One-Shot Exemplification Modeling via Latent Sense Representations

John Harvill[1], **Hee Suk Yoon**[2], **Eunseop Yoon**[2], **Mark Hasegawa-Johnson**[1], **Chang D. Yoo**[2]
[1]University of Illinois Urbana-Champaign,
[2]Korea Advanced Institute of Science and Technology,
{harvill2, jhasegaw}@illinois.edu, {hskyoon, esyoon97, cd_yoo}@kaist.ac.kr

Abstract

Exemplification modeling is a recently proposed task that aims to produce a viable sentence using a target word that takes on a specific meaning. This task can be particularly challenging for polysemous words since they can have multiple meanings. In this paper, we propose a one-shot variant of the exemplification modeling task such that labeled data is not needed during training, making it possible to train our system using a raw text corpus. Given one example at test time, our proposed approach can generate diverse and fluent examples where the target word accurately matches its intended meaning. We compare our approach to a fully-supervised baseline trained with different amounts of data and focus our evaluation on polysemous words. We use both automatic and human evaluations to demonstrate how each model performs on both seen and unseen words. Our proposed approach performs similarly to the fully-supervised baseline despite not using labeled data during training.

1 Introduction

Many vocabulary words can represent different meanings depending on the contexts in which they are placed. This inherent semantic ambiguity has led to two types of NLP tasks: **1**) Determining the correct meaning (sense) of a word in context and **2**) Generating a sentence where a target word takes on a desired meaning. Word Sense Disambiguation (WSD) and Definition Modeling (DM) are both tasks in the first category. WSD is a classification task where the correct sense must be chosen for a word in context from a predefined sense inventory (Navigli, 2009; Barba et al., 2021c,a; Raganato et al., 2017), and DM is a related task where the goal is to produce a string representing the definition for a word in context (Bevilacqua et al., 2020). Exemplification modeling (EM), recently proposed by Barba et al. (2021b), falls under the second category. EM can be seen as the reverse problem of DM where the desired inputs and outputs are inverted. Given a target word, DM seeks to produce a definition given the sentence, while EM seeks to produce a sentence given the desired definition. EM is useful for generating example sentences for specific word senses when constructing dictionaries (He and Yiu, 2022) and data augmentation for training a WSD system (Barba et al., 2021b).

The two prominent works on EM thus far are ExMaker (Barba et al., 2021b), where the task is first introduced, and Controllable Dictionary Example Generation (CDEG) (He and Yiu, 2022), where generated example sentences are controlled in terms of length and lexical complexity. Both methods require a corpus consisting of (lemma, definition, example sentence) tuples where each example sentence uses the target lemma with the given definition. An autoregressive neural model is trained via minimization of the cross-entropy loss such that inputting the lemma and definition encourages the production of the corresponding example sentence. We corroborate the finding from He and Yiu (2022) that the performance of these supervised approaches depends on the amount of training data available, where the full dataset (Oxford Dictionary) contains 1.3M examples. Collecting such a large dataset can be difficult and may not be feasible for other languages. This difficulty is the main motivation for the approach we propose in this paper. We demonstrate that it is possible to perform exemplification modeling in a one-shot setting, where (lemma, definition, example sentence) tuples are no longer necessary for training. Only a raw, unlabeled text corpus is required.

In this paper, we make the following contributions: **1**) Propose a one-shot format for EM where one example sentence using the target word correctly is provided at inference time, instead of giving the target word and its definition. **2**) Propose a neural architecture that can accurately perform our one-shot format of EM without requiring la-

Proceedings of the 8th Workshop on Representation Learning for NLP (RepL4NLP 2023), pages 303–314

beled data for training. **3**) Evaluate our proposed approach in terms of target word semantic match, sentence diversity, and sentence fluency against a variety of baseline settings that provide deeper insight into the strengths and limitations of both methods. **4**) Provide extensive examples of generated data from our proposed approach that demonstrate its ability to create fluent sentences using the target word with its intended meaning. **5**) Provide a qualitative analysis of how reference example sentence length affects generation length and example diversity.

2 Related Work

Constrained Text Generation. EM falls under constrained text generation, where generated text must satisfy a given set of requirements. There already exists a multitude of constrained text generation tasks, including sentiment transfer (Luo et al., 2019), style transfer (Fu et al., 2018), lexically-constrained decoding (Dinu et al., 2019; Lin et al., 2020; Lu et al., 2022), word ordering (Zhang and Clark, 2015), storytelling (Fan et al., 2018; Yao et al., 2019), essay generation (Yang et al., 2019), paraphrasing (Zhou and Bhat, 2021), and definition modeling (Bevilacqua et al., 2020). These approaches vary in whether finetuning is necessary or decoding is constrained in some way. Approaches involving finetuning require training datasets but are faster at inference time. In contrast, constrained decoding can be applied to a pre-trained model but requires much more computational power during generation. Our proposed work falls under the first category; we require a training dataset of raw text but do not interfere with the decoding process (i.e. nucleus sampling).

Definition Modeling. EM is most similar to DM, since the input of one task is the output of the other. Generationary (Bevilacqua et al., 2020) is a recent DM approach that learns to create definitions for an arbitrary text span within a sentence. Similar to ExMaker (Barba et al., 2021b), it uses a neural encoder-decoder architecture. During training, Generationary takes as input a sentence where the target word or phrase has been enclosed in the `<define>` token and is encouraged to generate the gold definition via the cross-entropy loss. At inference time, Generationary can produce a high-quality contextual definition of an arbitrary text span, generalizing to words or phrases not seen during training.

3 One-Shot Sentence Generation

Exemplification Modeling (EM) was first proposed by Barba et al. (2021b). The goal of EM is to produce a sentence containing a target word, where the word represents a specific meaning. The proposed system, ExMaker (Barba et al., 2021b), uses as input a (lemma, definition) pair. A proper input/output example is given below:

Input	Output
contract: be stricken by an illness	He might contract pneumonia.

Training of ExMaker (Barba et al., 2021b) requires paired data where the sense of the target word is known in the example sentence. Such paired data is not abundant and can be difficult to collect. We propose a one-shot variant of EM that allows us to train a system without paired data in both a self-supervised and semi-supervised fashion, requiring only one reference example at inference time. Instead of using a (lemma, definition) pair as input, we use as input an example sentence where the target word has a desired meaning. We then produce a new sentence where the target word has the same meaning. An example is given below:

Input	Output
contract: The athlete signed the contract.	They agreed to the terms of the contract.

As we will show in the following section, this form of EM has the advantage that it does not require sense inventories for training nor generation.

Training. We can use a neural autoencoder to solve our proposed format of EM. Given a sentence, we want to reconstruct it by conditioning on a latent vector representation of the target word meaning extracted from the sentence itself. We denote these vectors as Latent Sense Representations (LSR) and call our overall approach Sense2Sentence (S2S). We denote the LSR as l, the target word as w, and the sentence as s. During training, we select w uniformly from a word-level tokenization of s and maximize the following:

$$p(s|l,w) = \prod_{i=2}^{|s|} p(s_i|s_{1:i-1}, l, w) \quad (1)$$

We model the conditional distribution in Equation 1 by finetuning a pretrained BART (Lewis et al., 2020) model. The representation of w is extracted

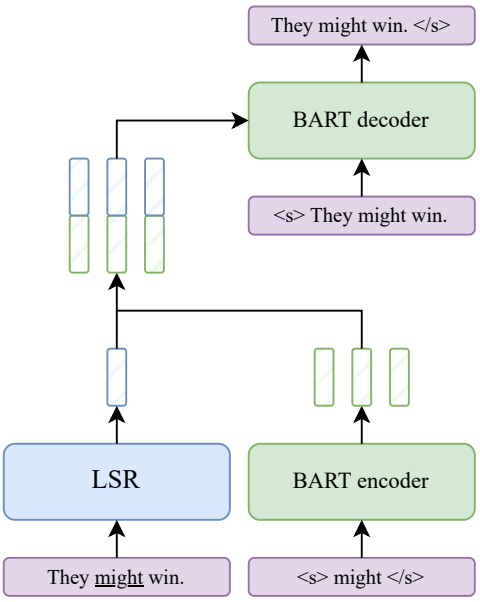

Figure 1: Training approach for S2S. LSR is kept frozen and produces a single fixed-length embedding of the target word sense that is concatenated to all timesteps of the BART encoder output.

from the BART encoder by passing the target word in isolation. The vector l is then concatenated to all timesteps of the BART encoder output. This joint representation is then passed to the BART decoder as conditioning input (see Figure 1). We use cross-entropy loss with teacher-forcing to train the model on BookCorpus (Zhu et al., 2015), an unlabeled dataset of English sentences taken from novels.

Latent Sense Representations. The choice of LSR is critical to the success of the S2S approach. We propose two techniques: **1**) contextual word embedding of target word w and **2**) sentence embedding of s. The contextual word embedding uses a local context to represent target sense information, whereas the sentence embedding uses the entire sentence as context (see Figure 2).

1. Contextual Word Embedding. Pretrained language models have been shown to create high-quality contextual representations of words (Vulić et al., 2020b), which are extracted by averaging the embeddings of the target word tokens at the output of the model. We initially experimented with this approach for computing l using pretrained BERT (Devlin et al., 2019) as the language model. Due to the BERT pretraining objective, we found empirically that the contextual word embedding contains too much information about its surrounding context such that prediction of the next token in the training sentence becomes trivial. The model does not learn

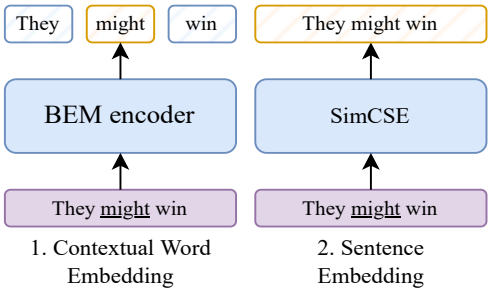

Figure 2: Depiction of both proposed types of LSR. Orange blocks represent the LSR for each method.

the intended task, because generated sentences are almost identical copies of the input. To restrict the information in l only to that of the intended target meaning of w, we use the contextual encoder of a pretrained Bi-Encoder Model (BEM) (Blevins and Zettlemoyer, 2020). BEM is a BERT model fine-tuned on Word Sense Disambiguation (WSD) and thus tries to encode a representation of the contextual meaning of each word in a given sentence. We find empirically that conditioning on this representation produces a training loss that plateaus slightly beneath that of the same model trained without any conditional input, indicating that little information about the surrounding context leaks through and the embedding contains mostly word sense information. To produce the LSR during training, we extract the contextual embedding of w when passing s through BEM. Note that this approach is semi-supervised, because it requires WSD training data[1] during pretraining of BEM. For this reason, we denote our approach using contextual word embeddings for the LSR as S2S_{semi}.

2. Sentence Embedding. We want to explore a fully self-supervised solution to our one-shot format of EM. Since word senses often appear in specific contexts, a semantic representation of the input sentence can be useful for determining the target sense for w in the generated sentence. To encode the sentence-level semantics into a vector that serves as the LSR, we use the unsupervised variant of SimCSE (Gao et al., 2021) to embed s. Since neither our training approach nor the pretraining process of SimCSE or BART require any labeled data, we denote this approach as S2S_{self}.

Generating Examples. To generate a new example sentence with a specific word sense, we provide

[1]The pretrained BEM we use for our experiments is taken from the original paper (Blevins and Zettlemoyer, 2020) and was pretrained using SemCor (Miller et al., 1993), one of the datasets used for training a baseline EM model in this paper.

one source sentence s containing the target word w with the desired sense to our system. The LSR is extracted from s, and w is passed to the BART encoder in isolation. The embedded representations are then concatenated and passed to the BART decoder, where we decode using nucleus sampling (Holtzman et al., 2019), setting $p = 0.5$.

4 Baseline

We reimplement ExMaker (Barba et al., 2021b) ourselves and train it to produce example sentences using one (lemma, definition) pair as input. For comparison to a sense-agnostic system, we additionally train a vanilla version of ExMaker where the input definition is replaced with the empty string. Validation is performed using the same approach proposed in ExMaker (Barba et al., 2021b). The final model chosen is the one that produces target words that match their intended meaning best according to ExMaker's automatic semantic match validation method (see Section 7 for details). For both ExMaker and S2S, we finetune the same pretrained BART[2] model (Lewis et al., 2020).

5 Data

Given that S2S requires no labeled EM data for training, but rather raw text only, we compare to the baseline in settings where evaluation words are either included or held out from training. For the "seen" baseline setting, all validation and test words are included in the training data. For the "unseen" baseline setting, all validation and test words are excluded. We discuss construction of the validation and test word sets in the following sections.

Training. We use the Huggingface version of BookCorpus[3] (Zhu et al., 2015) to train S2S, which consists of 74M sentences. We use SemCor (Miller et al., 1993) and Oxford Dictionary[4] to train ExMaker. SemCor is much smaller than Oxford Dictionary, containing only 33k unique sentences and 200k labeled instances; Oxford Dictionary contains 1.3M examples.

Validation. We use the automatic target word semantic match validation scheme (see Section 7) from ExMaker (Barba et al., 2021b) and thus need to generate text using each saved model checkpoint for a small set of validation words. We randomly choose 100 polysemous words disjoint from our test set for validation and evaluate on all word senses for each validation word.

Test Set. EM is most difficult for words that take on two or more meanings. For this reason, we choose to evaluate words that are known to have more than one distinct meaning, i.e. homographs. For a set of ground truth homographs, we use the Wikipedia Homograph Dataset (Gorman et al., 2018) and choose only those homographs that have the "Lexical" homograph type label. We evaluate only on those word senses that appear in SemCor such that S2S and ExMaker are evaluated on the same set of word senses. This gives us a total of 78 unique words with 334 total senses for our automatic evaluations. We use a manually-curated subset of these words for human evaluations (see Section 8).

6 Experimental Setup

Training and Validation Hyperparameters. We train all S2S and ExMaker models for 500k steps. For S2S, we train with a batch size of 64 and use the 500k model checkpoint (∼one half epoch for BookCorpus). For ExMaker, we train with a batch size of 32 and validate models every 25k steps when training with Oxford Dictionary and every 10k steps when training with SemCor. These baseline hyperparameters were chosen to balance the risk of overfitting (especially with SemCor) and computation time for the validation process. Final baseline models are chosen based on the best validation score. See Appendix A for further details.

Generation. For ExMaker, we provide each (lemma, definition) pair for each word sense in our test set and generate 50 examples per pair. For S2S, we choose one SemCor example sentence per test word sense from which we generate 50 new examples for that word sense (one-shot evaluation).[5]

[2] https://huggingface.co/facebook/bart-base

[3] https://huggingface.co/datasets/bookcorpus - This version of BookCorpus is uncased, which is critical to the success of S2S. When training with cased data, target polysemous words in generated sentences often do not match their intended meaning well. Cased examples in this paper have been truecased using the NLTK and StanfordNLP toolkits.

[4] We use the dataset prepared by He and Yiu (2022), available at https://github.com/NLPCode/CDEG.

[5] For a fair evaluation of S2S, we want the single input example sentence for a given word sense to be of reasonable quality. We found empirically that short input sentences tend to produce less diverse examples (see Section 10), so we first sort example sentences for a given word sense by length. Then, if there is at least one example sentence with length in the range of 70-180 characters, we randomly choose one sentence from the example sentences satisfying that requirement. Otherwise, we randomly choose the example input sentence from whatever is available.

	Training Data	Diversity SB4↓	Diversity B4↓	Sem. Match SM↑	Sem. Match SM_H ↑	Fluency F_H ↑
$S2S_{semi}$	BookCorpus	0.62	0.51	0.44	8.41	8.06
$S2S_{self}$	BookCorpus	0.70	0.57	0.39	<u>6.49</u>	<u>7.15</u>
ExMaker (unseen)	SemCor	0.62	0.41	0.32	-	-
ExMaker (seen)	SemCor	<u>0.91</u>	<u>0.89</u>	**0.52**	-	-
ExMaker (unseen)	Oxford Dict.	0.52	0.41	0.35	**8.67**	**8.70**
ExMaker (seen)	Oxford Dict.	0.52	0.40	0.36	-	-
$ExMaker_V$ (unseen)	Oxford Dict.	**0.48**	0.40	<u>0.26</u>	-	-
$ExMaker_V$ (seen)	Oxford Dict.	**0.48**	**0.39**	0.27	-	-
Gold (SemCor)	NA	0.51	1.00	0.52	9.59	8.74

Table 1: Automatic and human evaluation of semantic match, diversity, and fluency of generated sentences. Columns with missing values are human evaluations (we evaluated on the most interesting subset of approaches (based on automatic evaluations) to make the workload reasonable for our volunteer annotators). Abbreviations are as follows: $ExMaker_V$: ExMaker vanilla (no conditioning on definition), SB4: Self-BLEU-4, B4: BLEU-4, SM: Sense Match, SM_H: Semantic Match (human), F_H: Fluency (human). Human evaluation values are the mean of scores on a scale from 0-10 where results are statistically significant at $p < 0.05$ between all shown methods except gold/ExMaker for F_H and ExMaker/$S2S_{semi}$ for SM_H. Best score is bolded and worst score is underlined (excluding gold data).

Evaluations. We perform both automatic and human evaluations of text generation quality. For automatic analysis, we examine the target word semantic match and the diversity of generated sentences. For human evaluations, we examine overall fluency and target word semantic match.

7 Automatic Evaluations

Semantic Match. We use one automatic analysis technique to evaluate target word semantic match quality, which we call Sense Match (SM). This technique is identical to the ExMaker validation method (Barba et al., 2021b). Explicitly, we extract the contextual embedding of the target word from a pretrained BERT-large[6] model and compute cosine similarity with respect to its word sense embedding from ARES (Scarlini et al., 2020). ARES uses the same BERT-large model when extracting word sense embeddings, so both the target word contextual embeddings and sense embeddings are directly comparable.

Diversity. We want to explore how diverse the generated sentences are across two dimensions: **1**) similarity to other generated sentences, and **2**) similarity to SemCor data.[7] To analyze the diversity of generated sentences from themselves, we use Self-BLEU (Zhu et al., 2018). To analyze the diversity of generated sentences with respect to gold data (SemCor), we compute the BLEU score (Papineni et al., 2002) using generated sentences as hypotheses and SemCor sentences as references. Note that unlike in machine translation, we want BLEU scores (including Self-BLEU) to be as low as possible because we want diverse outputs.

Self-BLEU. For each word sense, we compute the Self-BLEU (Zhu et al., 2018) score using only sentences generated for that particular word sense. For a fair comparison to the gold Self-BLEU score, we limit the number of generated sentences per word sense to the number of SemCor examples for each given sense. The overall Self-BLEU score for a given model is the mean of all word sense Self-BLEU scores.

BLEU. We use this metric to gauge how much the generated sentences are partial copies of SemCor data. For each word sense, we compute the average BLEU score between the generated sentences for that word sense and the SemCor sentences containing that word sense (each generated sentence is a hypothesis and the SemCor sentences are the references). The overall score is the mean of the BLEU scores for each word sense.

8 Human Evaluations

We want to compare the quality of generated sentences from S2S and ExMaker, and gold sentences from SemCor. To evaluate the quality as judged by humans, we recruit five volunteer native-speaker annotators (see Appendix B) to judge both the fluency and target word semantic match of the example sentences. We ask annotators to assign each example sentence a score from zero to ten for each evaluation criterion where ten is the best score.

Dataset Construction. For human evaluations,

[6] https://huggingface.co/bert-large-cased

[7] SemCor is used as training data for ExMaker and as conditioning input for S2S, so it is important to check how different the generated sentences are from those already in SemCor.

Word Sense Information	Generated Sentences
Word: object **Definition**: A tangible and visible entity; an entity that can cast a shadow **Input**: Science is fully competent to deal with any element of experience which arises from an object in space and time.	1. But we can only think of such things as being in any dimension in relation to an object in a higher plane. 2. They could see things that no other person could, as if they were connected to an object in space. 3. We are given the most complete understanding of the existence of space and time by an object in a hologram.
Word: object **Definition**: Express or raise an objection or protest or criticism or express dissent **Input**: Most of these former churches are now used as warehouses, but "neither Anglicans nor nonconformists object to selling churches to Roman Catholics", and have done so.	1. "I have no interest in making a whole town, and none of my parents or any of my grandparents object to creating a new town, but we have no business doing that." 2. "I've worked in the mines for over a year, but no members of the nile nor of the mayan or christian nation object to using the power of the indian sea."
Word: contract **Definition**: A binding agreement between two or more persons that is enforceable by law **Input**: Enter into contracts with educational institutions, scientific organizations, and industrial and engineering firms;	1. They have joint contracts with firms in various departments of government and the defense industry. 2. You have formal contracts with academic and business organizations? 3. You have joint contracts with scientists, directors, managers, scientists, and teachers.
Word: contract **Definition**: Become smaller or draw together **Input**: Dr. Bonnor supports the idea that the universe both expands and contracts, that in several billion years the expansion will slow up and reverse itself and that the contraction will set in.	1. And then, when the universe has become so vast that it contracts, the universe will be able to pick up the energy in a way that isn't so obvious to the eye. 2. We are one hundred percent sure that the cosmos is larger and contracts, but the universe is still the same, and the theory is that the core is not.

Table 2: Examples of generated sentences using S2S_{semi} for distinct senses of the words "object" and "contract."

we use a subset of word senses from our test set. This subset is chosen by hand such that senses are noticeably distinct in meaning for a given word, resulting in a set of 40 word senses.[8] For each model, we generate one sentence per word sense. Each of our five annotators rates the fluency and target word semantic match of every sentence, giving us a total of 200 annotations per evaluation criterion per model.

Fluency and Semantic Match. We ask annotators to rate the fluency of each sentence, i.e. how well it adheres to proper grammar rules on a scale from zero to ten. For the semantic match judgment, we provide annotators with an example sentence, target word, and target definition. We then ask annotators to rate how well the contextual meaning of the target word in the example sentence matches the target definition on a scale from zero to ten.

Inter-Annotator Agreement. We use the Average Mean Inter-Annotator Agreement (AMIAA) metric (Vulić et al., 2020a) to judge how well annotators agree. AMIAA is written explicitly as:

$$\text{AMIAA} = \frac{\sum_i \rho(s_i, \mu_i)}{K}, \text{ where } \mu_{i,n} = \frac{\sum_{j \neq i} s_{j,n}}{K-1} \quad (2)$$

[8] https://github.com/jharvill23/LatentSenseRepresentationsEM

K is the number of annotators and $\rho(s_i, \mu_i)$ is the Spearman's rank correlation between the scores for annotator i and the mean scores from all other annotators. The mean score $\mu_{i,n}$ for each datapoint n is the average of each annotator's score $s_{j,n}$ when leaving out annotator i. For our human evaluations, the AMIAA for fluency and semantic match are 0.55 and 0.56, respectively, indicating moderate agreement.

9 Results

The results for automatic and human evaluations are in Table 1. We discuss the key takeaways below.

Baseline overfits on SemCor. All three automatic evaluation metrics (Self-BLEU, BLEU, Sense Match) indicate that the baseline severely overfits under the seen setting when trained with SemCor, generating only slight modifications of the sentences it was trained on. This results in an unrealistically good Sense Match score and unrealistically poor Self-BLEU and BLEU scores.

Baseline improves with more data. There is noticeable improvement in both Sense Match and Self-BLEU scores for the unseen setting when training with Oxford Dictionary compared to training with SemCor. Also, the baseline seems to generalize better with more data and not suffer from overfitting when training with Oxford Dictionary,

Word: project	**Definition**: A planned undertaking
	Input: They worked on the group project together.
Generated:	1. During the creative project, we worked together.
	2. We spent the day working on the group project.
	Input: It had been months since the two engineers started working on the project together, but unfortunately they had not made much progress.
Generated:	1. The crew was busy working on the preliminary design of the project in the back of the rover.
	2. I hadn't really paid much attention to the project in the past couple of years, so I had no idea how much effort it took to construct the model.
Word: project	**Definition**: Cause to be heard
	Input: The singer knew how to project her voice.
Generated:	1. I was able to manage to project my voice.
	2. She was the perfect person to project her voice.
	Input: The crowd was too large, and he knew if he was going to get help he would need to project his voice all the way up to security personnel at the front of the stage.
Generated:	1. The new bartender in the bar wasn't about to back down, but he was going to project his voice out to the crowd in a way that would get people to listen.
	2. We were just going to talk about the police and how they'd come in, but I was ready to project my voice out of the audience and into the crowd.

Table 3: Comparison of generated sentences using input sentences with different lengths for $S2S_{semi}$.

because the seen setting shows only the slightest improvement over the unseen setting in this case.

Semi-supervised approach performs similarly to baseline. $S2S_{semi}$ demonstrates impressive performance across all automatic and human evaluations, coming close to the baseline for some metrics and surpassing it for others. $S2S_{semi}$ only slightly underperforms the baseline for diversity and fluency metrics, while surpassing the baseline (not including overfit seen setting) for the automatic Sense Match evaluation and performing the same[9] as the baseline for the human semantic match evaluation.

Self-supervised approach underperforms semi-supervised approach. As expected, we see a reduction in performance across all metrics for $S2S_{self}$ compared to $S2S_{semi}$. The differences are relatively small for the automatic evaluations, but are more noticeable for the human evaluations. We see the largest relative drop in performance for the human evaluation of semantic match, indicating that $S2S_{self}$ frequently produces incorrect or vague meanings of target words in its generated sentences. This demonstrates that sentence-level semantics can often indicate the meaning of a target word but that representations of the immediate context are more reliable for one-shot EM.

10 Generation Examples

We provide example sentences generated using our best[10] proposed approach, $S2S_{semi}$, in Tables 2 and 3. Table 2 shows generated examples for two distinct senses each for the words "object" and "contract." Table 3 shows generated examples for two distinct senses of the word "project" using a short and long sentence as input. While the generated examples are fluent, diverse, and use the target word with its intended definition, there are several notable qualitative properties that we hypothesize are a result of information leakage in the contextual word embedding LSR. It appears that some information related to the input sentence's vocabulary, syntax and length is encoded in the LSR, in addition to the desired target word sense information.

Vocabulary, Topical and Syntactical Overlap. There is noticeable overlap between vocabulary and topics between input and output sentences. For example, all three generated sentences for the first sense of "object" in Table 2 revolve around themes present in the input like "science" and "outer space." This is evident through the use of words or phrases like "dimension," "higher plane," "space," and "hologram" that appear in the output. Beyond simple topical or vocabulary overlap, we see longer stretches of identical text with the presence of the 3-gram "space and time" in both the input and the third generated sentence. We also find syntactical overlap between input and output sentences. For example, the third generated sentence for the first sense of "contract" in Table 2 lists entities with which a contract may be held just as the input sentence does.

Dependence on Input Length. The examples in

[9] Difference between $S2S_{semi}$ and ExMaker is not statistically significant for SM_H, see Table 1.

[10] See Appendix C for examples generated by $S2S_{self}$.

Table 3 demonstrate that the length of each generated sentence is similar to that of its respective input sentence. Further, we see that the relative location of the target word in the generated sentence is similar to that in the input sentence, indicating that input sentence length and target word location information is encoded in the LSR.

Increased Diversity with Input Length. When comparing the outputs generated by short and long sentences in Table 3, we see that longer input sentences produce more diverse outputs. This makes sense given the previous observation about the output's dependence on input length, because there are more possible outputs that satisfy the EM constraints when the output is encouraged to be longer. This property is why we choose to use input sentences of length 70-180 characters, when available, for our one-shot evaluations (see Section 6).

11 Future Work

There are several avenues of future work including potential improvements to our proposed one-shot EM approach as well as applications of our technique to downstream tasks.

Self-Supervised Disentanglement of Sense Information. While we demonstrate that it is possible to perform EM in a self-supervised fashion by using sentence embeddings as the LSR, the quality is much lower than the semi-supervised approach that uses the contextual word embedding extracted from the BEM (Blevins and Zettlemoyer, 2020). We believe future research efforts should strive to create disentangled sense representations in a self-supervised fashion that are of similar quality to that of the BEM, instead of relying on WSD training data. One possible approach is to use product quantization as is done in Wav2Vec2.0 (Baevski et al., 2020) on top of the contextual word embedding output by a pretrained BERT model (Devlin et al., 2019). By mapping the continuous representation output by BERT to a set of discrete codebook vectors, it may be possible to create representations that resemble synsets and thus disentangle contextual information unrelated to the target word sense.

Optimizing One-Shot EM for Downstream Tasks via Filtering. Our proposed one-shot approach to EM slightly underperforms the baseline, but only in a statistical sense. We are still able to generate high-quality examples, and generation is fast and computationally cheap. Thus, a dedicated filtering scheme that picks out good examples may prove useful when applying our approach to downstream tasks like data augmentation for WSD or automatic dictionary construction in low-resource languages.

Improving Diversity via Feedback Loop. The generated examples in Tables 2 and 3 demonstrate topical overlap with their respective inputs while still being unique sentences. It may be possible to create more diverse examples starting from one input sentence by feeding the outputs back in as input in a feedback loop. It is likely that target word meaning would drift from its desired definition, but such a tradeoff could be explored in future work.

12 Conclusions

In this paper, we proposed a variant of Exemplification Modeling (EM) that uses an example sentence as input instead of the lemma and target definition. We demonstrated empirically that this format of EM can be performed in a one-shot setting, requiring no labeled data for training. Our approach, S2S, uses a Latent Sense Representation (LSR) as a conditional variable that encourages the generated sentence to use the target word with the intended definition. We proposed two types of LSR and performed extensive evaluation of fluency, diversity, and semantic match in the generated sentences. Additionally, we performed a thorough qualitative analysis that highlights properties such as topical and vocabulary overlap that appear in sentences generated by S2S.

This work is the first paper to analyze the ability of a system to perform EM by conditioning on a latent variable derived from contextual use of the target word instead of a string indicating its definition. While both proposed LSRs have drawbacks, the contextual word embedding LSR leads to impressive results and inspires future improvement. Optimization of the LSR to encode only word sense information in a self-supervised way could lead to improved performance while requiring no labeled data at any point in the LSR pretraining or S2S training process. Given the ability of S2S models to be trained with only raw text, it may be possible to improve performance beyond that of the fully-supervised baseline for a larger set of word senses than those in Oxford Dictionary when conditioning on an LSR that completely disentangles word sense information in a self-supervised fashion.

13 Limitations

Our initial work on one-shot EM shows promising results but comes with several limitations. By inspecting generated examples, we find evidence of information leakage in the LSR that causes generated sentences to be similar in topic and length to the input sentence. We also require one example at inference time in order to generate a new example for a specific word sense. We demonstrated empirically that the baseline does not have this restriction, because the baseline method generalizes well to words that were unseen during training (see Table 1). Additionally, our best approach, $S2S_{semi}$, still requires WSD training data for pretraining of the BEM, which is used for LSR extraction. Finally, the S2S models in this paper have the limitation that the target word looks identical in both the input and output sentence, whereas the baseline is capable of generating examples with various target word forms for a given target lemma. This could be fixed in the future by lemmatizing the target word before passing it to the S2S BART encoder during training and generation, but we did not run this experiment due to our initial focus on using as little supervision as possible to perform EM.

14 Ethics Statement

In this paper, we have focused on exemplification modeling using data (SemCor, Oxford Dictionary, BookCorpus) and models (BART) that may contain biases related to attributes like gender, race, or disability. It is possible that these biases surface in sentences generated using the techniques proposed in this paper (Nadeem et al., 2021; Liang et al., 2021), because we do not actively focus our efforts on the removal of such biases. For this reason, sentences generated by our proposed models or the baseline should be used with caution in order to prevent perpetuation of harmful stereotypes.

References

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems*, 33:12449–12460.

Edoardo Barba, Tommaso Pasini, and Roberto Navigli. 2021a. Esc: Redesigning wsd with extractive sense comprehension. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4661–4672.

Edoardo Barba, Luigi Procopio, Caterina Lacerra, Tommaso Pasini, and Roberto Navigli. 2021b. Exemplification modeling: Can you give me an example, please? In *IJCAI*, pages 3779–3785.

Edoardo Barba, Luigi Procopio, and Roberto Navigli. 2021c. Consec: Word sense disambiguation as continuous sense comprehension. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1492–1503.

Michele Bevilacqua, Marco Maru, and Roberto Navigli. 2020. Generationary or "how we went beyond word sense inventories and learned to gloss". In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7207–7221, Online. Association for Computational Linguistics.

Terra Blevins and Luke Zettlemoyer. 2020. Moving down the long tail of word sense disambiguation with gloss informed bi-encoders. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1006–1017, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Georgiana Dinu, Prashant Mathur, Marcello Federico, and Yaser Al-Onaizan. 2019. Training neural machine translation to apply terminology constraints. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3063–3068, Florence, Italy. Association for Computational Linguistics.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.

Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. 2018. Style transfer in text: Exploration and evaluation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Kyle Gorman, Gleb Mazovetskiy, and Vitaly Nikolaev. 2018. Improving homograph disambiguation with supervised machine learning. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Xingwei He and Siu Ming Yiu. 2022. Controllable dictionary example generation: Generating example sentences for specific targeted audiences. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 610–627.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *arXiv e-prints*, pages arXiv–1904.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Paul Pu Liang, Chiyu Wu, Louis-Philippe Morency, and Ruslan Salakhutdinov. 2021. Towards understanding and mitigating social biases in language models. In *International Conference on Machine Learning*, pages 6565–6576. PMLR.

Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. 2020. CommonGen: A constrained text generation challenge for generative commonsense reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1823–1840, Online. Association for Computational Linguistics.

Ximing Lu, Sean Welleck, Peter West, Liwei Jiang, Jungo Kasai, Daniel Khashabi, Ronan Le Bras, Lianhui Qin, Youngjae Yu, Rowan Zellers, Noah A. Smith, and Yejin Choi. 2022. NeuroLogic a*esque decoding: Constrained text generation with lookahead heuristics. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 780–799, Seattle, United States. Association for Computational Linguistics.

Fuli Luo, Peng Li, Pengcheng Yang, Jie Zhou, Yutong Tan, Baobao Chang, Zhifang Sui, and Xu Sun. 2019. Towards fine-grained text sentiment transfer. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2013–2022.

George A. Miller, Claudia Leacock, Randee Tengi, and Ross T. Bunker. 1993. A semantic concordance. In *Human Language Technology: Proceedings of a Workshop Held at Plainsboro, New Jersey, March 21-24, 1993*.

Moin Nadeem, Anna Bethke, and Siva Reddy. 2021. StereoSet: Measuring stereotypical bias in pretrained language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5356–5371, Online. Association for Computational Linguistics.

Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM computing surveys (CSUR)*, 41(2):1–69.

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

Alessandro Raganato, Jose Camacho-Collados, and Roberto Navigli. 2017. Word sense disambiguation: A unified evaluation framework and empirical comparison. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 99–110.

Bianca Scarlini, Tommaso Pasini, and Roberto Navigli. 2020. With more contexts comes better performance: Contextualized sense embeddings for all-round word sense disambiguation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3528–3539.

Ivan Vulić, Simon Baker, Edoardo Maria Ponti, Ulla Petti, Ira Leviant, Kelly Wing, Olga Majewska, Eden Bar, Matt Malone, Thierry Poibeau, Roi Reichart, and Anna Korhonen. 2020a. Multi-SimLex: A large-scale evaluation of multilingual and crosslingual lexical semantic similarity. *Computational Linguistics*, 46(4):847–897.

Ivan Vulić, Edoardo Maria Ponti, Robert Litschko, Goran Glavaš, and Anna Korhonen. 2020b. Probing pretrained language models for lexical semantics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7222–7240, Online. Association for Computational Linguistics.

Pengcheng Yang, Lei Li, Fuli Luo, Tianyu Liu, and Xu Sun. 2019. Enhancing topic-to-essay generation with external commonsense knowledge. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2002–2012.

Lili Yao, Nanyun Peng, Ralph Weischedel, Kevin Knight, Dongyan Zhao, and Rui Yan. 2019. Plan-and-write: Towards better automatic storytelling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7378–7385.

Yue Zhang and Stephen Clark. 2015. Discriminative syntax-based word ordering for text generation. *Computational linguistics*, 41(3):503–538.

Jianing Zhou and Suma Bhat. 2021. Paraphrase generation: A survey of the state of the art. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5075–5086.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1097–1100.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *The IEEE International Conference on Computer Vision (ICCV)*.

A Computational Details

We run experiments across two NVIDIA RTX 3090 Ti GPUs, but each individual experiment runs on one GPU (we train two models simultaneously, one on each GPU). The pretrained BART, BEM context encoder, and SimCSE models have 141M, 109M, and 109M parameters, respectively. It takes approximately 1.5 days to train each S2S or ExMaker model.

B Human Evaluations

The annotators are from the United States and are acquaintances of the authors, but are not from the same research group. Consent was obtained via email, and annotators were made fully aware that annotations would only be used for this paper to demonstrate differences in model performance (see Table 1).

C Generation Examples for Self-Supervised Approach

We provide examples of generated sentences from S2S_{self} in Table 4. There appears to be information leakage in the sentence embedding LSR, with some slight differences compared to the contextual word embedding LSR. We discuss the key qualitative takeaways below.

Vocabulary and Topical Overlap. We notice the same trend of overlapping vocabulary and topics that we observe in sentences generated by S2S_{semi}, although the overlap appears more pronounced for S2S_{self}. For example, for the first sense of "object" in Table 4, the subject of the input sentence, "science", appears as the subject in both generated sentences. In Table 2 (S2S_{semi}), we see for the same respective sense that the generated sentences revolve around scientific themes, but do not explicitly make use of the word "science" itself.

Dependence on Input Length. The length of generated sentences for S2S_{self} are similar to that of their respective input sentences, as is the case for sentences generated by S2S_{semi}. The key distinction between S2S_{self} and S2S_{semi} in this respect is that the target word location does not appear to be encoded in the LSR for S2S_{self}. For example, the two sentences generated for the first sense of "object" in Table 4 use the target word in the first half and the very end of the sentence, respectively, while the target word appeared towards the end of the input sentence.

Incorrect Target Word Meaning. We see two examples of the target word adopting the wrong meaning in Table 4 (incorrect target word usages are italicized). In both cases, the intended part-of-speech is verb, but the example sentences have used the target word as a noun. We hypothesize such mistakes are due to imperfect representation of the target word sense via sentence-level semantics, but further analysis is needed to confirm this. In cases where the target word meaning is unclear based on sentence-level semantics, the model may produce a sentence exemplifying the most frequent sense of the target word.

Word Sense Information	Generated Sentences
Word: object **Definition**: A tangible and visible entity; an entity that can cast a shadow **Input**: Science is fully competent to deal with any element of experience which arises from an object in space and time.	1. Science itself requires a scientific object to take practical, practical, and comprehending knowledge of the worlds that arise from it. 2. The field of science requires a great deal of experience to approach the subject matter as it exists in any form or object.
Word: object **Definition**: Express or raise an objection or protest or criticism or express dissent **Input**: Most of these former churches are now used as warehouses, but "neither Anglicans nor nonconformists object to selling churches to Roman Catholics", and have done so.	1. "However some of the latter have been converted into churches, and are now used as storage areas for goods or goods only, although most object to the practices of the church." 2. "Some of the churches remain empty or have not been converted into warehouses, although they are generally in use as auction boxes as a tourist *object*."
Word: contract **Definition**: A binding agreement between two or more persons that is enforceable by law **Input**: Enter into contracts with educational institutions, scientific organizations, and industrial and engineering firms;	1. Open up; engage in the creation of scientific and technical contracts; engage in various academic and industrial programs; employ engineers and teachers; obtain federal and state grants; and so forth. 2. Industrial companies must engage in employment, through partnership contracts, in private laboratories, and so forth.
Word: contract **Definition**: Become smaller or draw together **Input**: Dr. Bonnor supports the idea that the universe both expands and contracts, that in several billion years the expansion will slow up and reverse itself and that the contraction will set in.	1. He argues that the natural climate theory, which will take effect over the next two decades, creates massive gravitational pulls and contracts with time, expanding and expanding in a way that we both know will be so. 2. He explained that the general theory believes that the natural process of expansion will grow between decades, if not centuries, and that the economic climate will squeeze out both forces in a heartbeat and collapse out of the shrinking spaces between *contracts*.

Table 4: Examples of generated sentences using $S2S_{self}$ for distinct senses of the words "object" and "contract." The target word is italicized in examples where it takes on a different meaning from what was intended.