AMR-DA: Data Augmentation by Abstract Meaning Representation

Ziyi SHOU[1] **Yuxin JIANG**[2] **Fangzhen LIN** [1]
[1]Department of Computer Science and Engineering
[2]DSA Thrust, Information Hub
The Hong Kong University of Science and Technology
{zshou,flin}@cse.ust.hk, yjiangcm@connect.ust.hk

Abstract

Abstract Meaning Representation (AMR) is a semantic representation for NLP/NLU. In this paper, we propose to use it for data augmentation in NLP. Our proposed data augmentation technique, called AMR-DA, converts a sample sentence to an AMR graph, modifies the graph according to various data augmentation policies, and then generates augmentations from graphs. Our method combines both sentence-level techniques like back translation and token-level techniques like EDA (Easy Data Augmentation). To evaluate the effectiveness of our method, we apply it to the English tasks of semantic textual similarity (STS) and text classification. For STS, our experiments show that AMR-DA boosts the performance of the state-of-the-art models on several STS benchmarks. For text classification, AMR-DA outperforms EDA and AEDA and leads to more robust improvements.[1]

1 Introduction

Data augmentation (DA) techniques automatically generate additional data from existing data set for training machine learning models. They are widely used in computer vision (see, e.g. Perez and Wang, 2017) and can boost the performance of the trained models.

In NLP, DA methods can be roughly classified into token-level ones and sentence-level ones (Chen et al., 2021). Token-level DA methods generate new sample sentences from the original ones by changing some of their tokens (words). They include the method in Zhang et al. (2015) that replaces some random tokens by their synonyms using a thesaurus, the now widely used Easy Data Augmentation (EDA) methods in Wei and Zou (2019) that allow some random token insertion, deletion and swaps, and the more recent one in Liu et al. (2020) that performs token replacement using their embeddings. One advantage of these token-level DA methods is that they are easy to implement. However, they can sometimes generate ill-formed or incoherent sentences as they do not take the sentence structures into account. In contrast, sentence-level methods generate new sample sentences by modifying the whole original sentences. They typically work by having an encoder that converts the input sentence to an intermediate representation and a decoder that generates new sentences from the intermediate representations. For example, in back translation (Sennrich et al., 2016), the intermediate representation is a sentence in another natural language. In generation methods (Kumar et al., 2020; Yang et al., 2020), the intermediate representation is a hidden state. One advantage of sentence-level DA methods is that they can preserve the semantics of the sentences. A major limitation of current sentence-level DA methods is that there is not much variation in the generated sentences as the intermediate representations used are not easily controllable (Li et al., 2021). For example, modifying the sentences in back translation requires knowledge of other languages, and minor changes of hidden states severely increase training difficulty.

In this paper, we propose a new DA method called AMR-DA that uses the Abstract Meaning Representation (AMR, Banarescu et al., 2013) as the intermediate language. AMR is a well-known semantic meaning representation. It aims to remove syntactic idiosyncrasies and to represent the semantic structure of a sentence as a rooted, directed graph. It works well as an intermediate language for data augmentation as it allows us to combine the token-level and sentence-level methods in a single framework. Like the sentence-level method, our method encodes the entire sentence as an AMR graph. Like the token-level methods, our method manipulates AMR graphs at the node

[1]Codes will be at https://github.com/zzshou/amr-data-augmentation

Findings of the Association for Computational Linguistics: ACL 2022, pages 3082 - 3098

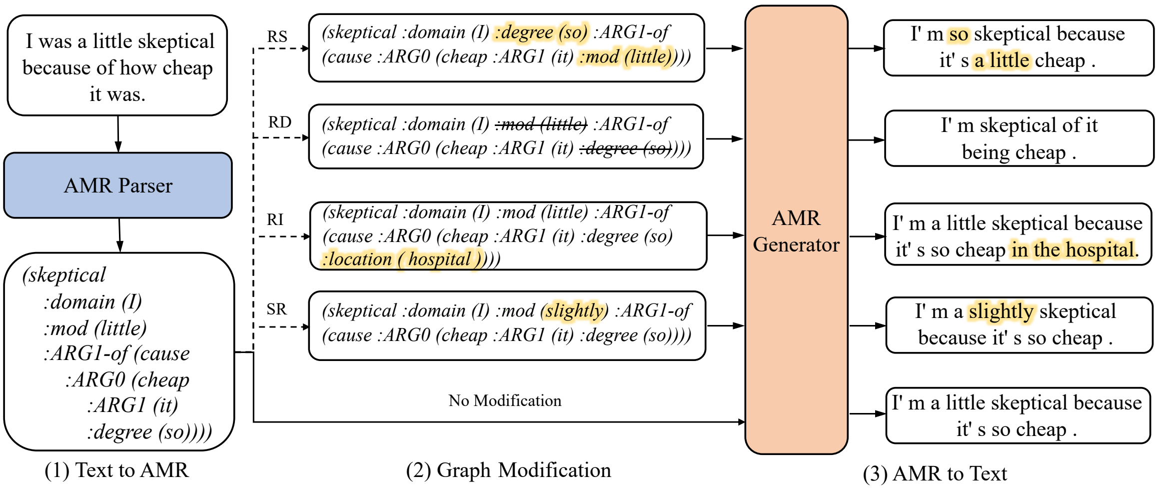


Figure 1: Overview of AMR-DA pipeline: (1) Text to AMR: the AMR parser captures the meaning of the input sentence and transduces it to an AMR graph. (2) Graph Modification: the fundamental choice is not to modify the AMR graph to preserve the entire semantics. Inspired by EDA (Wei and Zou, 2019), we apply four strategies to diversify the graph. RS: random swap; RD: random deletion; RI: random insertion; SR: synonym replacement. (3) AMR to Text: the AMR generator synthesizes sentences from AMR graphs.

(token) level. Thus our method can augment the original sample sentence in various ways without the need to retrain the decoder. This overcomes a key weakness of the current sentence-level methods. Figure 1 shows an overview of our AMR-DA: AMR parser first transduces the sentence into an AMR graph, followed by an AMR graph extender to diversify graphs with different augmentation strategies; finally, the AMR generator synthesizes augmentations from AMR graphs.

To demonstrate the effectiveness of our method, we evaluated AMR-DA on two downstream tasks, semantic textual similarity (STS) and text classification tasks. Experimental results show that our methods boosted unsupervised contrastive learning models to achieve new state-of-the-art results on several benchmarks in STS tasks and outperformed EDA and AEDA in text classification tasks.

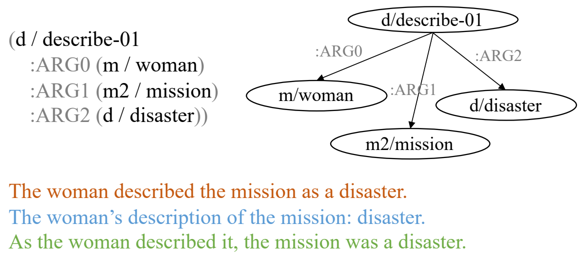


Figure 2: Three sentences with varied surface syntax share the same AMR. Textual and graphical representations are equal.

2 AMR-DA

2.1 Background

Abstract Meaning Representations (AMRs, Banarescu et al., 2013) are designed to abstract away from syntactic idiosyncrasies by encoding the concepts of the sentences into nodes and the relations between concepts into directed edges. They are represented as rooted, labeled graphs textually in PENMAN notation (Goodman, 2020) or graphically. Sentences with identical basic meanings are assigned to the same AMR graph. Figure 2 shows that three sentences with varied surface syntax share the same AMR. In AMR, variables are introduced for entities, events, properties, and states. For example, "d", "m" in the figure are variables. "d/describe-01" refers to an instance d of the AMR concept "describe-01". "describe" is the frame from Propbank (Kingsbury and Palmer, 2002) and "-01" is the sense of frame. AMR concepts can also be English words such as "woman". When an entity plays multiple roles in a sentence, we re-use the corresponding variable in graph notation, called reentrancy. The phrases begin with ":" are relations in AMR graphs. ":ARG0", ":ARG1", ":ARG2" are frame arguments, following PropBank conventions. AMR contains approximately 100 relations, in addition to the edges mentioned in the example, there are general semantic relations ("age", ":location"), relations for quantities (":quant") and relations for date-entities (":month", ":season"), etc.

2.2 AMR Parsing

AMR parser is the first component of AMR-DA (Figure 1). AMR parsing is the task of understanding the sentence and then transducing it to AMR graphs. Lack of explicit alignments between AMR nodes and tokens brings obstacles to AMR parsing. Previous AMR parsers always include complex and fine-grained pre- and post-processing processes. It is very brittle to extend and apply in other tasks. With the help of pretrained language models, sequence-to-sequence (seq2seq) methods win a continual growth of interests. This paper adopts SPRING[2] (Bevilacqua et al., 2021), which achieves state-of-the-art performance on AMR parsing, as our AMR parser. SPRING also implemented the generator in their work, however, we adopt another generator with better performance alternatively introduced in section 2.4.

SPRING first linearized AMR graphs to sequences through DFS-based PENMAN annotation. Nevertheless, when using seq2seq models, a lack of a clear distinction between variables and concepts may cause confusion. Considering that AMR variables have no semantics, SPRING proposed to use special tokens `<R0>, <R1>, ..., <Rn>` to represent variables in the linearization graph and to handle co-referring nodes. They also abandoned the redundant slash token "/". Under this setting, AMR graph in Figure 2 became: `(<R0>` describe-01 :ARG0 (`<R1>` woman) :ARG1 (`<R2>` mission) :ARG2 (`<R3>` disaster)). Adjacency information was still preserved in the linearization process.

After linearizing AMR graphs, SPRING extended a pretrained model, BART (Lewis et al., 2020) which is a transformer-based encoder-decoder model. In order to make BART vocabularies suitable for AMR, they added relations and frames frequently occurring in the training data and initialized the vectors as the average of words embeddings. The results from the seq2seq model need only slight post-processing to transfer sequences to standard PENMAN notations. Details can be found in SPRING paper (Bevilacqua et al., 2021). AMR-DA adopts the model which achieves state-of-the-art performance on AMR 2.0 as AMR parser.

2.3 AMR Graph Modification

Discreteness in languages is the obstacle to transferring data augmentation methods from vision to NLP. Token-level methods attempt to apply modifications on tokens but ignore the entire structure of sentences. However, modifications in sentence-level methods always increase the difficulty of training. The benefit of AMR-DA is that intermediate AMR graphs can be modified through low-cost operations to obtain diverse augmentations; meanwhile, AMR generator will adjust the entire structure of sentences. We shift operations in EDA to AMR graphs. Following EDA, we introduce α to control the percentage of data that operations in AMR-DA will modify.

Keep Original (Ori) The fundamental choice is to preserve the entire intermediate AMR graph. In this way, AMR-DA will generate paraphrased text for the input sentence.

Random Swap (RS) Traditionally, RS operation randomly chooses words and swaps their positions. However, randomly swapping concepts may impact the performance of AMR generator. In Figure 1, if we want to swap positions of "I" and "so" in the original AMR graph, the final graph becomes ":domain (so)" and ":degree (I)" which are not expected to appear in a regular AMR graph. Therefore, we swap concepts and their immediately adjacent edges at the same time. More specifically, we swap edge-node pairs ":degree (so)" and ":domain (I)" instead of tokens. There are two types of effect: if swapping nodes are not siblings, RS operation would change the graph structure, while sibling nodes swapping changes the linearization sequence instead of the graph structure. For one augmentation, RS repeats n times the operation of randomly selecting two edge-node pairs and swapping their positions where $n = max(1, \alpha \times |\text{edge-node pairs}|)$. $|\text{edge-node pairs}|$ means the number of edge-node pairs.

Random Deletion (RD) Instead of removing concepts, we randomly delete concepts with their adjacent edges to guarantee that the rest of graph has necessary components. To control the effects on the AMR graph, RD only applies to leaf nodes. Non-leaf nodes with descendants will possibly have a severe impact on original AMR graphs. For one augmentation, RD repeats random leaf deletion n times where $n = max(1, \alpha \times |\text{edges-node pairs}|)$.

Random Insertion (RI) RI inserts edge-node pairs instead of concepts to preserve the rationality of AMR graph. We collect edge-node pairs (leaves) from AMR 2.0 training data and filter un-

[2] `https://github.com/SapienzaNLP/spring`

suitable pairs based on their edges. For example, ":polarity -" which converts the polarity of semantics, is discarded in RI operation. More examples are listed in Appendix A. For one augmentation, RI randomly inserts n pairs where $n = max(1, \alpha \times |\text{edge-node pairs}|)$.

Synonym Replacement (SR) SR only cares about concepts for that AMR edges are well-designed in AMR. In the linearized graph, we filter tokens that begin with ":" and parentheses, randomly select other tokens, and replace them with one of their synonyms correspondingly. SR randomly replace n concepts where $n = max(1, \alpha \times |\text{concepts}|)$. We substitute similar words according to PPDB synonym (Pavlick et al., 2015). The substitution function is included in nlpaug[3].

2.4 AMR Generation

AMR generation generates sentences from the AMR graph, which is the inverse task of AMR parsing. Pretrained transformer-based architectures gradually dominate the development trend of generators (Mager et al., 2020; Bevilacqua et al., 2021). Ribeiro et al.[4] proposed a generator based on pretrained language models (PLMs-generator) and added extra task-adaptive pretraining. Compared with SPRING, PLMs-generator simplifies PENMAN annotations without adding special tokens as pointers. They examined and compared two PLMs, BART and T5 (Raffel et al., 2019). PLMs-generator continued task-specific pretraining using language model adaptation (LMA) or supervised task adaptation (STA) training with silver data they collected. Details can be found in the paper (Ribeiro et al., 2021). The default AMR generator in our experiments is based on T5-base.

3 Experiments

We conduct experiments on two NLP tasks, semantic textual similarity tasks and text classification tasks, to evaluate our augmentation method.

3.1 Semantic Textual Similarity Tasks

Semantic textual similarity deals with determining how similar two pieces of sentences are. Recently, contrastive learning has become an influential formalism for unsupervised sentence representation, based on the idea of concentrating similar samples and pushing apart dissimilar samples in the vector space (Chen et al., 2020). That is, given a set of paired sentences $\mathcal{D} = \left\{(x_i, x_i^+)\right\}_{i=1}^m$ where x_i and x_i^+ are semantically related, we regard x_i^+ as "positive" of x_i and other sentences in the same mini-batch as "negatives". Let $\mathbf{h}_i$ and $\mathbf{h}_i^+$ denote the representations of x_i and x_i^+, then the training objective for a mini-batch of size N is:

$$\ell_i = -\log \frac{\exp^{sim(\mathbf{h}_i, \mathbf{h}_i^+)} / \tau}{\sum_{j=1}^{N} \exp^{sim(\mathbf{h}_i, \mathbf{h}_j^+)} / \tau}$$

where τ is a temperature hyperparameter and $sim(\mathbf{h}_1, \mathbf{h}_2)$ is the cosine similarity function.

Data augmentation, as the central issue in unsupervised contrastive learning, is utilized to construct "positive pairs". SimCSE (Gao et al., 2021) puts one sentence through pretrained model twice with varied standard dropout masks inside transformers as a minimal form of data augmentation. Although it performs quite well, there still exists a large margin between unsupervised and supervised models. Here we propose a hypothesis that an effective data augmentation in this task requires distinct syntax but related semantics. For this reason, we use AMR-DA as data augmentation to construct positive instances.

3.1.1 Experimental Settings

To verify the effectiveness of AMR-DA, we choose recently proposed models unsup-ConSERT (Yan et al., 2021) and unsup-SimCSE (Gao et al., 2021), which are referred as ConSERT and SimCSE for simplification, as our baseline models. We only replace the original data augmentation methods inside the two models with AMR-DA.

We evaluate on seven STS datasets including STS 2012–2016 (Agirre et al., 2012, 2013, 2014, 2015, 2016), STS Benchmark (Cer et al., 2017) and SICK-Relatedness (Marelli et al., 2014) and report Spearman's correlation.

Following ConSERT, we use a mixture of unlabeled texts from seven STS datasets as training data and average the token embeddings at the last two layers as the sentence embedding. Following SimCSE, we use 1-million sentences randomly sampled from English Wikipedia as training data and adopt the [CLS] representation with an MLP layer on top of it as the sentence embedding. More training details could be found in Appendix B.

[3] https://github.com/makcedward/nlpaug
[4] https://github.com/UKPLab/plms-graph2text

Model	Avg.
BERT_{base}†	63.84
+token augmentations (ConSERT)†	72.74
+AMR-RS augmentation	76.11
+AMR-RD augmentation	74.34
+AMR-RI augmentation	75.31
+AMR-SR augmentation	75.68
+AMR-Ori augmentation	**76.14**

Table 1: Performance comparison of models with different AMR-DA operations. †: results from Yan et al., 2021.

3.1.2 Main Results

The first question is which operation we should choose for contrastive learning in the STS task. Table 1 shows the comparison on different augmentation strategies. ConSERT considered cutoff and shuffle token augmentations while we replaced their DA with AMR-DA. The results show that all operations in AMR-DA outperform ConSERT with token augmentations. Since we use AMR-DA to construct positive pairs for STS model training, Table 1 presents that AMR-Ori generates augmentations more similar to the original sentences than other operations. To access the diversity of augmented data, we adopt F1 measured between two bags of words as lexical overlap score. A higher lexical overlap F1 indicates more overlap between augmented data and original sentences and less diversity. Table 2 provides the summary statistics for various operations of AMR-DA.

AMR Operation	Ori	RS	RD	RI	SR
Overlap F1	0.554	0.531	0.476	0.510	0.449

Table 2: Overlap F1 score of AMR-DA operations.

Table 3 shows the main results, where the highest numbers among models with the same pretrained encoder are highlighted in bold. Only changing the data augmentation module in ConSERT and SimCSE to AMR-DA, the performance could be boosted substantially to the state-of-the-art. AMR-ConSERT obtains absolute improvements of 3.40 and 1.74 on BERT_{base} and BERT_{large} respectively compared with the original ConSERT that utilizes feature cutoff and shuffle on tokens as DA methods. While AMR-SimCSE outperforms SimCSE significantly on BERT_{base} (1.70 ↑), BERT_{large} (1.22 ↑), RoBERTa_{base} (1.86 ↑) and RoBERTa_{large} (0.80 ↑). We also make a comparison between our models and current state-of-the-art model ESimCSE (Wu et al., 2021), which uses word repetition to construct positive pairs and momentum contrast to expand negative pairs. Experimental results indicate that AMR-SimCSE surpasses ESimCSE on BERT_{large} (0.33 ↑), RoBERTa_{base} (0.99 ↑) and RoBERTa_{large} (0.25 ↑). If we discard momentum contrast in ESimCSE and only compare the effectiveness of DA methods, AMR-SimCSE (77.95) outperforms ESimCSE (77.43) on BERT_{base}.

Model	**Avg.**
unsup-ConSERT Setups	
ConSERT-BERT_{base}†	72.74
AMR-ConSERT-BERT_{base}	**76.14** (+3.40)
ConSERT-BERT_{large}†	76.45
AMR-ConSERT-BERT_{large}	**78.19** (+1.74)
unsup-SimCSE Setups	
SimCSE-BERT_{base}‡	76.25
+ back translation	71.71
ESimCSE-BERT_{base}§	**78.27**
- momentum contrast	77.43
AMR-SimCSE-BERT_{base}	77.95 (+1.70)
SimCSE-BERT_{large}‡	78.41
ESimCSE-BERT_{large}§	79.31
AMR-SimCSE-BERT_{large}	**79.63** (+1.22)
SimCSE-RoBERTa_{base}‡	76.57
ESimCSE-RoBERTa_{base}§	77.44
AMR-SimCSE-RoBERTa_{base}	**78.43** (+1.86)
SimCSE-RoBERTa_{large}‡	78.90
ESimCSE-RoBERTa_{large}§	79.45
AMR-SimCSE-RoBERTa_{large}	**79.70** (+0.80)

Table 3: The average sentence embedding performance on seven STS test sets, in terms of Spearman's correlation. †: results from Yan et al., 2021. ‡: results from Gao et al., 2021; ; §: results from Wu et al., 2021. Models begin with "AMR" are the models with AMR-DA.

In addition, we implemented SimCSE with back translation based on WMT'19 English-German translation models (Ng et al., 2019) as the DA method. We use random sampling for decoding as recommended by (Edunov et al., 2018a), and set the temperature to 0.8. Other training settings are the same as those of SimCSE. As shown in Table 3, back translation is inferior to AMR-DA in STS tasks. The possible reason is that augmentations with limited diversity are hard to improve pretrained models.

3.2 Text Classification Tasks

Text classification tasks are widely studied in many real applications, such as document categorization, email spam filtering, etc. The performance of machine learning methods in this task always depends on the quality of training data. How to use DA techniques to improve machine learning systems attracts a number of studies (Wang and Yang, 2015; Wei and Zou, 2019; Liu et al., 2020; Karimi et al., 2021). AMR-DA is partly inspired by EDA, which explores text editing techniques for data augmentation. EDA performs SR, RI, RS, or RD operations on tokens, whereas AMR-DA performs these DA strategies on AMR graphs. In order to answer whether DA strategies on AMR graphs perform better than on tokens, we conduct a fair assessment on EDA and AMR-DA. In addition, to show the effectiveness of AMR-DA, we take AEDA (Karimi et al., 2021), another strong DA, into comparison.

3.2.1 Experimental Settings

We conduct experiments on four benchmark datasets: Standford Sentiment Treebank (SST-2, Socher et al., 2013); Customer Reviews Dataset (CR, Hu and Liu, 2004; Liu et al., 2015b), Subjectivity/Objectivity Dataset (SUBJ, Pang and Lee, 2004); Pros and Cons Dataset (PC, Ganapathibhotla and Liu, 2008). The detailed statistics are listed in Table F.5.

We chose Recurrent Neural Network (RNN, Liu et al., 2016), Convolutional Neural Network (CNN, Kim, 2014) and BERT (Devlin et al., 2019) as backbone models.

Data selection module has been modified to be close to application scenarios in real life. We select proportions of original training data and then add the corresponding augmentations for that only visible data can be extended. Experimental setups are identical to all DA methods. All experiments are run with five different random seeds and reported as average performance. Training details are in Appendix C.

3.2.2 Main Results

We ran CNN, RNN and BERT across all four datasets using three DA methods. First, we added one augmented sentence for each instance to assess the effectiveness of single augmentation. We reported the average performance of all different operations in EDA and AMR-DA as final one augmentation performance. As the top part of Table 4 shows, the average improvement of AMR-DA on three models is 0.55%, which is 0.91% better than EDA and 0.14% better than AEDA, respectively. How about using all operations to augment data in the training process? To answer this question, we added each operation augmentations together in AMR-DA and trained models with all five augmentations. Correspondingly, we randomly selected five augmentations using AEDA and EDA operations. We reported the average performance in the bottom part of Table 4. AMR-DA achieved 0.83% performance gain with five augmentations better than one augmentation, which means our operations brought diversified information to improve models. Regarding the effectiveness of operations (SR, RI, RS and RD), we made a detailed comparison on EDA and AMR-DA. Figure 3 shows that AMR-DA outperforms EDA remarkably on various fractions of the training set.

Model	CNN	RNN	BERT	Avg.
Original	88.15	86.49	93.19	89.28
With 1 augmentation				
+EDA	87.29	86.16	93.39	88.92
+AEDA	88.30	87.59	93.19	89.69
+AMR-DA	**88.40**	**87.63**	**93.47**	**89.83**
With 5 augmentations				
+EDA	87.75	86.37	93.29	89.14
+AEDA	88.78	87.21	93.53	89.84
+AMR-DA	**88.80**	**88.00**	**93.54**	**90.11**

Table 4: Average performance of CNN, RNN and BERT trained on original, EDA, AEDA and AMR-DA (with 1 or 5 augmentations for each instance) data across all datasets.

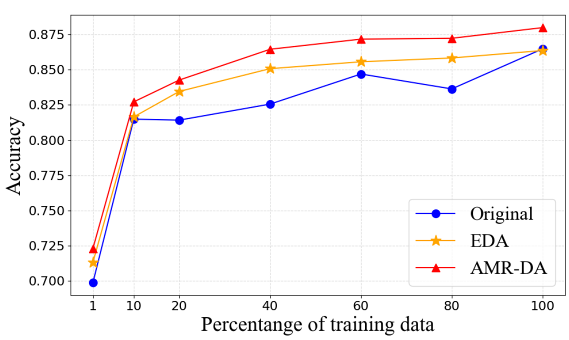


Figure 3: Average performance of RNN model trained on different proportions of original, EDA and AMR-DA training data for four datasets.

4 Analysis

Effect of AMR Generators From the introduction in Section 2.1, paraphrased sentences correspond to the identical AMR graph. In other words, AMR graph to sentences is a one-to-many relationship. Since there is no uniform evaluation of AMR generators, it is necessary to study the impact of AMR generators on the performance of AMR-DA. We compared AMR-Ori with various generators based on BART$_{base}$, T5$_{small}$ and T5$_{base}$. Table 5 shows comparison on PLMs-generators. We found that pretrained models with larger sizes are capable of generating better quality augments. So we choose AMR generator with T5$_{base}$ as final generator in AMR-DA.

Model	Avg.
BERT$_{base}$-flow‡	66.55
SimCSE-BERT$_{base}$‡	76.25
AMR(BART$_{base}$ generator)-SimCSE	77.81
AMR(T5$_{small}$ generator)-SimCSE	77.65
AMR(T5$_{base}$ generator)-SimCSE	**77.95**

Table 5: Performance of AMR-DA (Ori) in STS tasks with various generators.‡: results from Gao et al., 2021 ;§: results from Wu et al., 2021.

Why does AMR-DA work in STS task? To answer this question, we use alignment and uniformity, which are proposed by (Wang and Isola, 2020) to measure the quality of representations. Alignment calculates how close the positive instances stay, while uniformity evaluates how uniformly the random instances are scattered on the hypersphere. For both metrics, *lower numbers are better*. We take the checkpoint of SimCSE and AMR-SimCSE every 10 steps during training (100 steps in total) and visualize the alignment and uniformity computed on STSB dataset. Figure 4 demonstrates that both SimCSE and AMR-SimCSE improve the uniformity steadily. Additionally, AMR-SimCSE provides a continuously decreasing alignment. It verifies our hypothesis that data augmentation with different syntax but highly related semantics results in better sentence embeddings.

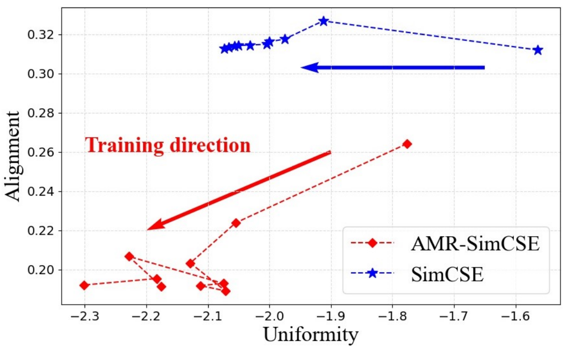


Figure 4: Alignment-uniformity plot on STSB dataset.

Analysis of Generated Outputs To analyze generated outputs by back-translation and AMR-Ori, we use supervised SimCSE-RoBERTa$_{large}$, which achieves the state-of-the-art performance on various semantic textual similarity benchmarks, to compute the sentence embedding cosine similarity between the generated sentences and the original ones. Figure 5 summarizes the results. First we can see

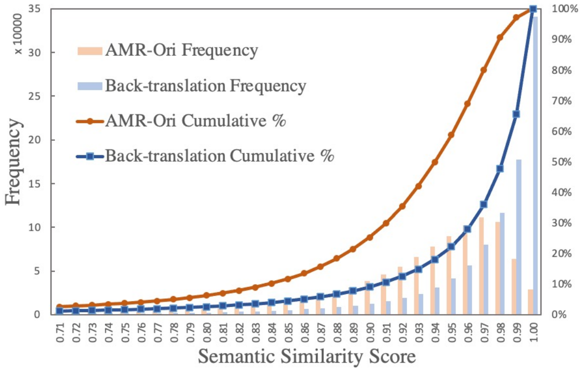


Figure 5: Semantic similarity scores of back-translation and AMR-Ori augmentations (data from Table 3).

that for both AMR-Ori and back-translation, their generated sentences have high similarity scores with the original sentences. However, AMR-Ori generates much more diversified outputs. For back-translation, more than 30% of the generated sentences have the similarity score of 1.0 (highest) with their original sentences, and more than 50% of them have the similarity score of 0.99 or above. While AMR-Ori is more uniform. The highest frequency rate, about 10%, is at the similarity score of 0.97.

We also computed the F1 scores measured between two bags of words. We find that the overlap score of back-translation method is 0.760, compared to 0.566 for AMR-Ori (evaluated using unsupervised SimCSE experiment data in Table 3).

For illutration, we list some examples of back-translation and AMR-Ori in Table 6 and more in Table D.3 in the appendix. One could see that back-translation paraphrases source sentences with little

Source	IDS Tirana is a football club based in Tirana, Albania.
Back Translation	IDS Tirana is a football club from Tirana, Albania.
AMR-Ori	The football club IDS Tirana is based in Tirana, Albania.
Source	The library was established through the philanthropy of Martha Bayard Stevens.
Back Translation	The library was founded through the philanthropy of Martha Bayard Stevens.
AMR-Ori	Martha Bayard Stevens philanthropy has established a library.
Source	A meeting of promoters was also held at Presbyterian Church.
Back Translation	A meeting of the project promoters was also held in the Presbyterian Church.
AMR-Ori	The promoters also held a meeting at the Presbyterian Church.

Table 6: Augmented examples generated by back-translation and AMR-Ori (no edits on intermediate AMR graphs) from source sentences.

modification. On the other hand, AMR-Ori can produce quite different sentences even though it does not modify the intermediate representations. A key factor is that AMR graphs abstract away from syntactic idiosyncrasies while retain semantic frame arguments.

Finally, Table D.4 in the appendix lists some example outputs from EDA and AMR-DA. The original sentence is the same as EDA-None. Except for between EDA-None and AMR-Ori, AMR-DA generated outputs are more fluent than their corresponding outputs by EDA.

5 Related Work

Our proposed data augmentation method is based on manipulating AMR graphs. Similar tree-edit techniques on syntax trees have been found to be useful in paraphrases generation (Heilman and Smith, 2010; Vila and Dras, 2012). Other applications of AMR have also been based on graph manipulation. For example, Liu et al. (2015a) used AMR in summarization task by first parsing the source text to a set of graphs, transforming it to a summary graph, and then generating a summary using the summary graph. Sachan and Xing (2016) represented text and questions as AMR graphs and reduced the machine comprehension problem to a graph containment problem. We have seen a growing body of work that makes use of AMR in other applications such as dialogue modeling, information extraction and commonsense reasoning (Bai et al., 2021; Zhang et al., 2021; Lim et al., 2020).

Based on the influence scope of augmentation, related data augmentation methods can be roughly classified into token-level and sentence-level methods (Chen et al., 2021).

In token-level, synonyms replacement, random swap, random insertion, random deletion (Zhang et al., 2015; Wei and Zou, 2019) have been proven to improve the performance in classification tasks. In STS task, plenty of data augmentation techniques have been utilized such as shuffling, cutoff (Yan et al., 2021), synonyms replace (Wang et al., 2021), word repetition (Wu et al., 2021), etc. However, these methods all risk impairing structure information, resulting in incoherent augmentations.

In contrast, sentence-level take the whole sentence into consideration. Widely used back translation (Sennrich et al., 2016; Edunov et al., 2018b; Qu et al., 2021) translates sentences into intermediate languages and then translates back. Some studies attempt to incorporate syntactic information (Chen et al., 2019) or latent variables (Gupta et al., 2018) to guide generators synthesize various augmentations. But these methods significantly increase the training difficulty. AMR-DA uses AMR as an intermediate language, which can modify graphs as easily as in token-level methods, and synthesizes high-quality and diversified augmentations without grinding in training.

6 Conclusion and Future Work

We propose a novel data augmentation method called AMR-DA. AMR-DA transduces sentences to AMR graphs, applies multiple strategies to modify graphs, and then generates diversified augmentations. To the best of our knowledge, this paper is the first work that utilizes AMR for data augmentation. AMR-DA overcomes the deficiency of previous sentence-level generation methods and diversifies augmentations without retraining decoders. Our experiments show that AMR-DA boosts the performance of models to achieve state-of-the-art results in several STS benchmarks and outperforms

EDA and AEDA in text classification tasks. In this paper, we mainly use AMR-DA to generate positive augmentations. Further research could use AMR-DA to carefully construct adversarial samples for specific tasks and.

References

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel M. Cer, Mona T. Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Iñigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, German Rigau, Larraitz Uria, and Janyce Wiebe. 2015. Semeval-2015 task 2: Semantic textual similarity, english, spanish and pilot on interpretability. In *Proceedings of the 9th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2015, Denver, Colorado, USA, June 4-5, 2015*, pages 252–263. The Association for Computer Linguistics.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel M. Cer, Mona T. Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2014. Semeval-2014 task 10: Multilingual semantic textual similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation, SemEval@COLING 2014, Dublin, Ireland, August 23-24, 2014*, pages 81–91. The Association for Computer Linguistics.

Eneko Agirre, Carmen Banea, Daniel M. Cer, Mona T. Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2016. Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2016, San Diego, CA, USA, June 16-17, 2016*, pages 497–511. The Association for Computer Linguistics.

Eneko Agirre, Daniel M. Cer, Mona T. Diab, and Aitor Gonzalez-Agirre. 2012. Semeval-2012 task 6: A pilot on semantic textual similarity. In *Proceedings of the 6th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2012, Montréal, Canada, June 7-8, 2012*, pages 385–393. The Association for Computer Linguistics.

Eneko Agirre, Daniel M. Cer, Mona T. Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. *sem 2013 shared task: Semantic textual similarity. In *Proceedings of the Second Joint Conference on Lexical and Computational Semantics, *SEM 2013, June 13-14, 2013, Atlanta, Georgia, USA*, pages 32–43. Association for Computational Linguistics.

Xuefeng Bai, Yulong Chen, Linfeng Song, and Yue Zhang. 2021. Semantic representation for dialogue modeling. 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 4430–4445, Online. Association for Computational Linguistics.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*, pages 178–186.

Michele Bevilacqua, Rexhina Blloshmi, and Roberto Navigli. 2021. One spring to rule them both: Symmetric amr semantic parsing and generation without a complex pipeline. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12564–12573.

Daniel M. Cer, Mona T. Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation, SemEval@ACL 2017, Vancouver, Canada, August 3-4, 2017*, pages 1–14. Association for Computational Linguistics.

Jiaao Chen, Derek Tam, Colin Raffel, Mohit Bansal, and Diyi Yang. 2021. An empirical survey of data augmentation for limited data learning in nlp. *arXiv preprint arXiv:2106.07499*.

Mingda Chen, Qingming Tang, Sam Wiseman, and Kevin Gimpel. 2019. Controllable paraphrase generation with a syntactic exemplar. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5972–5984.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR.

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.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018a. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 489–500. Association for Computational Linguistics.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018b. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium. Association for Computational Linguistics.

Murthy Ganapathibhotla and Bing Liu. 2008. Mining opinions in comparative sentences. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 241–248.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Michael Wayne Goodman. 2020. Penman: An open-source library and tool for AMR graphs. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, ACL 2020, Online, July 5-10, 2020*, pages 312–319. Association for Computational Linguistics.

Ankush Gupta, Arvind Agarwal, Prawaan Singh, and Piyush Rai. 2018. A deep generative framework for paraphrase generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Michael Heilman and Noah A Smith. 2010. Tree edit models for recognizing textual entailments, paraphrases, and answers to questions. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 1011–1019.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177.

Akbar Karimi, Leonardo Rossi, and Andrea Prati. 2021. Aeda: An easier data augmentation technique for text classification.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Paul Kingsbury and Martha Palmer. 2002. From TreeBank to PropBank. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02)*, Las Palmas, Canary Islands - Spain. European Language Resources Association (ELRA).

Varun Kumar, Ashutosh Choudhary, and Eunah Cho. 2020. Data augmentation using pre-trained transformer models. In *Proceedings of the 2nd Workshop on Life-long Learning for Spoken Language Systems*, pages 18–26. Association for Computational Linguistics.

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.

Bohan Li, Yutai Hou, and Wanxiang Che. 2021. Data augmentation approaches in natural language processing: A survey. *arXiv preprint arXiv:2110.01852*.

Jungwoo Lim, Dongsuk Oh, Yoonna Jang, Kisu Yang, and Heui-Seok Lim. 2020. I know what you asked: Graph path learning using amr for commonsense reasoning. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2459–2471.

Fei Liu, Jeffrey Flanigan, Sam Thomson, Norman Sadeh, and Noah A. Smith. 2015a. Toward abstractive summarization using semantic representations. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1077–1086, Denver, Colorado. Association for Computational Linguistics.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016. Recurrent neural network for text classification with multi-task learning. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, page 2873–2879. AAAI Press.

Qian Liu, Zhiqiang Gao, Bing Liu, and Yuanlin Zhang. 2015b. Automated rule selection for aspect extraction in opinion mining. In *Twenty-Fourth international joint conference on artificial intelligence*.

Sisi Liu, Kyungmi Lee, and Ickjai Lee. 2020. Document-level multi-topic sentiment classification of email data with bilstm and data augmentation. *Knowledge-Based Systems*, 197:105918.

Manuel Mager, Ramón Fernandez Astudillo, Tahira Naseem, Md Arafat Sultan, Young-Suk Lee, Radu Florian, and Salim Roukos. 2020. GPT-too: A language-model-first approach for AMR-to-text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1846–1852, Online. Association for Computational Linguistics.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, Roberto Zamparelli, et al. 2014. A sick cure for the evaluation of compositional distributional semantic models. In *Lrec*, pages 216–223. Reykjavik.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook FAIR's WMT19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 314–319, Florence, Italy. Association for Computational Linguistics.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 271–278.

Ellie Pavlick, Pushpendre Rastogi, Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2015. Ppdb 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 425–430.

Luis Perez and Jason Wang. 2017. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*.

Yanru Qu, Dinghan Shen, Yelong Shen, Sandra Sajeev, Weizhu Chen, and Jiawei Han. 2021. Coda: Contrast-enhanced and diversity-promoting data augmentation for natural language understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Leonardo FR Ribeiro, Martin Schmitt, Hinrich Schütze, and Iryna Gurevych. 2021. Investigating pretrained language models for graph-to-text generation. In *Proceedings of the 3nd Workshop on Natural Language Processing for Conversational AI*.

Mrinmaya Sachan and Eric Xing. 2016. Machine comprehension using rich semantic representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 486–492, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96.

Richard Socher, John Bauer, Christopher D Manning, and Andrew Y Ng. 2013. Parsing with compositional vector grammars. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 455–465.

Marta Vila and Mark Dras. 2012. Tree edit distance as a baseline approach for paraphrase representation. *Procesamiento del lenguaje natural*, 48:89–95.

Dong Wang, Ning Ding, Piji Li, and Haitao Zheng. 2021. CLINE: Contrastive learning with semantic negative examples for natural language understanding. 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 2332–2342, Online. Association for Computational Linguistics.

Tongzhou Wang and Phillip Isola. 2020. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 9929–9939. PMLR.

William Yang Wang and Diyi Yang. 2015. That's so annoying!!!: A lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviors using# petpeeve tweets. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2557–2563.

Jason Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. 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 6382–6388.

Xing Wu, Chaochen Gao, Liangjun Zang, Jizhong Han, Zhongyuan Wang, and Songlin Hu. 2021. Esimcse: Enhanced sample building method for contrastive learning of unsupervised sentence embedding. *CoRR*, abs/2109.04380.

Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. 2021. Consert: A contrastive framework for self-supervised sentence representation transfer. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, Virtual Event, August 1-6, 2021*, pages 5065–5075. Association for Computational Linguistics.

Yiben Yang, Chaitanya Malaviya, Jared Fernandez, Swabha Swayamdipta, Ronan Le Bras, Ji-Ping Wang, Chandra Bhagavatula, Yejin Choi, and Doug

Downey. 2020. Generative data augmentation for commonsense reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1008–1025, Online. Association for Computational Linguistics.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28:649–657.

Zixuan Zhang, Nikolaus Parulian, Heng Ji, Ahmed Elsayed, Skatje Myers, and Martha Palmer. 2021. Fine-grained information extraction from biomedical literature based on knowledge-enriched abstract meaning representation. 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 6261–6270.

A Discarding pairs in RI operation

We filter pairs based on edge properties. The discarding edges are listed in the following table.

Edge	Reasons
:ARGn	potential ambiguity of arguments
:polarity	convert the polarity of semantics
:wiki	Unsuitable for most graphs
:op$n$	Unsuitable for most graphs
:snt$n$	Unsuitable for most graphs
:value	Unsuitable for most graphs

B STS tasks Training Details

For AMR-SimCSE, grid-search of batch size $\in \{64, 96, 128, 160\}$ and learning rate $\in$ {5e-6, 1e-5, 3e-5, 5e-5} is carried out on STS-B development set, and the hyperparameter settings are listed in Table B.1. The dropout rate is set to 0.1 for base models and 0.15 for large models. We use the temperature $\tau = 0.05$ for all the experiments. During training, we found that a larger maximum sequence length equal to 96 benefits our AMR-SimCSE, while in SimCSE the value is 32. So we also enlarge the maximum sequence length to 96 in SimCSE but do not observe any improvement.

	BERT		RoBERTa	
	base	large	base	large
Batch size	96	128	160	96
Learning rate	3e-5	3e-5	5e-5	5e-6

Table B.1: Hyperparameters for AMR-SimCSE.

For AMR-ConSERT, we use hyperparameter settings that are the same as the original paper.

C Text Classification Training Details

For CNN models, we followed the architecture in EDA and modified filters. The entire architecture of our CNN: input layer; the concatenation of 1D convolutional layer of 128 filters of size 3, 4 and 5 with global 1D max pool layer for each convolutional layer; dropout layer with $\rho = 0.2$; dense layer of 20 hidden units with ReLU activation function, softmax output layer. Other CNN settings and RNN settings are identical to EDA. As for BERT experiments, we adopt base, uncased version BERT as backbone and the Adam optimizer (Kingma and Ba, 2015) with an initial learning rate of 2e-5. We pick the best checkpoint according to the validation loss. Random seeds are from 0 to 4. The default alpha setting for 4 operations are listed in the following table:

	RS	RD	RI	SR
α	0.05	0.1	0.05	0.1

Table C.2: Setting of α for four different operations.

D Comparison on Data Augmentation Outputs

More examples on generated outputs from back-translation and AMR-Ori are presented in Table D.3. Augmented examples using EDA and AMR-DA are presented in Table D.4.

E Effect of alpha in Augmentation Operations

We test each of operations individually for different training set sizes to determine their ability with α=0.05, 0.1, 0.2, 0.3, 0.4, 0.5. For each value, we randomly synthesized two augmentations and ran CNN models in this experiment. In Figure 6, all operations in AMR-DA contribute to performance gain. On average, operations achieve more significant gains in smaller datasets.

F Detailed Experimental Results

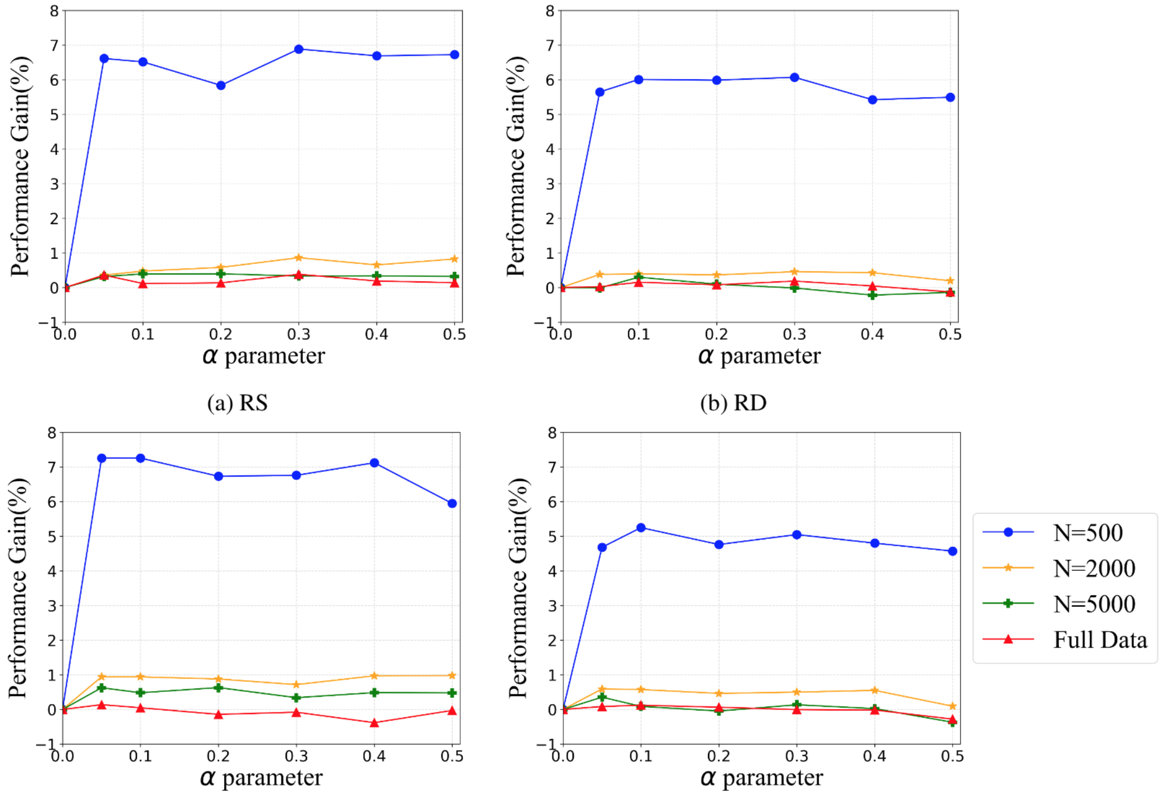
Table F.6 and F.7 are detailed versions of Table 3. Table F.8 is the detailed version of Table 5. Table F.9 is the detailed version of Table 1. Table F.10 is the detailed version of Table 4.

Source	IDS Tirana is a football club based in Tirana, Albania.
BT	IDS Tirana is a football club from Tirana, Albania.
AMR-Ori	The football club IDS Tirana is based in Tirana, Albania.
Source	The library was established through the philanthropy of Martha Bayard Stevens.
BT	The library was founded through the philanthropy of Martha Bayard Stevens.
AMR-Ori	Martha Bayard Stevens philanthropy has established a library.
Source	A meeting of promoters was also held at Presbyterian Church.
BT	A meeting of the project promoters was also held in the Presbyterian Church.
AMR-Ori	The promoters also held a meeting at the Presbyterian Church.
Source	He died suddenly on his way home from work on 23 December 1970.
BT	On December 23, 1970, he died suddenly on his way home from work.
AMR-Ori	On 23 December 1970, when he went home from work, he suddenly died.
Source	Supported by a senior leadership team he assembled he took the organization from near insolvency to financial security and a higher level of service delivery.
BT	Supported by a management team he assembled, he led the organization from near bankruptcy to financial security and improved service delivery.
AMR-Ori	With the support of his assembled senior leadership team, he took the organization from near non-financial security to higher levels of service delivery.
Source	Malaika Arora, Geeta Kapoor, and Terence Lewis is going to Judge of Sony TV’s dance reality show India’s Best Dancer.
BT	Malaika Arora, Geeta Kapoor and Terence Lewis will be the judges of Sony TV’s dance reality show India’s Best Dancer.
AMR-Ori	Malaika Arora, Geeta Kapoor and Terence Lewis are judges for Sony TV ’ s dance reality show Best Dancer.
Source	The Yurts lay the foundation for the whole philosophy of family relationships to which nomadic societies have always attached significant importance.
BT	The yurts form the basis of the whole philosophy of family relations, to which nomadic societies have always attached great importance.
AMR-Ori	The whole philosophy of family relationships, which nomad societies always attach significant importance, was laid by the Yurts.
Source	From then on, I went through different adventures and endangered my life many times.
BT	From then on, I experienced various adventures and was in danger of my life many times.
AMR-Ori	From then on, I have gone through different adventures, and have put my life in danger many times.
Source	Comedian Bharti Singh will Host this show along with her husband writer Haarsh Limbachiyaa.
BT	Comedian Bharti Singh will host the show with her husband, writer Haarsh Limbachiyaa.
AMR-Ori	This show will be hosted by comedian Bharti Singh’s husband, writer Haarsh limbachiyaa.

Table D.3: Sentences generated using back-translation and using AMR-Ori. BT: back-translation

Operation	EDA	AMR-DA
None	A sad, superior human comedy played out on the back roads of life.	The superior human sad comedy plays out on the back road of life.
SR	A **lamentable**, superior human comedy played out on the **backward** road of life.	A **top** human **regrettable** comedy plays out on the backroads of life .
RI	A sad, superior human comedy played out on **funniness** the back roads of life.	The superior human sad comedy **of warmth** plays out on the back road of life.
RS	A sad, superior human comedy played out on **roads** back **the** of life.	The superior human **back** comedy plays out on the **sad** road of life .
RD	A sad, superior human ~~comedy played~~ out on the ~~back~~ roads of life.	The sad ~~superior~~ human comedy plays out on the ~~back~~ road of life .
None	the solid filmmaking and convincing characters makes this a high water mark for this genre.	Solid filmthings and convincing characters make this a high - watermark for these genera.
SR	the solid filmmaking and **convert** characters makes this a high water mark for this genre	Solid **motion pictures** and convincing characters make these high - watermarks for this genre.
RI	**in high spirits** the solid filmmaking and convincing characters makes this a high water mark for this genre.	This solid, **entertaining** filmthings, and convincing character, makes a high water mark for this genre.
RS	the solid filmmaking and convincing characters makes this a high water mark this genre **for**	This is a high water mark for this genre , with convincing characters and solid films.
RD	the solid ~~filmmaking~~ and convincing characters makes this a ~~high~~ water mark for this genre	Solid films~~making~~ and ~~convincing~~ characters make a high water mark for this genre.
None	in addition, his album bat out of hell stayed nine years on the english charts, and sold more than 40 million copies worldwide.	And his album, Bat Out of Hell, has stayed on the English charts for 9 years, and sold more than 40 million copies worldwide.
SR	in addition, his album **lick** out of hell stayed niner years on the english charts and sold more than 40 million **replicate** worldwide.	And his album "Bat Out of Hell" has stayed on the charts in **England** for 9 years and sold more than 40 million copies worldwide.
RI	**holdup delay** in addition, his **more than** album bat out of hell stayed nine years on the english charts, and sold more than 40 million copies worldwide.	And his album, Bat Out of Hell, has stayed on the charts in England **correctly** for 9 years, and sold more than 40 million copies worldwide .
RS	**the** addition, his album bat out of hell stayed nine years on **in** english charts and sold **copies** than million **more** worldwide.	And his album, Bat out of Hell, stayed at **more than 40 million copies** for 9 years, and sold worldwide **on the chart in England**.
RD	in addition, his album bat out of hell stayed nine years on the english charts, ~~and~~ sold more than ~~40~~ million copies worldwide.	And his album, Bat Out of Hell, has stayed on the English charts **for a long time**, selling more than 40 million copies worldwide.

Table D.4: Sentences generated using EDA and using our data augmentation method AMR-DA. EDA returns the input sentence with "None" operation, while AMR-DA returns a paraphrased sentence. SR: synonym replacement. RI: random insertion. RS: random swap. RD: random deletion.



(a) RS (b) RD

(c) RI (d) SR

Figure 6: Average performance gain of individual AMR-DA operations over four text classification datasets for different training set sizes. α roughly controls the range that the operation can impact in each augmentation.

Dataset	# Classes	# Train samples	# Test samples	Average length	Vocabulary size
SST-2	2	7,791	1,821	19	15,771
CR	2	4,068	451	19	9,048
SUBJ	2	9,000	1,000	25	22,715
PC	2	40,000	26,090	7	26,090

Table F.5: Statistics of four text classification datasets.

Model	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
ConSERT-BERT$_{base}$†	64.64	78.49	69.07	79.72	75.95	73.97	67.31	72.74
AMR-ConSERT-BERT$_{base}$	**71.98**	**81.96**	**72.91**	**82.00**	**76.31**	**77.00**	**70.85**	**76.14**
ConSERT-BERT$_{large}$†	70.69	82.96	74.13	82.78	76.66	77.53	70.37	76.45
AMR-ConSERT-BERT$_{large}$	**73.93**	**85.45**	**76.27**	**82.86**	**77.87**	**79.28**	**71.65**	**78.19**

Table F.6: The performance comparison of ConSERT with AMR-ConSERT in the unsupervised setting. We report Spearman correlation magnified by a factor of 100 on all splits of seven STS datasets. †: results from Yan et al., 2021.

Model	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
SimCSE-BERT$_{base}$‡	68.40	82.41	74.38	80.91	78.56	76.85	72.23	76.25
ESimCSE-BERT$_{base}$§	**73.40**	83.27	73.83	82.66	78.81	**80.17**	**72.30**	**78.27**
AMR-SimCSE-BERT$_{base}$	72.51	**83.40**	**75.91**	**83.35**	**79.70**	78.94	71.86	77.95
SimCSE-BERT$_{large}$‡	70.88	84.16	76.43	84.50	79.76	79.26	73.88	78.41
ESimCSE-BERT$_{large}$§	73.21	**85.37**	**77.73**	84.30	78.92	**80.73**	**74.89**	79.31
AMR-SimCSE-BERT$_{large}$	**75.47**	84.77	77.56	**85.49**	**80.06**	80.28	73.81	**79.63**
SimCSE-RoBERTa$_{base}$‡	70.16	81.77	73.24	81.36	**80.65**	80.22	68.56	76.57
ESimCSE-RoBERTa$_{base}$§	69.90	82.50	74.68	**83.19**	80.30	**80.99**	70.54	77.44
AMR-SimCSE-RoBERTa$_{base}$	**74.80**	**82.67**	**75.42**	82.57	80.49	80.36	**72.70**	**78.43**
SimCSE-RoBERTa$_{large}$‡	72.86	83.99	75.62	84.77	**81.80**	81.98	71.26	78.90
ESimCSE-RoBERTa$_{large}$§	73.20	**84.93**	76.88	84.86	81.21	**82.79**	72.27	79.45
AMR-SimCSE-RoBERTa$_{large}$	**74.35**	84.72	**77.32**	**85.90**	81.77	81.07	**72.76**	**79.70**

Table F.7: The performance comparison of unsupervised SimCSE and its varients on seven STS test splits. The reported score is Spearman correlation magnified by a factor of 100. ‡: results from Gao et al., 2021; §: results from Wu et al., 2021.

Model	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
Using STS unlabeled texts								
BERT$_{base}$-flow†	63.48	72.14	68.42	73.77	75.37	70.72	63.11	69.57
ConSERT-BERT$_{base}$†	64.64	78.49	69.07	79.72	75.95	73.97	67.31	72.74
+AMR-SR augmentation	71.33	78.37	71.99	83.34	75.24	76.89	72.62	75.68
+AMR-RD augmentation	64.31	80.69	71.87	81.73	**76.76**	75.78	69.28	74.34
+AMR-RI augmentation	67.40	79.24	71.35	**82.56**	76.07	**77.31**	**73.22**	75.31
+AMR-RS augmentation	**72.01**	**82.19**	**72.94**	81.93	76.15	77.24	70.31	76.11
+AMR-Ori augmentation	71.98	81.96	72.91	82.00	76.31	77.00	70.85	**76.14**

Table F.8: Performance comparison of models with different DA methods. †: results from Yan et al., 2021.

Model	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
Using Wiki texts								
BERT$_{base}$-flow‡	58.40	67.10	60.85	75.16	71.22	68.66	64.47	66.55
SimCSE-BERT$_{base}$‡	68.40	82.41	74.38	80.91	78.56	76.85	72.23	76.25
+word repetition§	69.79	**83.43**	75.65	82.44	79.43	**79.44**	71.86	77.43
+back translation	66.50	74.53	66.34	76.61	77.33	72.15	68.54	71.71
+AMR(BART$_{base}$ generator)-SimCSE	72.30	83.15	75.53	83.17	79.23	78.15	**73.16**	77.81
+AMR(T5$_{small}$ generator)-SimCSE	72.26	81.77	**75.93**	**83.44**	**79.78**	77.93	72.44	77.65
+AMR(T5$_{base}$ generator)-SimCSE	**72.51**	83.40	75.91	83.35	79.70	78.94	71.86	**77.95**

Table F.9: Performance comparison of AMR-DA (Ori) with different generators. ‡: results from Gao et al., 2021; §: results from Wu et al., 2021.

	CR	SST2	SUBJ	PC	Avg.
RNN	79.38	82.32	91.96	92.31	86.49
+EDA (num_aug=1)	80.95	82.04	91.43	90.22	86.16
+AEDA (num_aug=1)	**82.22**	82.86	92.56	92.70	87.59
+AMR-DA (num_aug=1)	81.70	**83.37**	**92.68**	**92.76**	**87.63**
+EDA (num_aug=5)	80.93	82.99	91.14	90.42	86.37
+AEDA (num_aug=5)	80.53	83.10	92.62	92.59	87.21
+AMR-DA (num_aug=5)	**82.93**	**83.74**	**92.72**	**92.60**	**88.00**
CNN	83.68	84.28	91.84	**92.79**	88.15
+EDA (num_aug=1)	82.90	83.62	91.51	90.79	87.20
+AEDA (num_aug=1)	83.55	84.50	**92.48**	92.65	88.30
+AMR-DA (num_aug=1)	**83.85**	**84.68**	92.38	**92.70**	**88.40**
+EDA (num_aug=5)	83.59	84.12	91.90	91.40	87.75
+AEDA (num_aug=5)	84.75	**85.11**	**92.68**	92.59	88.78
+AMR-DA (num_aug=5)	**85.05**	84.94	92.54	**92.67**	**88.80**
BERT	89.67	90.72	96.38	**95.98**	93.19
+EDA (num_aug=1)	**90.73**	**91.22**	95.88	95.74	93.39
+AEDA (num_aug=1)	90.15	90.42	96.26	**95.94**	93.19
+AMR-DA (num_aug=1)	90.53	90.90	**96.52**	95.92	**93.47**
+EDA (num_aug=5)	89.80	**91.76**	95.70	95.88	93.29
+AEDA (num_aug=5)	90.01	91.71	96.50	95.89	93.53
+AMR-DA (num_aug=5)	**90.47**	91.02	**96.70**	**95.97**	**93.54**

Table F.10: Average performance of CNN, RNN and BERT on four classification datasets.