Iterative Paraphrastic Augmentation with Discriminative Span Alignment

Ryan Culkin J. Edward Hu Elias Stengel-Eskin
Guanghui Qin Benjamin Van Durme

Johns Hopkins University
{rculkin, edward.hu, elias, gqin, vandurme}@jhu.edu

Abstract

We introduce a novel paraphrastic augmentation strategy based on sentence-level lexically constrained paraphrasing and discriminative span alignment. Our approach allows for the large-scale expansion of existing datasets or the rapid creation of new datasets using a small, manually produced seed corpus. We demonstrate our approach with experiments on the Berkeley FrameNet Project, a large-scale language understanding effort spanning more than two decades of human labor. With four days of training data collection for a span alignment model and one day of parallel compute, we automatically generate and release to the community 495,300 unique $(\texttt{Frame}, \texttt{Trigger})$ pairs in diverse sentential contexts, a roughly 50-fold expansion atop FrameNet v1.7. The resulting dataset is intrinsically and extrinsically evaluated in detail, showing positive results on a downstream task.

1 Introduction

Data augmentation is the process of automatically increasing the size or diversity of a dataset with the goal of improving performance on a task of interest. It has been applied in many areas of machine learning including computer vision (Shorten and Khoshgoftaar, 2019) and speech recognition (Ragni et al., 2014; Ko et al., 2015).

With text-based datasets in particular, *paraphrastic augmentation*, a technique to automatically expand datasets in their overall size and lexico-syntactic diversity via the use of a paraphrase model, may be applied. In general, a paraphrase model outputs a sentence $\mathbf{S}'$ given an input sentence $\mathbf{S}$ such that $\texttt{meaning}(\mathbf{S}) \approx \texttt{meaning}(\mathbf{S}')$ and $\mathbf{S} \neq \mathbf{S}'$. Prior work has demonstrated that paraphrastically augmented datasets are beneficial when applied to a variety of sentence-level tasks including machine translation, natural language inference, and intent classification (Ribeiro et al., 2018; Hu et al., 2019a; Kumar et al., 2019).

Often in paraphrastic augmentation an input sentence is rewritten one or more times, with the assumption the transformed sentence(s) preserve the original label. For example, in sentiment analysis, data consists of $(\texttt{Sentence}_i, \texttt{Label}_i)$ pairs, where each $\texttt{Label}_i$ is in $\{0, 1\}$, indicating negative or positive sentiment. To augment this kind of dataset, we can paraphrase each $\texttt{Sentence}_i$ with a model f and thereby produce an additional $(f(\texttt{Sentence}_i), \texttt{Label}_i)$ pair, doubling the size of the dataset.

In many natural language understanding tasks, however, data contains *span* labels of the form: $(\texttt{Sentence}_i, \{(\texttt{start}_{i,1}, \texttt{end}_{i,1}, \texttt{type}_{i,1}), ...\})$, where the latter element is a set of tuples indicating each label's location (as a contiguous subsequence of the input tokens) and type. In this paper, we develop a data augmentation strategy for span labeling problems where we are concerned with balancing the joint objectives of finding different ways to express meaning at the level of a word or phrase while ensuring the paraphrase is sensitive to the context of the surrounding sentence.

Although a paraphrase is expected to have the same meaning as the sentence from which it was generated, words and phrases are usually added, removed, or reordered. Thus for a given sentence annotated with span labels, while we expect the same label *types* to still apply to a paraphrase, the *locations* (start and end) are expected to shift.

To address this issue, we introduce a new model for span-based discriminative alignment. Given an input sentence $\mathbf{S}$, a paraphrase $f(\mathbf{S})$, and a span of tokens in $\mathbf{S}$ representing a label location, the alignment model finds a semantically equivalent span in $f(\mathbf{S})$. We present the architectural details of this model, a dataset for span alignment, and corresponding results in §4.

A second problem is that most paraphrase models offer no control over specific words or phrases

Transactions of the Association for Computational Linguistics, vol. 9, pp. 494–509, 2021. https://doi.org/10.1162/tacl_a_00380
Action Editor: Chris Quirk. Submission batch: 10/2020; Revision batch: 12/2020; Published 4/2021.

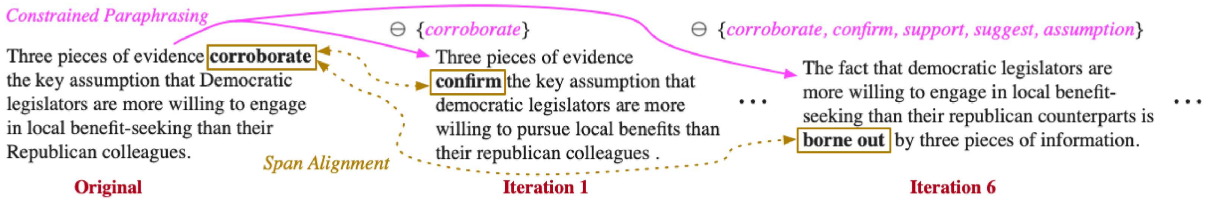

Constrained Paraphrasing
⊖ {corroborate}
⊖ {corroborate, confirm, support, suggest, assumption}
Three pieces of evidence corroborate the key assumption that Democratic legislators are more willing to engage in local benefit-seeking than their Republican colleagues.
Three pieces of evidence confirm the key assumption that democratic legislators are more willing to pursue local benefits than their republican colleagues .
The fact that democratic legislators are more willing to engage in local benefit-seeking than their republican counterparts is borne out by three pieces of information.
Span Alignment
Original
Iteration 1
Iteration 6

Figure 1: Framework for iterative paraphrastic augmentation illustrated on an actual system output. The original, manually annotated sentence contains a tag over the word ‘‘corroborate’’. In Iteration 1, the sentence is paraphrased using a lexically constrained decoder with negative constraints on ‘‘corroborate’’ and all associated inflectional forms, guaranteeing that it will not appear in the paraphrase. Next, a span alignment model is used to obtain a link between ‘‘corroborate’’ in the original sentence and ‘‘confirm’’ in the paraphrase. All inflectional forms of ‘‘confirm’’ are then unioned with the prior set of negative constraints and the process repeats for a predetermined number of iterations.

to be included in or excluded from the final output. As text-based data augmentation typically aims to increase lexical diversity, it is useful to force span label text to be rewritten in the paraphrase as a synonymous or semantically similar phrase via *lexically constrained decoding* (§3).

In §5 we describe an augmentation framework that utilizes lexically constrained paraphrasing and alignment together, iteratively, to expand datasets for span labeling problems. An illustrative diagram is given in Figure 1.

Finally, we demonstrate the application of this framework to FrameNet in §6, resulting in a new dataset with 495,300 unique (`Frame`, `Trigger`) pairs in diverse sentential contexts. The intrinsic quality of the dataset is evaluated manually and its utility on external tasks is demonstrated with positive results on the task of Frame ID.

2 Background

Monolingual Paraphrasing Coinciding with the improvement of machine translation, several works have explored sentential paraphrasing through back-translation (Mallinson et al., 2017; Wieting and Gimpel, 2018). One such model (Wieting and Gimpel, 2018) was used for sentence canonicalization, although its further usefulness was hindered by lack of control over the paraphrasing process. Hu et al. (2019b) introduced constrained decoding (Post and Vilar, 2018) to sentential paraphrasing, enabling lexical control over the paraphrases. Wang et al. (2018) incorporated semantic frames and roles into Transformers to produce better paraphrases. Our work can be seen as taking their work in the opposite direction. While they used semantic information to inform paraphrases, we leverage high-quality paraphrases to generate new lexical units in semantic frames.

Automatic Lexicon Expansion As an alternative to manual labor, past work has sought to automatically build on existing semantic resources. Snow et al. (2006) used hypernym predictions and coordinate term classifiers to add 10,000 new WordNet entries with high precision. FrameNet+ (Pavlick et al., 2015) tripled the size of FrameNet by substituting words from PPDB (Ganitkevitch et al., 2013), a collection of primarily word-level paraphrases obtained via bilingual pivoting. PPDB paraphrases lack sentential context; for example, ‘‘river bank’’, ‘‘bank account’’, and ‘‘data bank’’ are listed as paraphrases of ‘‘bank’’, in addition to the broader and incorrectly cased ‘‘organizations’’ and less related still, ‘‘administrators’’,[1] without any means of determining when one might not be a valid substitute.[2] While the FrameNet+ expansion itself involved little cost, the lexicalized nature of their procedure failed to capture word senses in context and resulted in many false positives, requiring costly manual evaluation of every sentence. In contrast, we seek to mitigate false positives and enhance lexical and syntactic diversity by using a context-aware paraphrase model.

Paraphrasing for Structured Prediction Structured prediction finds a mapping between a surface form and some aspect of its underlying structure.

[1] `http://paraphrase.org/#/search?q=bank&filter=%5BNN%5D,%5BNNP%5D,%5BNP%5D&lang=en.`

[2] Even if we could determine contextually synonymous words for a (sentence, word) pair, they may not be grammatically or semantically valid when substituted back into the sentence, further motivating sentence-level paraphrasing.

that express the same meaning (i.e., paraphrases) which makes learning this mapping nontrivial.

Berant and Liang (2014) leveraged unstructured Q&A data by learning a paraphrasing model that maps a new query to existing ones with known structures. More relevant to our work, Wang et al. (2015) built a semantic parser from a small seed lexicon by generating canonical utterances from a domain-general grammar and then manually collecting paraphrases of these utterances through crowd-sourcing. A semantic parser is then trained on the paraphrases to produce the underlying structures that generated them. Our work is distinct in that we *automatically* expand our seed lexicon, collecting human judgments on a small subset of outputs in order to assess quality. Moreover, we introduce a general framework for augmenting data for span labeling, whereas Wang et al. (2015) focused on parsing. Choe and McClosky (2015) improved parsing performance by jointly parsing a sentence and its paraphrases. In addition, they constructed the paraphrases manually and discouraged syntactic diversity, as it lowered parsing performance.

Monolingual Span Alignment Yao et al. (2013a) introduced a discriminatively trained CRF model for monolingual word alignment, expanded to span alignment by Yao et al. (2013b). Ouyang and McKeown (2019) introduced a pointer-network-based phrase-level aligner for paraphrase alignment that obtains high recall on several tasks. Syntactic chunking is used to build a candidate set of phrases in both source and paraphrase sequences, which the model is then tasked with aligning. Their model is applied to an open alignment task, where more than one phrase in the source and paraphrase should be aligned, differing from the setting described in §4.

While we have chosen to make use of span-pooled BERT representations in our alignment model, a natural direction for future work would be to use span-based representations such as SpanBERT (Joshi et al., 2020).

The Berkeley FrameNet Project FrameNet (Baker et al., 2007) is the application of frame-semantic theory (Fillmore, 1982) to real-world data. Each FrameNet *frame* contains a description of a concept, a list of entities participating in the frame (*frame elements*), and a list of *lexical units*, which are the semantically similar words that evoke, or *trigger*, the given concept. Figure 2 illustrates a sentence labeled under the FrameNet protocol. FrameNet v1.7 contains roughly 1,200 frames, 8,500 annotated lexical units, and 200,000 annotations over English text taken from newspapers, journals, popular fiction, and other sources.

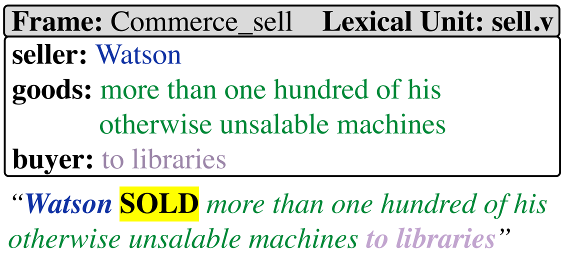

Frame: Commerce_sell **Lexical Unit: sell.v**
seller: Watson
goods: more than one hundred of his otherwise unsalable machines
buyer: to libraries
"*Watson* **SOLD** *more than one hundred of his otherwise unsalable machines to libraries*"

Figure 2: An example annotation from FrameNet. The trigger, ''sold'', an instance of the `sell.v` lexical unit, evokes the `Commerce_sell` frame. The participating entities, or *frame elements*, are represented as colored text.

FrameNet has been used in tasks ranging from question-answering (Shen and Lapata, 2007) and information extraction (Ruppenhofer and Rehbein, 2012) to semantic role labeling (Gildea and Jurafsky, 2002) and recognizing textual entailment (Burchardt and Frank, 2006), in addition to finding utility as a lexicographic compendium. As a manually created resource, FrameNet is limited by the size of its lexical inventory and number of annotations (Shen and Lapata, 2007; Pavlick et al., 2015).

3 Lexically Constrained Paraphrasing

Sentential paraphrasing is a sequence generation problem where the goal is to find an output sequence conveying similar semantics to the input sequence while also ensuring that the two sequences are lexically or syntactically distinct. Recent prior work has approached this problem with sequence-to-sequence neural networks (Wieting and Gimpel, 2018; Hu et al., 2019a), where an encoder embeds the input sequence into a fixed-dimensional space and a decoder produces a sequence auto-regressively. Often, the decoder uses beam search to explore the output space more efficiently.

Lexically constrained decoding allows one to dynamically include or exclude token sequences from the output via user-supplied positive or negative constraints. When combined with paraphrasing, it can boost external NLP task performance via data augmentation (Hu et al., 2019a). Our work uses negative constraints, which exclude certain token sequences from the output by setting the likelihood of the last token in the negative constraint phrase to zero when all preceding tokens in the phrase have been generated (Hu et al., 2019a).

In our experiments, we follow the model architecture[3] described by Hu et al. (2019a) with minor changes: 1) we use SentencePiece (Kudo and Richardson, 2018) unigrams instead of tokenization, following Hu et al. (2019c); 2) we do not use source factors, as SentencePiece unigrams are case-sensitive. These changes allow us to rewrite raw text without tokenization. The model is trained to convergence on a corpus (Hu et al., 2019c) with rich lexical and syntactic diversity, as measured by human judgment and parse-tree edit-distance, respectively.

4 Alignment Models

4.1 BERT-based Span Alignment Model

We present a model based on BERT (Devlin et al., 2018) to align spans of text between paraphrastic sentence pairs. The model is trained and evaluated on a new dataset[4] released alongside this paper, consisting of 36,417 labeled sentence pairs.

Architecture Our model takes as input two tokenized English-language sentences $\mathbf{S}$ (*source*, with n tokens) and $\mathbf{S}'$ (*reference*, with m tokens), where $\mathbf{S}'$ is a paraphrase of $\mathbf{S}$. The model also takes as input a span $\mathbf{s}$ in $\mathbf{S}$: a contiguous subsequence of tokens with length between 1 and n, initially represented as a tuple of $(\texttt{start}, \texttt{end})$ offsets into the source-side token sequence. Given this input the model predicts a span $\hat{\mathbf{s}} \in \{(i,j) | 1 \leq i \leq j \leq m\}$, representing the best alignment between $\mathbf{s}$ and the $O(n^2)$ possible candidate spans[5] in $\mathbf{S}'$.

In the forward pass, we embed $\mathbf{S}$ and $\mathbf{S}'$ using a pretrained 12-layer BERT-Base model with frozen parameters, obtaining a hidden vector $t_i \in \mathbb{R}^{768}$ for each of the $(m + n + 3)$ input tokens. $\mathbf{S}$ and $\mathbf{S}'$ are embedded at the same time, that is, as `[CLS]` $\mathbf{S}$ `[SEP]` $\mathbf{S}'$ `[SEP]`, following the Microsoft Research Paraphrase Corpus (Dolan and Brockett, 2005) paraphrase classification experiments of Devlin et al. (2018).

Next, we obtain a fixed-size representation $\mathcal{S} \in \mathbb{R}^{768}$ of the source-side span by mean-pooling the corresponding hidden states. In the same way, we compute span representations $\mathcal{C}_i$ for each of the $O(n)$ reference-side candidate answer spans whose length[6] is within k of the length of the source-side span $\mathbf{s}$. For each span pair representation $(\mathcal{S}, \mathcal{C}_i)$ we create an aggregate $\mathcal{V}_i \in \mathbb{R}^{1540}$ by concatenating three vectors:

- Element-wise difference (Df): $\mathcal{S} - \mathcal{C}_i$
- Element-wise maxima (Mx): $\texttt{max}(\mathcal{S}, \mathcal{C}_i)$
- Positional cues (Cue): start index and length per span[7]

We expect that the element-wise difference of the two span representations is close to the zero vector when the spans are close in meaning; a useful signal for the model. Concatenating element-wise maxima to the representation was beneficial empirically. Since word spans in the source likely start in a similar position and are of a similar length as compared to corresponding word spans in the reference, positional cues provide useful information. Finally, the aggregate vector $\mathcal{V}_i$ is fed into a simple feedforward neural network f, consisting of one layer with 770 hidden units, PReLU activations, batchnorm, and a sigmoid output layer.

We use binary cross entropy loss with *soft* labels: Rather than labeling each $\mathcal{C}_i$ candidate span as 1 or 0 depending on whether it is the gold-standard span, we assign labels according to the function $2^{-d(\mathcal{S}, \mathcal{C}_i)}$, where d measures the absolute difference of the start and end offsets between two spans: $d(a, b) = |a_1 - b_1| + |a_2 - b_2|$. In this way, the gold span is given a label of 1, candidate spans that are close to the gold-standard span are given partial credit, and partial credit exponentially approaches 0 as the distance between the candidate span and gold-standard span increases. This labeling strategy has two motivations. First, since only one of the $O(n)$ candidates is correct, there are many more negative examples than positive

[3] Transformer with 6-layer encoder, 4-layer decoder, 8 heads, 512-d embeddings, and feed-forward size of 2048.

[4] `http://nlp.jhu.edu/parabank.`

[5] The model only explicitly scores the $O(n)$ reference spans whose length is within k of the source-side span. Remaining spans are implicitly assigned zero probability.

[6] In our experiments we used $k = 5$; this was the lowest value that guaranteed the gold-standard reference span would be considered as a possible candidate 100% of the time in the training set.

[7] This vector contains four elements: the start index and length corresponding to the $\mathcal{S}$ representation, and the start index and length corresponding to the $\mathcal{C}_i$ representation.

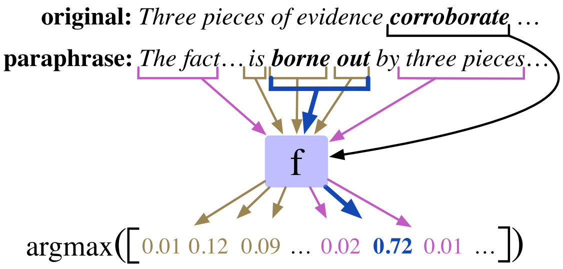

original: Three pieces of evidence corroborate …
paraphrase: The fact… is borne out by three pieces…
f
argmax([0.01 0.12 0.09 … 0.02 0.72 0.01 …])

Figure 3: Span alignment inference. A BERT-based representation of the source-side span ''corroborate'' is passed to a neural network *f*, scoring against possible reference-side candidate spans.

ones; thus, this strategy decreases the label imbalance. Second, we believe that tokens close to the gold span are more likely to be semantically similar to the gold span than far away tokens on average, so this strategy avoids harshly penalizing the model when it predicts a nearby (and likely semantically similar) span.

At inference time, we choose the span corresponding to the aggregate representation $\mathcal{V}_i$ that is assigned the highest score by the neural network f (i.e., $\hat{\mathbf{s}} = \arg\max_i f(\mathcal{V}_i)$). A diagram illustrating the inference procedure is given in Figure 3.

Data To train and evaluate our model we crowdsourced a span alignment dataset consisting of 36,417 labeled sentence pairs. Each instance in the dataset consists of a natural language sentence taken from FrameNet, a span in the sentence corresponding to a FrameNet trigger span, an automatic paraphrase, and a span in the automatic paraphrase that has been manually aligned with the source-side span. In our experiments, we split the data randomly as 80% train, 10% dev, and 10% test.

Automatic paraphrases of FrameNet sentences were generated using the model described in §3, where a negative constraint was placed on the source-side span text and its morphological variants in order to force the model to replace the original trigger with a semantic equivalent. Each paraphrase was decoded using top-k sampling with $k = 10$. In order to ensure broad lexical coverage we paraphrased up to[8] four sentences for each of the roughly 10k lexical units in FrameNet.

Annotators were presented with a highlighted trigger span from a FrameNet sentence and asked to identify an analogous span in the automatic paraphrase. The annotation interface allowed workers to state that the paraphrase did not contain any semantically equivalent phrase, which occurred 9% of the time. In a 1260-sentence study of span labeling inter-annotator agreement with 3-way redundancy, of the cases where the three annotators did select a span, they chose the same span (exact match) 88% of the time.

4.2 Word-level Baselines

We compare our span alignment model with two word-level alignment baselines: FastAlign (Dyer et al., 2013) and DiscAlign (Stengel-Eskin et al., 2019). The former is a fast implementation of IBM Model 2 (Brown et al., 1993), which decomposes the conditional probability of a target sequence given a source sequence into a lexical model and an alignment model. FastAlign is an asymmetric model, meaning that it must be run in both directions (source to paraphrase and paraphrase to source) and then these alignments must be combined using some heuristic—we use the *grow-diag-final-and* heuristic. A FastAlign model was run over the concatenation of the test data, the train data, and paraphrased FrameNet data to obtain the final test alignments.

DiscAlign is a discriminatively trained neural alignment model that uses the matrix product of contextualized encodings of the source and paraphrase word sequences to directly model the probability of an alignment given the source and paraphrase sequences. Unlike FastAlign, which is trained on bitext alone, DiscAlign is pre-trained on bitext and fine-tuned on gold-standard alignments. For this task, a DiscAlign model was pre-trained with 141 million sentences of ParaBank data (Hu et al., 2019b) and finetuned on a 713 sentence subset of the Edinburgh++ corpus (Cohn et al., 2008).[9] Both DiscAlign and FastAlign have been successfully used for cross-lingual word alignment, with DiscAlign outperforming FastAlign on Arabic-English and Chinese-English alignment by a large margin (Stengel-Eskin et al., 2019).

4.3 Evaluation

Since the baseline aligners are word-level and our model is span-level, in order to have a fair comparison we evaluate on span F_1 (Table 1), computing the overlap between predicted and gold spans.

[8] On rare occasion, some lexical units had fewer than four annotated sentences.

[9] Because the aligner requires fully aligned training data, we did not use larger partially aligned corpora such as the Microsoft Research Paraphrase Corpus.

Method	**P**	**R**	**F_1**
DiscAlign	34.11	39.69	36.69
FastAlign	78.64	72.13	75.25
Df+Mx+Cue+SBCE	**96.75**	**88.24**	**92.30**

Table 1: Soft-match span F_1 on the test set, calculated using the precision and recall of predicted tokens vs. gold truth tokens; allows for partial matches. Word-level baselines are compared against our best performing BERT-based span alignment model.

Method	**P**	**R**	**F_1**
DiscAlign	(29.82)	(29.82)	29.82
FastAlign	(71.02)	(71.02)	71.02
Cue	10.39	9.77	10.07
Mx	80.65	77.92	79.26
Df	87.31	85.42	86.36
Mx+Cue	87.50	86.49	86.99
Df+Cue	88.74	86.96	87.84
Df+Mx	**89.27**	87.29	88.27
Df+Mx+Cue	89.15	88.19	88.67
Df+Mx+Cue+SBCE	89.14	**88.99**	**89.06**

Table 2: Exact-match span F_1 on the test set; does not allow for partial matches. $\{$Disc, Fast$\}$Align are both word alignment models, where ours were trained for span alignment. **Cue** adds positional information, **Mx** adds max pooling of span representations, **Df** adds element-wise difference of span representations, and **SBCE** adds soft binary cross entropy.

Predicted spans are obtained from word-level alignments by following alignments of each word in the source span to the paraphrase and taking the maximal span covered by those alignments. The span F_1 metric allows partial credit to be awarded in cases where the predicted span and gold span do not match exactly. We also evaluate exact span match (Table 2), where credit is awarded only if the predicted span matches the gold span exactly.

4.4 Results

Table 1 shows that when evaluated on span overlap, our model significantly outperforms both baselines. Table 2 shows that these results generalize to the more difficult exact match setting. While all models experience a drop in performance, our model continues to outperform both baselines. Because no prediction threshold was used in the baselines (unlike in our model) the values for precision and recall are equal for the baselines but can differ slightly for our model, as the addition of a threshold allows the model to incur a false negative without a false positive.

4.5 Discussion

Because our model is trained to choose spans by design, the probability of an exact match is higher a priori: Rather than choosing the words of a span independently, it chooses them as a set, with limits on the difference in length between the source and target spans. This is reflected in the better performance of our model on both evaluation metrics. The bottom two rows of Table 2 show that SBCE boosts recall with almost no loss of precision. Our intuition is that the increased proportion of non-zero labels causes the model to make more threshold-exceeding predictions on reasonable candidate spans. We expect that future work—for example, experimenting with alternative labeling strategies or model architectures—may lead to improvements in the span alignment component of our overall framework, although our core intended contribution is the framework itself and its successful application to data augmentation and subsequent improved performance on a downstream task. In particular, we expect model performance to increase as contextualized representations become more powerful.

4.6 Analysis

Memorization Since our model could be memorizing a large static mapping between lexical units, we tested the ability of our model to generalize by running an experiment where all source-side spans in the test set were guaranteed to not have been observed at training time.[10] Under this setting, the loss of F_1 was minimal (roughly 2 points), suggesting that the model is robust to unseen lexical units.

Syntactic Diversity In Table 3 we measure the amount of syntactic diversity that is introduced by

[10] In our main experiments, (original sentence, trigger, paraphrase, alignment) combinations are disjoint between train and test, but it is possible to observe the same trigger (with a different sentence, paraphrase, or alignment) at both train- and test-time.

Comparison	¬EXT	JCD	WJCD
Source vs. Model	31.08	28.94	29.31
Source vs. Gold	34.54	30.60	31.05

Table 3: Percentage of part of speech differences between source-side and reference-side spans. **¬EXT** is the percentage of span pairs whose POS tags did not exactly match. Since an exact match would be precluded in the case of differing span lengths we also include Jaccard distance[11] (**JCD**) and weighted Jaccard distance[12] (**WJCD**), the latter of which is sensitive to tag frequency. In row one the reference-side spans are produced by the alignment model whereas in row two the analysis uses gold manually annotated span labels.

running a part-of-speech tagger[13] over each (source, reference) pair and then comparing the POS tag(s) of the source-side trigger span (over natural language) to the POS tag(s) of the reference-side span (over automatically paraphrased text). Spans predicted by the alignment model are reasonably syntactically diverse, having different POS tags than those of the source-side span 31.08% of the time. The alignment model has a slight inclination to retain the part of speech of source-side span given that gold spans are more diverse (34.54%).

Multi-word Spans Figure 4 shows the distribution of length for source-side spans, reference-side gold spans, and model-predicted spans. Alignment model F_1 over the test set is given for each bin. Model-predicted spans and reference spans are shorter than source-side spans on average; 1.22, 1.34, and 1.53 tokens, respectively. Multi-word spans constitute 14.71% of model-predicted spans, 21.88% of reference-side spans, and 34.18% of source-side spans. The shorter average span length of the gold spans (annotated over automatic paraphrases) suggests the synthetic text from our paraphrase model may be biased in ways that distinguish it from natural language. Although the alignment model predicts shorter spans on average, when it does predict a longer span, F_1 is higher.

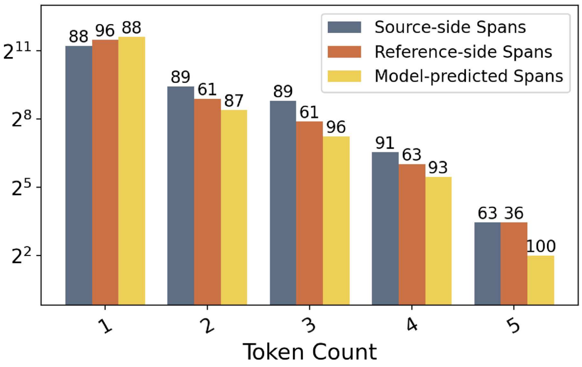


Figure 4: Distribution of source-side, reference-side, and model-predicted span length in the test set, with per-bin F_1 above each bar.

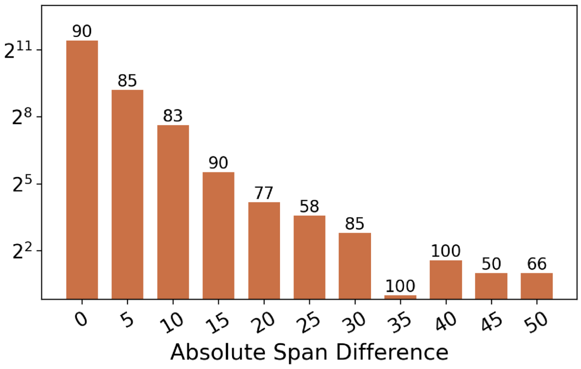


Figure 5: Distribution of absolute difference between source-side and reference-side span positions in the test set, with per-bin F_1.

Source and Reference Span Positions Figure 5 shows the distribution of absolute difference between source and reference spans, defined as $d(a,b) = |a_1 - b_1| + |a_2 - b_2|$, giving a measure of the positional differences between spans in FrameNet sentences and their corresponding paraphrases. The first three bins (0, 5, and 10) contain 97.49% of the data. F_1 experiences a modest decrease across the first three bins and is unsteady in subsequent bins due to data sparsity.

5 Iterative Augmentation Procedure

Our alignment model (§4) is paired with a lexically constrained paraphrase model (§3) to form an iterative procedure for augmenting data of the form: $(\texttt{Sentence}_i, \{(\texttt{start}_{i,1}, \texttt{end}_{i,1}, \texttt{type}_{i,1}), ...\})$. The process consists of three steps: constraint expansion, paraphrasing, and alignment. In constraint expansion, we negatively constrain on a text span of interest, including its upper/lowercase counterparts and morphological variants using the `pattern` software

[11] Of two sets S and T: $1 - \frac{|S \cap T|}{|S \cup T|}$.

[12] Of two vectors $\mathbf{u}$ and $\mathbf{v}$: $1 - \frac{\sum_i \min(\mathbf{u}_i, \mathbf{v}_i)}{\sum_i \max(\mathbf{u}_i, \mathbf{v}_i)}$.

[13] `https://github.com/explosion/spacy-models/releases//tag/en_core_web_lg-2.3.1`.

package (Smedt and Daelemans, 2012). By applying negative constraints, the paraphrase model is forced to generate a semantically equivalent sentence with a different surface form of the labeled text, creating a target for the alignment model. In the alignment stage, we score the original text span's representation together with each candidate span in the paraphrase and choose the one with the highest score under the model. Using the newly obtained aligned phrase as input to constraint expansion, we repeat the process for a predetermined number of iterations.

Although we apply the iterative augmentation procedure to English language text, the method could be applied to other languages as long as a dataset exists on which to train a monolingual paraphrase model, which could then be used to generate data to be manually annotated for the span alignment training set. Our paraphrase model is trained on data that ultimately needs backtranslation, which requires a set of aligned bilingual sentence pairs, though there are other types of paraphrase models that use only monolingual data (Roy and Grangier, 2019). Software such as `pattern` that allow the procedure to negatively constrain on morphological inflections of a given word would speed up the rate at which new lemmas are generated; however, even without such software, the paraphrase model would eventually discover inflections independently and negatively constrain on them. Languages with richer morphological structure would benefit more from this kind of software as the paraphrase model might otherwise waste computational resources generating sentences with many inflections of the same word.

6 Experiments

Our approach lends itself to two applications: In §6.1 we are concerned with building a semantic resource from scratch, whereas in §6.2 we are concerned with expanding a pre-existing resource. We demonstrate the usefulness of our approach on downstream tasks in §6.3, where we apply our generated paraphrastic dataset to the task of Frame ID. Following Pavlick et al. (2015), we consider FrameNet as an illustrative resource motivating augmentation. In all experiments we treat each system output (paraphrase and alignment) as evoking the same frame as the original FrameNet input sentence.

6.1 Building FrameNet (nearly) from Scratch

To simulate constructing a resource using iterative paraphrastic augmentation, we consider what FrameNet would have looked like in its earliest stages of development.[14] Using each object's ''created date'' attribute, we ablate all but the 20 earliest-added frames, the three earliest-added lexical units per frame, and the three earliest-added annotations per lexical unit, for a total of at most[15] 180 annotations in our seed corpus.

We then ran 10 iterations of augmentation with a beam size of 30 for the paraphrase model. For each input, we ran the alignment model on each of the top-20 beam elements and chose the beam element with the highest score under the alignment model. This resulted in 1710 paraphrased and aligned sentences[16] and 1316 unique (`Frame`, `LexicalUnit`) pairs. Some generated words lemmatized to the same form, causing the number of lexical units to be less than the number of sentences.

Automatic Evaluation Prior to ablation, the 20 frames in the seed corpus contained a total of 360 lexical units, of which 60 were chosen to remain in the seed. We treat the set of 300 unobserved lexical units as gold standard and compute precision and recall of the lexical units contained within the 1710-sentence system output. Lexical units were only considered correct if they were in the correct frame; comparisons were made between (`Frame`, `LexicalUnit`) pairs.

Our system produced 128 true positives, 1188 false positives, and 112 false negatives,[17] yielding a precision of 9.7% and recall of 53.33%.

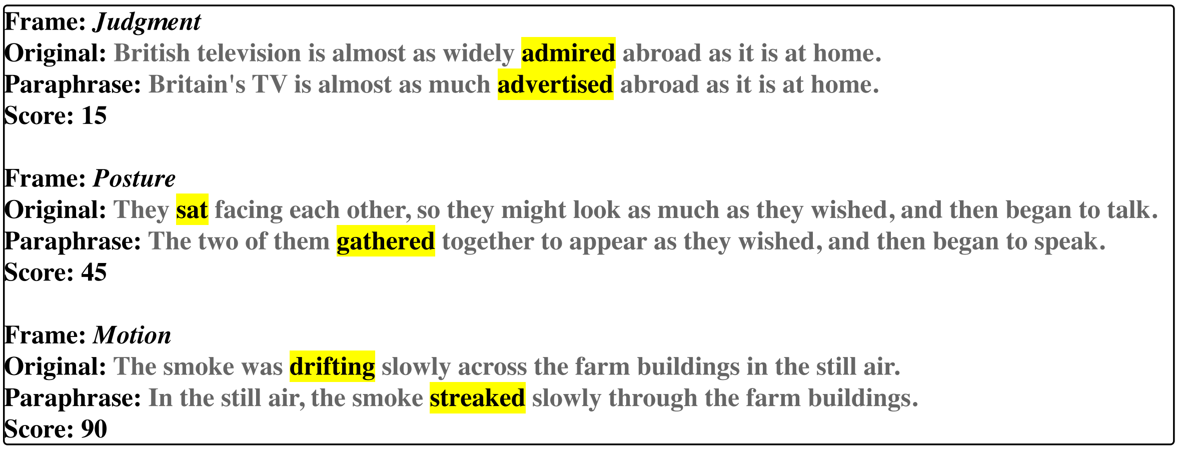
Since the hypothetical complete set of lexical units for a given frame is vast and the lexical units already in FrameNet constitute a small subset of the complete set, we are not surprised to see the probability is low that the lexical units generated by our framework fall into the small subset

[14] The decision to select our seeds based on frame creation date—in contrast to some other sub-selection strategy—was informed by discussions with FrameNet creators.

[15] In practice we were left with slightly fewer (171), as we removed sentences that were observed by the alignment model at training time, and some lexical units contained fewer than three annotations.

[16] 171 sentences rewritten 10 times each.

[17] We exclude from the false negative count the 60 lexical units in the seed corpus since they are guaranteed to not be generated due to the negative constraints placed upon them.



Frame: ***Judgment***
Original: **British television is almost as widely admired abroad as it is at home.**
Paraphrase: **Britain's TV is almost as much advertised abroad as it is at home.**
Score: 15

Frame: ***Posture***
Original: **They sat facing each other, so they might look as much as they wished, and then began to talk.**
Paraphrase: **The two of them gathered together to appear as they wished, and then began to speak.**
Score: 45

Frame: ***Motion***
Original: **The smoke was drifting slowly across the farm buildings in the still air.**
Paraphrase: **In the still air, the smoke streaked slowly through the farm buildings.**
Score: 90

Figure 6: Sample of actual system outputs and associated manually judged scores. Annotators did not see the original sentence when assigning scores but they are provided here for reference. In the first example, the paraphrase model makes a mistake; in the second, the sentence is roughly synonymous but borderline out-of-frame; in the third, both the paraphrase and alignment are high-quality.

already in FrameNet. Upon manual inspection, we found that many of the words predicted by the framework were valid yet absent from FrameNet, motivating us to develop a more sophisticated evaluation method.

Manual Evaluation We conducted a 3-way-redundant manual evaluation of the 1710 system outputs using skilled, locally trained annotators. For each system output—a paraphrase with a highlighted phrase corresponding to the span predicted by the alignment model—we provided a description of the anticipated frame[18] and three gold-standard example annotations[19] to reinforce the frame definition. Workers were then asked to rate three candidate sentences, each with a highlighted trigger phrase, on a scale of 0–100, as to how well the highlighted trigger evoked the given frame in the context of the sentence. Unbeknownst to annotators, of the three candidate sentences in each task, only one of them (in a random position) was an actual system output; the other two were positive or negative gold-standard sentences taken from FrameNet:

1. System output: Frame a and lexical unit b.
2. Gold in-frame: Frame a and lexical unit $\neg b$.
3. Gold out-of-frame (adversarial): Frame $\neg a$.

The scores collected on gold in- and out-of-frame control sentences provide a means to ground the interpretation of scores on system outputs and also enable us to gauge overall annotator understanding of the task by scoring sentences for which we know the correct response.

Since each system output was judged by three distinct annotators, we average each triple of judgments and treat values less than 50 as a rejection (''the highlighted trigger, in the context of the sentence, does not evoke the given frame'') and values greater than or equal to 50 as an acceptance. Gold in- and out-of-frame sentences had acceptance rates of 95.26% and 6.57%, respectively, suggesting workers possessed a relatively strong understanding of the task. Figure 6 provides a sample of actual system outputs and associated individual scores.

Inter-annotator Agreement Fleiss' kappa for the binarized scores of judgments of system outputs is 0.5641, indicating moderate to substantial agreement. Separately, all three annotators made the same binarized judgment 71.18% of the time.

Analysis Figure 7 shows how human judgments distribute over the [0,100] range for in-frame sentences (average 77.36), system outputs (average 59.35), and out-of frame sentences (average 23.17). Judgments of in- and out-of-frame sentences are neatly partitioned, with in-frame sentences being concentrated in the [50,100] range and out-of-frame judgments concentrating in the [0,50] range. Annotators tend not to make

[18] We assume that the paraphrase transformation is label-preserving so the anticipated frame is simply the frame of the original FrameNet sentence.

[19] The trigger words in the example sentences were made to be disjoint with the trigger words in the candidate sentences in order to avoid biasing annotators.

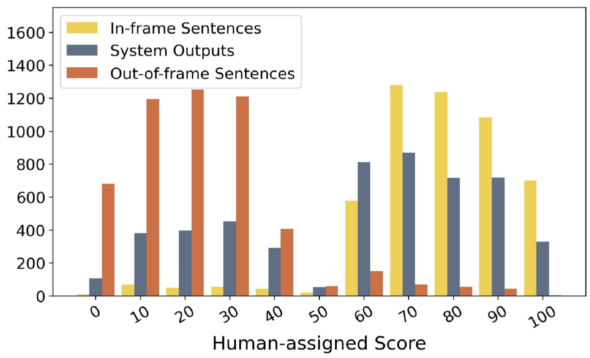


Figure 7: Distribution of human judgments for in-frame sentences, system outputs, and adversarial out-of-frame sentences. System output is unfiltered; in §6.1 we experiment with methods to automatically remove low quality system outputs.

Filtering	**P**	**R**	**Multiple**
Unfiltered	68.25	100	11x
Iter = 1	90.06	13.20	2x
Iter ≤ 3	81.29	35.73	4x
Paraphrase score ≤ 0.6	90.14	5.48	1.42x
Paraphrase score ≤ 0.8	74.86	34.45	4.14x
Aligner score ≥ .99	85.01	32.56	3.61x
Aligner score ≥ .95	76.72	85.00	8.56x
Lax conjunction	87.73	20.82	2.62x
Strict conjunction	92.54	5.31	1.39x
P-Classifier	**95.00**	15.61	2.28x
R-Classifier	81.19	**96.99**	10.27x

Table 4: Human evaluation of system outputs across several filtering methods, with manually judged **P**recision for the subset of outputs remaining after applying the given filter, **R**ecall of sentences manually judged to be acceptable, and the **Multiple** (in terms of number of sentences) of the resulting dataset in relation to the original seed corpus. **Filtering** methods consider the iteration number, and scores from the paraphrase and aligner models for a given system output. The ‘‘lax’’ row applies a filter consisting of the conjunction of the criteria from rows 3, 5, and 7 (relatively lenient conditions) whereas the ‘‘strict’’ row conjoins the criteria from rows 2, 4, and 6 (which are stricter, and lead to higher precision but fewer lexical units).

judgments at the extrema of the range. Judgments of system outputs skew towards the the upper half of the range although they are more split than judgments of in- and out-of-frame sentences. This distribution and the associated averages are calculated using the unfiltered set of system outputs; in §6.1 we test several ways of automatically identifying system outputs that are likely to be low quality, enabling the removal of such outputs and the creation of a higher quality dataset.

Filtering Methods We experiment with several methods of filtering system outputs, providing a trade-off between the competing goals of quality and size. Each system output has an associated iteration number, score under the paraphrase model, and score under the alignment model; each filtering method then uses this information to select a subset of the unfiltered system outputs.

We report the precision (the ratio of elements in the subset that had a score over 50) and recall (the number of elements in the subset with a score over 50, divided by the number of elements in the unfiltered set that also had a score over 50) in Table 4. The upper section of Table 4 presents results for a variety of heuristic filtering methods, for example, the subset of system outputs with an iteration number of three or lower, while the lower section presents results for a neural filtering model.

The neural model takes as input a system output's iteration number, score under the paraphrase model, and score under the alignment model, and produces a score between 0 and 1, where 0 represents a decision to filter an output, and 1 represents a decision to keep it. Architecturally, the model is a feed-forward neural network with two hidden layers, 10 units per hidden layer, and a sigmoid output layer, trained to minimize binary cross entropy loss. We trained one model to favor precision by downweighting the training loss when the label was 1, and a second model to favor recall by downweighting when the label was 0. As training data, we used the 1710 aggregated manual judgments from above (where each system output has a label of 0 or 1), plus 2988 additional judgments collected specifically for this model. We split the data as 90% train (4228) and 10% test (470), and present results,[20] in the lower section of Table 4.

Discussion The upper section of Table 4 suggests that iteration number, paraphrase model score, and aligner model score each have slightly different filtering characteristics, and a simple

[20] Results in the upper section of Table 4 are reported over the 1710 system outputs from §6.1 while the results in the lower section are reported over the 470-element test set.

conjunction of criteria achieves higher precision than any condition alone. The P-Classifier, optimized to select a high-precision subset of the data, achieves higher precision than any of the heuristic methods, and higher recall than the highest-precision heuristic method. The precision of the P-classifier (95%) is roughly the same as the human-level acceptance rate on gold in-frame sentences (95.26%) while generating a resource that is 2.28x as large as the original. A higher recall subset may be obtained with the R-Classifier, which retains 96.99% of acceptable outputs with a precision of 81.19%.

6.2 Expanding Existing FrameNet

In this section we report the results of applying large-scale iterative augmentation to an existing resource. As in our reconstruction experiment, we ran 10 iterations of augmentation, but with minor configuration changes[21] to enable faster processing over the roughly 200,000 FrameNet annotations.[22]

Our unfiltered dataset,[23] which excludes the original FrameNet data, contains 1,983,680 automatically paraphrased and aligned English-language sentences and 495,300 (`Frame`, `Trigger`) pairs[24] in diverse sentential contexts. As the underlying text of our generated resource is automatically paraphrased, it is synthetic and may contain biases that distinguish it from natural language. Of the 495,300 new triggers, 428,416 are unique after applying lemmatization; each lemma has 4.63 automatic in-context annotations on average. We use the filter models from §6.1 to select high quality and high quantity subsets of the unfiltered data; each system output in our data release has an associated score from both filter classifiers to enable post-hoc filtering. The P-Classifier retains 138,797 sentences and 33,332 (`Frame`, `Trigger`) pairs, while the R-classifier retains 1,807,235 sentences and 425,050 pairs. To enable further experimentation, each sentence in our release is linked to FrameNet v1.7.

Because our data only contains alignments of triggers and not frame elements, it cannot be directly used for full FrameNet semantic role labeling (SRL). However, by additionally applying *positive* constraints on frame element spans during lexically constrained decoding, an alignment link may be trivially obtained, allowing our framework to be used for full SRL.

6.3 Using Paraphrastic Data on a Downstream Task

In this section, we use the expanded FrameNet resource from §6.2 to improve model robustness on the task of Frame ID, a key subtask in FrameNet SRL (Das et al., 2010; Hermann et al., 2014).

It is often prohibitively expensive to annotate entire documents under protocols such as FrameNet, and full-document annotation may not provide full coverage of the ontology due to the rarity of some ontological types. A commonly used alternative to full-document annotation is exemplar-based annotation, where several canonical examples (or ''exemplars'') are identified for each ontological type, ensuring full coverage of the ontology. Below, we conduct experiments to show that the addition of paraphrastic data to full-document and exemplar annotations boosts Frame ID model performance.

Task FrameNet parsing (Das et al., 2014; Kshirsagar et al., 2015; Roth and Lapata, 2015; Swayamdipta et al., 2018) is an established task in the field of semantic parsing. Most previous work has viewed FrameNet parsing as SRL, where the goal is to identify the frame and label all frame elements given a sentence with a known trigger span, but little attention has been paid to identifying trigger spans themselves (Das et al., 2014).

Given the practical importance of finding triggers, we focus on jointly identifying both triggers *and* frames, rather than frames alone.

Specifically, given a sequence of words, our task is to find all contiguous subsequences[25] that trigger a frame and to identify the corresponding frames. We pose this as a span tagging problem, with trigger spans being tagged with the associated frame and non-trigger spans tagged as `NULL`.[26]

[21] We used a beam size of 20 to decode paraphrases and ran the alignment model on each of the top-3 beam elements, choosing the beam element with the highest score under the alignment model.

[22] In practice, we filtered out sentences with greater than 80 tokens due to a limitation in the paraphrase model, leaving 198,368, or 99.55% of the original sentences.

[23] `http://nlp.jhu.edu/parabank`.

[24] A (`Frame`, `Trigger`) pair can be thought of as an inflected surface form of a given word sense.

[25] Following the convention of Das et al. (2014) we do not capture discontiguous trigger spans; e.g., we treat *there would be* as an instance of the lexical unit `there be.v`

[26] 0.05% of the full-text annotations contained triggers that evoked two frames; we discard the second frame for simplicity.

Model We adopt a two-pass Long Short-Term Memory (LSTM) model for the Frame ID task. We first convert the sentence $\mathbf{S} = \langle s_1, s_2, \ldots, s_I \rangle$ into a sequence of embedding vectors $\langle \mathbf{e}_1^0, \mathbf{e}_2^0, \ldots, \mathbf{e}_I^0 \rangle$, where each embedding $\mathbf{e}_i^0$ is a concatenation of GloVe, BERT (first subtoken, fixed), character, and POS embeddings (Pennington et al., 2014; Devlin et al., 2018; Alberti et al., 2019). Next, we use a l-layer stacked bidirectional LSTM model (Hochreiter and Schmidhuber, 1997) to obtain a contextual embedding for each word:

$$\langle \mathbf{e}_1^l, \mathbf{e}_2^l, \ldots, \mathbf{e}_I^l \rangle = \text{BiLSTM}(\langle \mathbf{e}_1^0, \mathbf{e}_2^0, \ldots, \mathbf{e}_I^0 \rangle)$$

We then apply another unidirectional LSTM model on top to get a representation for a span $\mathbf{s}_{i:j}$:

$$\mathbf{e}_{i:j} = \text{LSTM}(\langle \mathbf{e}_i^l, \mathbf{e}_{i+1}^l, \ldots, \mathbf{e}_j^l \rangle)$$

As in the alignment model, we empirically choose a maximum span length[27] to reduce the computational complexity from $O(I^2)$ to $O(I)$.

A fully connected neural network is then applied to transform the representation $\mathbf{e}_{i:j}$ into a logit vector, which is then translated by softmax into a distribution over the label set composed of frames and NULL. We train with cross-entropy loss.

The FrameNet corpus provides two sets of annotated sentences: full-text and exemplar, where full-text annotations consist of exhaustively annotated documents, whereas exemplar annotations are only annotated with one frame for every sentence. For the full-text sentences, we treat both the trigger and non-trigger spans as training examples, but for the exemplar and paraphrastic sentences, non-trigger spans are excluded due to the fact that they represent incomplete annotations rather than true negative examples. Furthermore, Das et al. (2014) pointed out that some triggers are not annotated in the full-text sentences, leading to false negative training examples. In light of this, we apply the label smoothing trick (Szegedy et al., 2016)[28] on negative examples to smooth the point distribution, resulting in a 3-point F_1 improvement.

Experiments To illustrate the utility of the paraphrastic data generated by our augmentation framework in low-resource settings, we sample $\{10\%, 20\%, \ldots, 100\%\}$ of full-text sentences as training data. In two experiments we also incorporate exemplar sentences.[29] For each sample k of original FrameNet sentences, we conduct a parallel experiment adding in corresponding paraphrases of sentences in k, taken from our resource in §6.2.

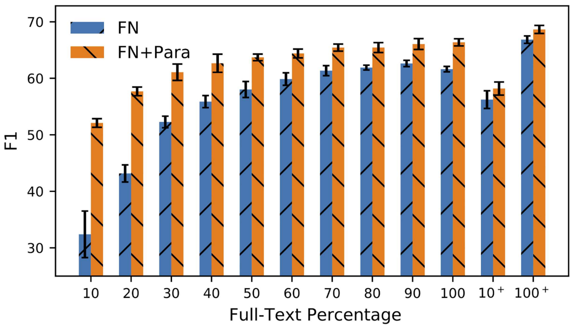


Figure 8: Frame ID results with different full-text percentages, with and without paraphrastic data. Each experiment is repeated 5 times with resampled training data. ''FN'' is the original FrameNet data, and ''FN + Para'' uses both FrameNet and paraphrastic data for training. 10^+ and 100^+ indicate that exemplar sentences are added.

Using the FrameNet v1.7 release,[30] we adopt the same development and test split proposed by Das and Smith (2011), treating all other documents as training examples. We use greedy search to find the optimal hyperparameters, used for conducting all experiments. We evaluate model performance using Frame ID F_1 score, where a frame prediction is viewed as true positive when both the trigger span and frame match exactly.

Results and Analysis Based on the results shown in Figure 8, we can see that we get higher F_1 when using more data, and paraphrases boost the F_1 for every experiment, particularly in low-resource settings where only a small fraction of the full-text data is accessed. If we provide the model with both full-text and exemplar sentences, the improvement brought by paraphrases is less, but still significant. Peng et al. (2018) reported state-of-the-art results on Frame ID with 90.00% accuracy on FrameNet v1.5; however, this is not comparable with our result because their model is provided gold triggers and has only to identify the

[27] In this case 3, which only excludes 0.24% of the target words, which are treated as false negative during evaluation.

[28] A smoothing factor 0.2 is empirically chosen.

[29] We extract the first 3 lexical units for every frame and the first 3 exemplar sentences for every lexical unit. The lexical units and exemplar sentences are sorted by the annotation date.

[30] Accessed using the FrameNet support within NLTK (Schneider and Wooters, 2017) to process the raw data.

frame, whereas our model jointly identifies both triggers and frames.

Comparison to Lexical Substitution To demonstrate[31] that paraphrases are more beneficial when they are contextual, with sentence-level alterations to the input, rather than the result of simple word- or short phrase-level substitutions (c.f. Ganitkevitch et al., 2013) we conducted additional experiments using paraphrases obtained via lexical substitution.

For each FrameNet sentence and its corresponding paraphrase, we replace the original trigger in the FrameNet sentence with the automatically aligned trigger from the paraphrase. To ensure that the resulting sentences are grammatical, we only keep sentences that have the same part-of-speech tag(s) over the trigger span as the original sentences; this filter removed approximately 35% of the resulting word-level paraphrases. We use 100% of the full-text FrameNet annotations as the base training data and reuse the same hyperparameters as our previous experiments.

The model trained on full-text annotation + word-level paraphrases achieved an F_1 score of 49.59 ± 2.59 (averaged over 5 runs), which is lower than the full-text only result (61.63 ± 0.49) and lower still than the full-text + sentence-level paraphrase result (66.37 ± 0.63). This suggests that simple lexical substitution produces lower quality paraphrases, translating into the result that these sentences actually hurt performance on Frame ID.

Future Work While we have shown that paraphrasing is beneficial for training a Frame ID model in a low-resource setting, it is important to be aware of the limitations of paraphrastic data. The paraphrasing generation process does not guarantee that the resulting data will be beneficial since it is possible that some of the paraphrases are already well understood by the model (Ribeiro et al., 2018). Furthermore, paraphrases could include lexical units that fall outside of the ontology being used, leading to a negative impact with respect to evaluation.

Future work may investigate tactical data augmentation such as the filtering score proposed by Ribeiro et al. (2018) or automatic scoring functions such as those proposed by Lee et al. (2019). Our method might be extended to task-oriented dialog in a number of domains, for example, SMCalFlow (Andreas et al., 2020), where data sparsity often poses a problem.

7 Conclusion

We introduced a novel approach for iterative construction of semantic resources via automatic paraphrasing. To demonstrate two possible uses of our framework, we simulated the rapid creation of a new semantic resource from a small seed corpus and generated a large-scale expansion of an existing resource. The latter experiment, run on FrameNet data, generated a lexically diverse dataset with 495,300 unique (`Frame`, `Trigger`) pairs in diverse sentential contexts, 50x the number of such pairs originally in FrameNet, which we release to the community alongside our 36,417-instance span alignment dataset.

Acknowledgments

We thank Xu Han for creating the span alignment annotation interface and the anonymous reviewers from current and past versions of this paper for their insightful comments and suggestions. This research was supported by DARPA AIDA, DARPA KAIROS, and NSF (award 1749025). The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes. The views and conclusions contained in this publication are those of the authors and should not be interpreted as representing official policies or endorsements of DARPA, NSF, or the U.S. Government.

References

C. Alberti, K. Lee, and M. Collins. 2019. A BERT Baseline for the Natural Questions. arXiv 1901.08634.

Jacob Andreas, John Bufe, David Burkett, Charles Chen, Josh Clausman, Jean Crawford, Kate Crim, Jordan DeLoach, Leah Dorner, Jason Eisner, et al. 2020. Task-oriented dialogue as dataflow synthesis. *Transactions of the Association for Computational Linguistics*, 8:556–571. **DOI:** `https://doi.org/10.1162/tacl_a_00333`

C. Baker, M. Ellsworth, and K. Erk. 2007. Frame semantic structure extraction. In *SemEval*.

[31] Aside from these empirical results we also describe two a priori issues with lexical substitution-based methods in §2 – Automatic Lexicon Expansion.

Jonathan Berant and Percy Liang. 2014. Semantic parsing via paraphrasing. In *ACL*, Baltimore, Maryland. Association for Computational Linguistics. **DOI:** https://doi.org/10.3115/v1/P14-1133

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

Aljoscha Burchardt and Anette Frank. 2006. Approaching textual entailment with lfg and framenet frames. **DOI:** https://doi.org/10.3115/1654536.1654540

Do Kook Choe and David McClosky. 2015. Parsing paraphrases with joint inference. 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 1223–1233, Beijing, China. Association for Computational Linguistics. **DOI:** https://doi.org/10.3115/v1/P15-1118

Trevor Cohn, Chris Callison-Burch, and Mirella Lapata. 2008. Constructing corpora for the development and evaluation of paraphrase systems. *Computational Linguistics*, 34(4):597–614. **DOI:** https://doi.org/10.1162/coli.08-003-R1-07-044

D. Das, D. Chen, A. F. T. Martins, N. Scneider, and N. A. Smith. 2014. Frame-Semantic Parsing. *Computational Linguistics*, 40(1):9–56. **DOI:** https://doi.org/10.1162/COLI_a_00163

D. Das and N. A. Smith. 2011. Semi-supervised frame-semantic parsing for unknown predicates. In *Association for Computational Linguistics and Human Language Technology (ACL-HLT)*.

Dipanjan Das, Nathan Schneider, Desai Chen, and Noah A Smith. 2010. Probabilistic frame-semantic parsing. In *NAACL*, pages 948–956. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *IWP*.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM Model 2. In *NAACL:HLT*, pages 644–648.

Charles J. Fillmore. 1982. *Linguistics in the Morning Calm: Selected Papers from SICOL-1981*. Hanshin.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: The paraphrase database. In *NAACL-HLT*, pages 758–764.

Daniel Gildea and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288. **DOI:** https://doi.org/10.1162/089120102760275983

Karl Moritz Hermann, Dipanjan Das, Jason Weston, and Kuzman Ganchev. 2014. Semantic frame identification with distributed word representations. In *ACL*, Baltimore, Maryland. Association for Computational Linguistics. **DOI:** https://doi.org/10.3115/v1/P14-1136

S. Hochreiter and J. Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780. **DOI:** https://doi.org/10.1162/neco.1997.9.8.1735, **PMID:** 9377276

J. Edward Hu, Huda Khayrallah, Ryan Culkin, Patrick Xia, Tongfei Chen, Matt Post, and Benjamin Van Durme. 2019a. Improved lexically constrained decoding for translation and monolingual rewriting. In *NAACL*.

J. Edward Hu, Rachel Rudinger, Matt Post, and Benjamin Van Durme. 2019b. ParaBank: Monolingual bitext generation and sentential paraphrasing via lexically-constrained neural machine translation. *AAAI*.

J. Edward Hu, Abhinav Singh, Nils Holzenberger, Matt Post, and Benjamin Van Durme. 2019c. Large-scale, diverse, paraphrastic bitexts via sampling and clustering. In *CoNLL*.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77. **DOI:** https://doi.org/10.1162/tacl_a_00300

Tom Ko, Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur. 2015. Audio augmentation for speech recognition. In *Sixteenth Annual Conference of ISCA*.

M. Kshirsagar, S. Thomson, N. Schneider, J. Carbonell, N. A. Smith, and C. Dyer. 2015. Frame-semantic role labeling with heterogeneous annotations. In *Association for Computational Linguistics and International Joint Conference on Natural Language Processing (ACL-IJCNLP)*. **DOI:** https://doi.org/10.3115/v1/P15-2036

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *EMNLP 2018: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics. **DOI:** https://doi.org/10.18653/v1/D18-2012, **PMID:** 29382465

Ashutosh Kumar, Satwik Bhattamishra, Manik Bhandari, and Partha Talukdar. 2019. Submodular optimization-based diverse paraphrasing and its effectiveness in data augmentation. In *NAACL: HLT*. **DOI:** https://doi.org/10.18653/v1/N19-1363

Kyungjae Lee, Sunghyun Park, Hojae Han, Jinyoung Yeo, Seung-won Hwang, and Juho Lee. 2019. Learning with limited data for multilingual reading comprehension. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2833–2843.

Jonathan Mallinson, Rico Sennrich, and Mirella Lapata. 2017. Paraphrasing revisited with neural machine translation. In *EACL*, pages 881–893, Valencia, Spain. Association for Computational Linguistics. **DOI:** https://doi.org/10.18653/v1/E17-1083

Jessica Ouyang and Kathleen McKeown. 2019. Neural network alignment for sentential paraphrases. In *ACL*, pages 4724–4735. **DOI:** https://doi.org/10.18653/v1/P19-1467

Ellie Pavlick, Travis Wolfe, Pushpendre Rastogi, Chris Callison-Burch, Mark Dredze, and Benjamin Van Durme. 2015. FrameNet+: Fast Paraphrastic Tripling of FrameNet. In *ACL 2015*. **DOI:** https://doi.org/10.3115/v1/P15-2067

H. Peng, S. Thomson, S. Swayamdipta, and N. A. Smith. 2018. Learning Joint Semantic Parsers from Disjoint Data. In *North American Association for Computational Linguistics (NAACL)*, pages 1492–1502. **DOI:** https://doi.org/10.18653/v1/N18-1135, **PMCID:** PMC6327562

J. Pennington, R. Socher, and C. D. Manning. 2014. GloVe: Global Vector for Word Representation. In *Empirical Methods in Natural Language Processing (EMNLP)*. **DOI:** https://doi.org/10.3115/v1/D14-1162

Matt Post and David Vilar. 2018. Fast lexically constrained decoding with dynamic beam allocation for neural machine translation. In *NAACL*, pages 1314–1324, New Orleans, Louisiana. Association for Computational Linguistics. **DOI:** https://doi.org/10.18653/v1/N18-1119

Anton Ragni, Kate M. Knill, Shakti P. Rath, and Mark JF Gales. 2014. Data augmentation for low resource languages. In *Fifteenth Annual Conference ISCA*.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Semantically equivalent adversarial rules for debugging NLP models. In *ACL*. Association for Computational Linguistics. **DOI:** https://doi.org/10.18653/v1/P18-1079

M. Roth and M. Lapata. 2015. Context-aware Frame-Semantic Role Labeling. *Transactions of the Association for Computational Linguistics (TACL)*. **DOI:** https://doi.org/10.1162/tacl_a_00150

Aurko Roy and David Grangier. 2019. Unsupervised paraphrasing without translation. In

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 6033–6039, Florence, Italy. Association for Computational Linguistics. **DOI:** https://doi.org/10.18653/v1/P19-1605

Josef Ruppenhofer and Ines Rehbein. 2012. Semantic frames as an anchor representation for sentiment analysis. In *The Workshop in Computational Approaches to Subjectivity and Sentiment Analysis*, pages 104–109, Jeju, Korea. Association for Computational Linguistics.

N. Schneider and C. Wooters. 2017. The NLTK FrameNet API: Designing for discoverability with a rich linguistic resource. In *Empirical Methods in Natural Language Processing (EMNLP)*. **DOI:** https://doi.org/10.18653/v1/D17-2001

Dan Shen and Mirella Lapata. 2007. Using semantic roles to improve question answering. In *EMNLP-CoNLL*, pages 12–21, Prague, Czech Republic. Association for Computational Linguistics.

Connor Shorten and Taghi M. Khoshgoftaar. 2019. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):60. **DOI:** https://doi.org/10.1186/s40537-019-0197-0

Tom De Smedt and Walter Daelemans. 2012. Pattern for Python. *Journal of Machine Learning Research*, 13(Jun):2063–2067.

Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. 2006. Semantic taxonomy induction from heterogenous evidence. In *ICCL/ACL*. **DOI:** https://doi.org/10.3115/1220175.1220276

Elias Stengel-Eskin, Tzu-ray Su, Matt Post, and Benjamin Van Durme. 2019. A discriminative neural model for cross-lingual word alignment. In *EMNLP-IJCNLP*, pages 909–919. **DOI:** https://doi.org/10.18653/v1/D19-1084

S. Swayamdipta, S. Thomson, K. Lee, L. S. Zettlemoyer, C. Dyer, and N. A. Smith. 2018. Syntactic Scaffolds for Semantic Structures. In *Empirical Methods in Natural Language Processing (EMNLP)*. **DOI:** https://doi.org/10.18653/v1/D18-1412

C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. 2016. Rethinking the Inception Architecture for Computer Vision. In *Computer Vision and Pattern Recognition (CVPR)*. **DOI:** https://doi.org/10.1109/CVPR.2016.308

Su Wang, Rahul Gupta, Nancy Chang, and Jason Baldridge. 2018. A task in a suit and a tie: paraphrase generation with semantic augmentation. **DOI:** https://doi.org/10.1609/aaai.v33i01.33017176

Yushi Wang, Jonathan Berant, and Percy Liang. 2015. Building a semantic parser overnight. In *ACL-IJCNLP*, Beijing, China. Association for Computational Linguistics. **DOI:** https://doi.org/10.3115/v1/P15-1129

John Wieting and Kevin Gimpel. 2018. ParaNMT-50M: Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. In *ACL*, pages 451–462. ACL. **DOI:** https://doi.org/10.18653/v1/P18-1042

Xuchen Yao, Benjamin Van Durme, Chris Callison-Burch, and Peter Clark. 2013a. A lightweight and high performance monolingual word aligner. In *ACL*, volume 2, pages 702–707.

Xuchen Yao, Benjamin Van Durme, Chris Callison-Burch, and Peter Clark. 2013b. Semi-Markov phrase-based monolingual alignment. In *EMNLP*, pages 590–600, Seattle, Washington, USA. Association for Computational Linguistics.