seq2seq with copy	82.8%	39.3%	**59.5%**	60.2%	75.0%	53.3%	47.1%	67.6%
Paraphrase Generation (ParaGen)	82.09%	40.9%	54.8%	59.6%	**75.5%**	**53.9%**	**49.2%**	**68.3%**
Paraphrase Detection (ParaDetect)	**83.8%**	**42.4%**	54.2%	60.9%	74.5%	51.5%	44.4%	**68.3%**
ParaGen + ParaDetect	82.6%	38.6%	56.5%	**63.4%**	70.4%	52.4%	45.5%	67.1%
Simple Seq2Seq (Damonte et al.)	69.6%	25.1%	43.5%	32.9%	58.3%	37.3%	29.6%	51.2%
Transfer Learning (Damonte et al.)	71.1%	25.1%	48.8%	40.4%	63.4%	39.2%	38.1%	54.5%

Table 4: Results on semantic parsing on all domains of the Overnight dataset.

Method	**SP emrQA**	**SP Overnight (*Publication*)**	**NMT Eng→Czech**
Baseline: Seq2seq with Copy*	54.65%	60.2 %	42.77
Baseline + Corpus Word2Vec	27.66%	57.1 %	N/A
Baseline + Large-Scale Word2Vec	**67.57%**	44.1%	42.23
BERT	52.48%	26.1%	N/A
GRAN	58.25%	58.6%	N/A
Paraphrase Generation (ParaGen)	61.97%	59.6%	**44.29**
ParaGen + Corpus Word2Vec	51.14%	60.25%	N/A
ParaGen + Large-scale Word2Vec	64.86%	39.8%	43.76
Paraphrase Detection (ParaDetect)	62.04%	60.9%	41.77
ParaDetect + Corpus Word2Vec	46.92%	57.8%	N/A
ParaDetect + Large-scale Word2Vec	63.02%	56.5%	43.90
Para(Gen+Detect)	**63.75%**	**63.4%**	40.72
Para(Gen+Detect) + Corpus Word2Vec	53.04%	60.2%	N/A
Para(Gen+Detect) + Large-scale Word2Vec	66.67%	51.55%	41.38

Table 5: Results on semantic parsing (SP) for the emrQA dataset (“realistic” split scheme, averaged over 4 splits), Overnight and neural machine translation (NMT) for EngCzech translation. Metrics are exact match and BLEU for respective the first two and the third column.*For neural machine translation, Nematus was used as baseline. Unseen Word acc. denotes the accuracy over validation inputs with tokens that never appeared during training.

ing data (2) when only corpus-trained embeddings are available. As large-scale embeddings, we use clinical word2Vec (Mikolov et al., 2013) trained on all i2b2 (Uzuner et al., 2011) datasets for emrQA, and officially released general English word2vec for Overnight.

6.1.2 Results

emrQA. For emrQA (Table 3), we can see that the proposed models outperform the baseline under both split schemes, but with a significant gap under the “realistic” split; this shows that our models are capable of robustly generalizing to unseen syntactic variants. We further compare our models with the different generalization methods mentioned (Table 5). ParaGen + ParaDetect is overwhelmingly dominant over other methods When large-scale corpus word embeddings are not available.

In emrQA, there were 338 test inputs with words that never appear during training (such as “considered” in 2nd example of emrQA’s Substitution paraphrase in Table 1). These inputs largely determined model performance, with overall accuracy being proportional to accuracy on these. Especially, ParaGen could not capture the topic of the question (e.g. medical evaluation, treatment, etc) when specific words were replaced with more general ones (e.g. “diagnosed for” → “considered for”). ParaDetect’s error usually occured in mistakenly copying entities.

Overnight Across 7 out of 8 domains of Overnight, the best performing model (ParaGen) outperformed baseline up to 3.2% (*Publications*) with 1.6% boost on average (Table 4). We also report results from Damonte et al. (2019), which is an existing work on Overnight with exact match accuracy. With our implementation of Jia and Liang (2016), we achieved a baseline higher than both of the baseline and proposed methods of Damonte et al. (2019). Results across all 8 domains is in Table 4. We further compare our models with different generalization methods. Word2vec was not effective, as in Su and Yan (2017), and pre-training with BERT and GRAN were less effective than Para(Gen+Detect).

Discussion Proposed models produced pronounced improvements on emrQA where paraphrases express the multiple ways a physician may phrase information needs. “Has the patient ever been considered for |problem|”, “any |problem|

history” involve matching general semantics to that of clinical sub-language. But “considered for” has a broad meaning in the general domain; it is synonymous to “seen for, diagnosed with”, collocated with a |clinical problem|. Overnight’s paraphrases are open-domain (“which recipes require milk”, “which recipes need milk”) or require quantitative knowledge (“person that is author of at most two articles”, “author of one or two articles” require knowing that there is only “one” between zero and two). This is different from identifying a sub-language meaning of a general phrase.

While joint paraphrase learning cannot learn actual knowledge of a domain (such as quantitative knowledge), it is useful in identifying meanings of general-sense phrases in a specific domain sub-language, that is much needed in clinical settings.

6.2 Experiments with Automated Noisy Augmentation

We noisily augmented a subset of CzEng 1.6 with ParaNMT-50M (Wieting and Gimpel, 2018), for training and evaluation on machine translation from English to Czech; because the authors released paraphrases of the English instances of CzEng 1.6 generated from it, we directly used them. We chose the *subtitles* domain from CzEng 1.6 to cover the most open-domain language, which was relatively less covered in the other two datasets (because Overnight has specific domain-specific questions for each domain); we randomly chose 33.33 thousand utterances from CzEng 1.6 *subtitles* and augmented each utterance with 3 more paraphrases. Among the four paraphrases in each group, we randomly assigned one to the test split and the rest to training; the training set consists of roughly 0.1M instances.

We use a state-of-the-art seq2seq model, Nematus (Sennrich et al., 2017), the winning solution for WMT 2017 (one of whose training corpora was CzEng 1.6), as the backbone baseline. The four models were evaluated with and without initialization with general English Word2vec of encoder and decoder parameters. We report the BLEU of all four models in Table 5; we achieved improvement in BLEU by 1.5 in comparison to the baseline, with ParaGen. We further note that the use of Word2vec actually harmed performance.

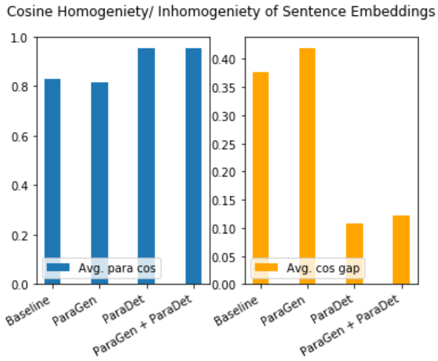

Figure 3: Results on cosine similarity between test question pairs in emrQA. Blue: homogeneity of paraphrases; orange: nonhomogeneity of non-paraphrases.

7 Discussion: Cosine Distance Analysis (emrQA)

To understand the contribution of the proposed paraphrase models, we study the cosine similarity between embeddings of sentence pairs, a general metric in textual similarity and paraphrase detection (Agirre et al., 2016; Fern and Stevenson) in Figure 3. We do this by calculating the similarity between the last hidden state of the encoder for a pair of input utterances in the test set, for the four splits of emrQA’s “realistic” splitting scheme. We calculate two metrics: (1) the average cosine similarity between pairs of paraphrase utterances (*Avg. para cos*, blue bar in Fig 4) and (2) the average difference between cosine similarity of paraphrase pairs and that of non-paraphrase pairs (*Avg. cos gap*, orange bar in Fig 4). They respectively quantify (1) how *homogenously* paraphrase utterances are embedded as vectors, and (2) how *nonhomogeneously* non-paraphrase utterances are embedded; *high* numbers in both quantities are ideal, if our models behave as intended in the methods section. We observed that ParaDetect achieves noticeably the highest *Avg. para cos*, and ParaGen the highest *Avg. cos gap*; ParaGen + ParaDetect shows something in between the two but closer to ParaDetect. These cosine statistics of embeddings seem to be indicative of model performance. ParaGen + ParaDetect, which embeds both paraphrases homogenously and non-paraphrases nonhomogenously, performs the best in terms of exact match accuracy; the other two models also achieve higher performance than baseline, with much higher *Avg. para cos* and *Avg. cos gap* than baseline.

8 Conclusion

We presented a new general seq2seq framework where the main task is trained together with a paraphrase-learning objective to enhance model generalization. We also introduced new splitting schemes that reflect realistic evaluation for practical use. Our proposed approaches outperform the state-of-the-art across three datasets across diverse domains and tasks.

References

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2016. Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 497–511, San Diego, California.

Emily Alsentzer, John R. Murphy, Willie Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew B. A. McDermott. 2019. Publicly available clinical BERT embeddings. *CoRR*, abs/1904.03323.

Yonatan Belinkov and Yonatan Bisk. 2017. Synthetic and natural noise both break neural machine translation. *arXiv preprint arXiv:1711.02173*.

Jonathan Berant and Percy Liang. 2014. Semantic parsing via paraphrasing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1415–1425, Baltimore, Maryland. Association for Computational Linguistics.

Rahul Bhagat, Eduard Hovy, and Siddharth Patwardhan. 2009. Acquiring paraphrases from text corpora. In *Proceedings of the Fifth International Conference on Knowledge Capture*, K-CAP '09, pages 161–168, New York, NY, USA. ACM.

Rahul Bhagat and Eduard H. Hovy. 2013. What is a paraphrase? *Computational Linguistics*, 39:463–472.

Ondřej Bojar, Ondřej Dušek, Tom Kocmi, Jindřich Libovický, Michal Novák, Martin Popel, Roman Sudarikov, and Dušan Variš. 2016. CzEng 1.6: Enlarged Czech-English Parallel Corpus with Processing Tools Dockered. In *Text, Speech, and Dialogue: 19th International Conference, TSD 2016*, number 9924 in Lecture Notes in Computer Science, pages 231–238, Cham / Heidelberg / New York / Dordrecht / London. Masaryk University, Springer International Publishing.

Antoine Bordes, Jason Weston, and Nicolas Usunier. 2014. Open question answering with weakly supervised embedding models. In *Proceedings of the 2014th European Conference on Machine Learning and Knowledge Discovery in Databases - Volume Part I*, ECMLPKDD'14, pages 165–180, Berlin, Heidelberg. Springer-Verlag.

Chris Callison-Burch, Philipp Koehn, and Miles Osborne. 2006. Improved statistical machine translation using paraphrases. In *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, HLT-NAACL '06, pages 17–24, Stroudsburg, PA, USA. Association for Computational Linguistics.

Marco Damonte, Rahul Goel, and Tagyoung Chung. 2019. Practical semantic parsing for spoken language understanding. *CoRR*, abs/1903.04521.

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

Li Dong, Jonathan Mallinson, Siva Reddy, and Mirella Lapata. 2017. Learning to paraphrase for question answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 875–886, Copenhagen, Denmark. Association for Computational Linguistics.

Li Dong, Furu Wei, Ming Zhou, and Ke Xu. 2015. Question answering over Freebase with multi-column convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 260–269, Beijing, China. Association for Computational Linguistics.

Allyson Ettinger, Rao Sudha, Daumé Hal, and M. Bender Emily. 2017. Towards linguistically generalizable nlp systems: A workshop and shared task. *ArXiv*, abs/1711.01505.

Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. 2013a. Paraphrase-driven learning for open question answering. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1608–1618, Sofia, Bulgaria. Association for Computational Linguistics.

Anthony Fader, Luke S. Zettlemoyer, and Oren Etzioni. 2013b. Paraphrase-driven learning for open question answering. In *ACL*.

Samuel Fern and Mark Stevenson. A semantic similarity approach to paraphrase detection.

Samuel Fernando and Mark Stevenson. 2008. A semantic similarity approach to paraphrase detection. In *Proceedings of the 11th Annual Research Colloquium of the UK Special Interest Group for Computational Linguistics*, pages 45–52.

Ian.J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *CoRR*, abs/1412.6572.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1640, Berlin, Germany. Association for Computational Linguistics.

Mohit Iyyer, , John Wieting, Kevin Gimpel, and Luke S. Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. In *NAACL-HLT*.

Robin Jia and Percy Liang. 2016. Data recombination for neural semantic parsing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12–22, Berlin, Germany. Association for Computational Linguistics.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. *arXiv preprint arXiv:1707.07328*.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Rada Mihalcea, Courtney Corley, and Carlo Strapparava. 2006. Corpus-based and knowledge-based measures of text semantic similarity. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 1*, AAAI'06, pages 775–780. AAAI Press.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Dmitrijs Milajevs, Dimitri Kartsaklis, Mehrnoosh Sadrzadeh, and Matthew Purver. 2014. Evaluating neural word representations in tensor-based compositional settings. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 708–719, Doha, Qatar. Association for Computational Linguistics.

Jeff Mitchell, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Extrapolation in nlp. *arXiv preprint arXiv:1805.06648*.

Anusri Pampari, Preethi Raghavan, Jennifer Liang, and Jian Peng. 2018. emrQA: A large corpus for question answering on electronic medical records. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2357–2368, Brussels, Belgium. Association for Computational Linguistics.

Rico Sennrich, Orhan Firat, Kyunghyun Cho, Alexandra Birch, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel Läubli, Antonio Valerio Miceli Barone, Jozef Mokry, and Maria Nadejde. 2017. Nematus: a toolkit for neural machine translation. In *EACL*.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hananneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. In *International Conference on Learning Representations*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Yu Su and Xifeng Yan. 2017. Cross-domain semantic parsing via paraphrasing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1235–1246, Copenhagen, Denmark. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc.

Özlem Uzuner, Brett R South, Shuying Shen, and Scott L DuVall. 2011. 2010 i2b2/va challenge on concepts, assertions, and relations in clinical text. *Journal of the American Medical Informatics Association*, 18(5):552–556.

Yushi Wang, Jonathan Berant, and Percy Liang. 2015. Building a semantic parser overnight. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1332–1342, Beijing, China. Association for Computational Linguistics.

John Wieting and Kevin Gimpel. 2017. Revisiting recurrent networks for paraphrastic sentence embeddings. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

John Wieting and Kevin Gimpel. 2018. ParaNMT-50M: Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 451–462, Melbourne, Australia. Association for Computational Linguistics.

Appendices

A More Details on Types of Paraphrases

We explain on the syntactic variation and synonymous substitution types of paraphrases in more detail. While syntactic variant paraphrases can be similarly identified (by switch of active/ passive tenses or ordering of clauses in Table 1), synonymous substitution paraphrases show different fashion of assumed knowledge across domains/ datasets. emrQA(Pampari et al.)'s paraphrases represent the multiple ways a physician may phrase their information needs; the substitution paraphrases acknowledge the clinical sublanguage of equating "considered for" with "seen for, diagnosed with" when collocated with a $|$clinical problem$|$. In overnight (Wang et al.), many substitution paraphrases are regarding quantitative knowledge., while hard to exactly quantify the proportion. While some are easier to identify ("more than two" $\equiv$ "greater than two"), others involve some numerical knowledge that models trained on non-numerical benchmark corpora may lack ("at least two" $\equiv$ "one or two") - for example, that there is only "one" between "zero" and "two".

B Implementation and Training Details

B.1 Fine-tuning BERT for Paraphrase Detection

We chose learning rate among $\{2e-5, 3e-5, 5e-5\}$, and trained for 5 epochs, stopping early at the highest validation accuracy.

B.2 Hyperparameter Selection

Hyperparameters consist of learning rate and α, β from Section 5. They were grid-searched iteratively; first, learning rate for the baseline model was grid-searched, and then α, β for each of the proposed models were grid-searched, with the learning rate fixed to what was found for the baseline. Finally, each of the proposed models' learning rates were grid-searched, with α, β fixed. emrQA's hyperparmeters were selected among $\alpha \in \{1, 0.1, 0.01\}, \beta \in \{1.25, 1, 0.75, 0.5\}$, learning rate $\in \{5e-4, 1e-3, 1.5e-3\}$; Overnight's hyperparameters among $\alpha \in \{1, 0.1, 0.01\}, \beta \in \{1.25, 1, 0.75, 0.5\}$, learning rate $\in \{1e-4, 3e-4, 5e-4\}$; Finally, CzEng 1.6's were among $\alpha \in \{1, 0.1, 0.01\}, \beta \in \{1.25, 1, 0.75, 0.5\}$, learning rate $\in \{1e-4, 3e-4, 5e-4, 7.5e-4\}$.

We also note that for each of emrQA, Overnight, and CzEng 1.6, models were trained up to 20, 50, and 100 epochs with early stopping at the epoch that returns best validation accuracy.

B.3 Implementation Details

All code was implemented with PyTorch. Average runtime of the experiments for was 1.5$\sim$2 hours per run for emrQA, and $\sim$30 minutes for Overnight.