Data Augmentation Approaches in Natural Language Processing: A Survey

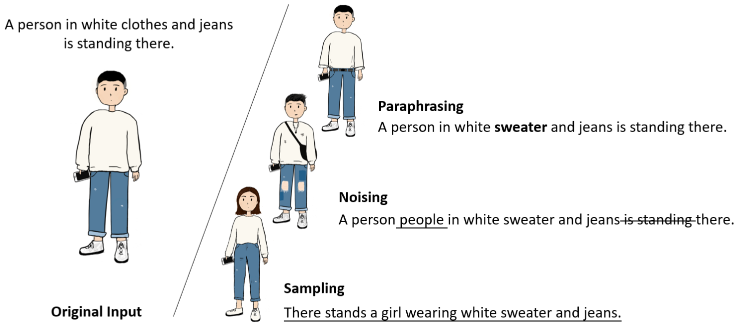
Bohan Li, Yutai Hou, Wanxiang Che*

Harbin Institute of Technology, Harbin, China

Abstract

As an effective strategy, data augmentation (DA) alleviates data scarcity scenarios where deep learning techniques may fail. It is widely applied in computer vision then introduced to natural language processing and achieves improvements in many tasks. One of the main focuses of the DA methods is to improve the diversity of training data, thereby helping the model to better generalize to unseen testing data. In this survey, we frame DA methods into three categories based on the **diversity** of augmented data, including paraphrasing, noising, and sampling. Our paper sets out to analyze DA methods in detail according to the above categories. Further, we also introduce their applications in NLP tasks as well as the challenges.

Keywords: Data Augmentation, Natural Language Processing
2010 MSC: 00-01, 99-00


A person in white clothes and jeans is standing there.
Original Input
Paraphrasing
A person in white **sweater** and jeans is standing there.
Noising
A person people in white sweater and jeans ~~is standing~~ there.
Sampling
There stands a girl wearing white sweater and jeans.

*Corresponding author

Email addresses: bhli@ir.hit.edu.cn (Bohan Li), ythou@ir.hit.edu.cn (Yutai Hou), car@ir.hit.edu.cn (Wanxiang Che)

Preprint submitted to Journal of LaTeX Templates *October 6, 2021*

Contents

1 Introduction 4

2 Data Augmentation Methods in NLP 6

2.1 Paraphrasing-based Methods . . . 7
2.1.1 Thesauruses . . . 7
2.1.2 Semantic Embeddings . . . 8
2.1.3 Language Models . . . 9
2.1.4 Rules . . . 10
2.1.5 Machine Translation . . . 11
2.1.6 Model Generation . . . 12

2.2 Noising-based Methods . . . 13
2.2.1 Swapping . . . 13
2.2.2 Deletion . . . 15
2.2.3 Insertion . . . 15
2.2.4 Substitution . . . 15
2.2.5 Mixup . . . 16

2.3 Sampling-based Methods . . . 17
2.3.1 Rules . . . 18
2.3.2 Seq2Seq Models . . . 18
2.3.3 Language Models . . . 20
2.3.4 Self-training . . . 20
2.4 Analysis . . . 21

3 Strategies and Tricks 23
3.1 Method Stacking . . . 23
3.2 Optimization . . . 24
3.2.1 The Use of Augmented Data . . . 24
3.2.2 Hyperparameters . . . 24
3.2.3 Training Strategies . . . 24
3.2.4 Training Objects . . . 25
3.3 Filtering . . . 25

4 Applications on NLP Tasks 26

5 Related Topics 30
5.1 Pretrained Language Models . . . 30
5.2 Contrastive Learning . . . 30
5.3 Other Data Manipulation Methods . . . 30
5.4 Generative Adversarial Networks . . . 30
5.5 Adversarial Attacks . . . 31

6 Challenges and Opportunities 31

7 Conclusion **32**

1. Introduction

Data augmentation refers to methods used to increase the amount of data by adding slightly modified copies of already existing data or newly created synthetic data from existing data.[1] Such methods alleviate data scarcity scenarios where deep learning techniques may fail, so DA has received active interest and demand recently. Data augmentation is widely applied in the field of computer vision [1], such as flipping and rotation, then introduced to natural language processing (NLP). Different to images, natural language is discrete, which makes the adoption of DA methods more difficult and underexplored in NLP.

Large numbers of DA methods have been proposed recently, and a survey of existing methods is beneficial so that researchers could keep up with the speed of innovation. Liu et al. [2] and Feng et al. [3] both present surveys that give a bird's eye view of DA for NLP. They directly divide the categories according to the methods. These categories thus tend to be too limited or general, e.g., *back-translation* and *model-based techniques*. Bayer et al. [4] post a survey on DA for text classification only. In this survey, we will provide an inclusive overview of DA methods in NLP. One of our main goals is to show the nature of DA, i.e., *why data augmentation works*. To facilitate this, we category DA methods according to the **diversity** of augmented data, since improving training data diversity is one of the main thrusts of DA effectiveness. We frame DA methods into three categories, including paraphrasing, noising, and sampling.

Specifically, *paraphrasing*-based methods generate the paraphrases of the original data as the augmented data. This category brings limited changes compared with the original data. *Noising*-based methods add more continuous or discrete noises to the original data and involve more changes. *Sampling*-based methods master the distribution of the original data to sample new data as augmented data. With the help of artificial heuristics and trained models, such methods can sample brand new data rather than changing existing data and therefore generate even more diverse data.

Our paper sets out to analyze DA methods in detail according to the above categories. In addition, we also introduce their applications in NLP tasks as well as the challenges. The rest of the paper is structured as follows:

- Section 2 presents a comprehensive review of the three categories and analyzes every single method in those categories. We also compare the methods in several aspects like external knowledge, granularity, etc.

- Section 3 refers to a summary of common skills in DA methods to improve the quality of augmented data, including method stacking, optimization, and filtering strategies.

- Section 4 analyzes the application of the above methods in NLP tasks. We also show the development of DA methods on several specific tasks.

[1] `https://en.wikipedia.org/wiki/Data_augmentation`

- Section 5 introduces some related topics of data augmentation, including pre-trained language models, contrastive learning, similar data manipulation methods, generative adversarial networks, and adversarial attacks. We aim to connect data augmentation with other topics and show their difference at the same time.
- Section 6 lists some challenges we observe in NLP data augmentation, including theoretical narrative and generalized methods. These points also reveal the future development direction of data augmentation.
- Section 7 concludes the paper.

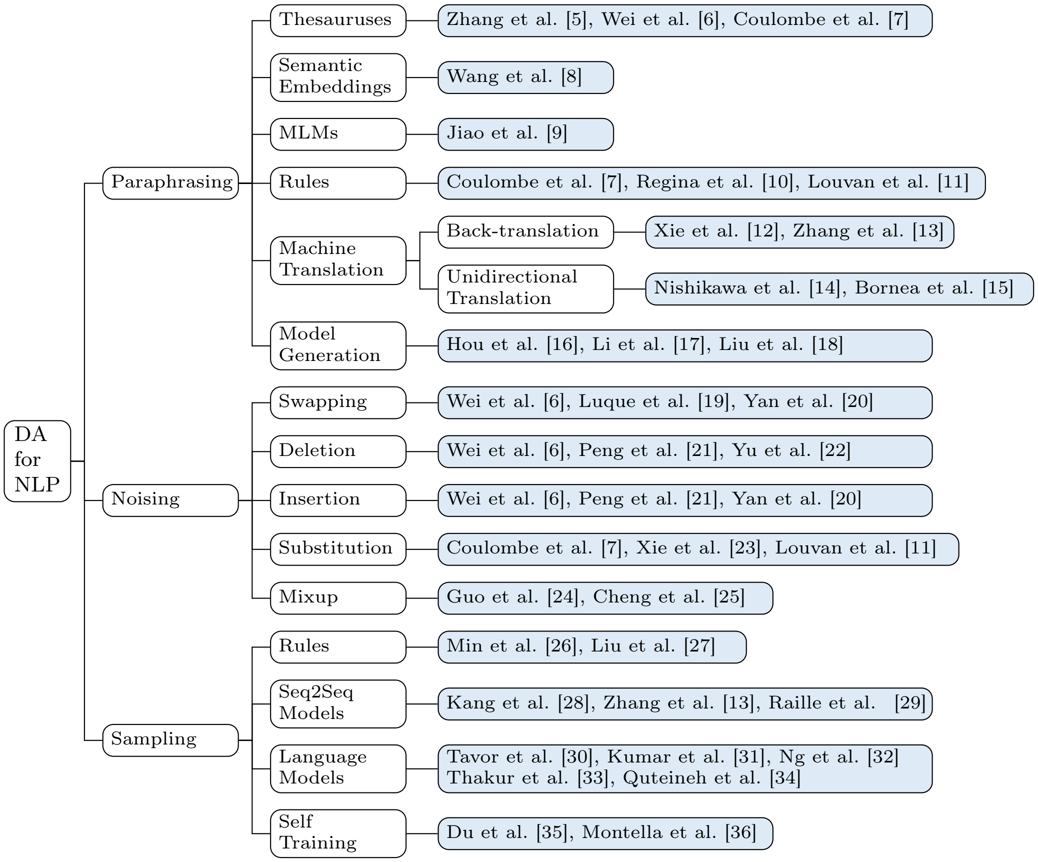

DA for NLP
Paraphrasing
Thesauruses
Zhang et al. [5], Wei et al. [6], Coulombe et al. [7]
Semantic Embeddings
Wang et al. [8]
MLMs
Jiao et al. [9]
Rules
Coulombe et al. [7], Regina et al. [10], Louvan et al. [11]
Machine Translation
Back-translation
Xie et al. [12], Zhang et al. [13]
Unidirectional Translation
Nishikawa et al. [14], Bornea et al. [15]
Model Generation
Hou et al. [16], Li et al. [17], Liu et al. [18]
Noising
Swapping
Wei et al. [6], Luque et al. [19], Yan et al. [20]
Deletion
Wei et al. [6], Peng et al. [21], Yu et al. [22]
Insertion
Wei et al. [6], Peng et al. [21], Yan et al. [20]
Substitution
Coulombe et al. [7], Xie et al. [23], Louvan et al. [11]
Mixup
Guo et al. [24], Cheng et al. [25]
Sampling
Rules
Min et al. [26], Liu et al. [27]
Seq2Seq Models
Kang et al. [28], Zhang et al. [13], Raille et al. [29]
Language Models
Tavor et al. [30], Kumar et al. [31], Ng et al. [32]
Thakur et al. [33], Quteineh et al. [34]
Self Training
Du et al. [35], Montella et al. [36]

Figure 1: Taxonomy of NLP DA methods

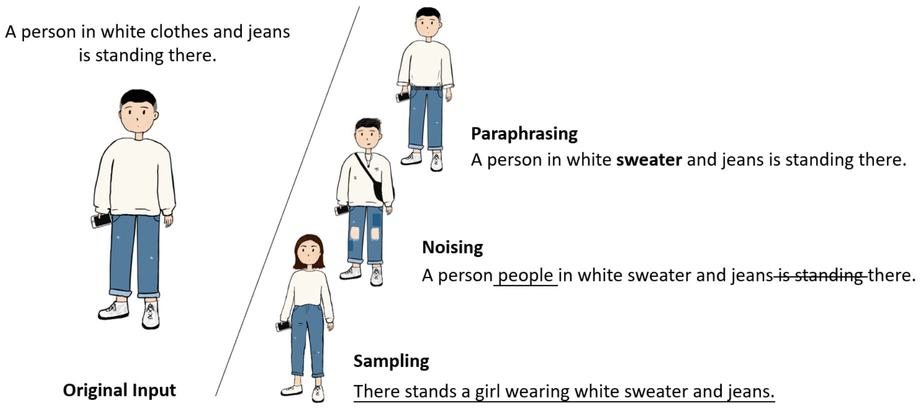

A person in white clothes and jeans is standing there.
Original Input
Paraphrasing
A person in white sweater and jeans is standing there.
Noising
A person people in white sweater and jeans ~~is standing~~ there.
Sampling
There stands a girl wearing white sweater and jeans.

Figure 2: Data augmentation techniques include three categories. The examples of the original data and augmented data are on the left and right, respectively. As we can see, the **diversity** of *paraphrasing*, *noising*, and *sampling* increases in turn compared to the original input.

2. Data Augmentation Methods in NLP

Data Augmentation aims at generating additional, synthetic training data in insufficient data scenes. Data augmentation ranges from simple techniques like rules-based methods to learnable generation-based methods, and all the above methods essentially guarantee the validity of the augmented data [29]. That is to say, DA methods need to make sure the augmented data is valid for the task, i.e., be considered part of the same distribution of the original data [29]. For example, similar semantics in machine translation and the same label in text classification as the original data.

Based on validity, the augmented data is expected to be various for a better generalization capacity of the downstream methods. This involves the **diversity** of augmented data. Different diversity involves different methods and the corresponding augmentation effects. In this survey, we novelly divide DA methods into three categories according to the diversity of their augmented data: paraphrasing, noising, and sampling.

- The paraphrasing-based methods generate augmented data that has limited semantic difference from the original data, based on proper and restrained changes to sentences. The augmented data convey very similar information as the original form.

- The noising-based methods add discrete or continuous noise under the premise of guaranteeing validity. The point of such methods is to improve the robustness of the model.

- The sampling-based methods master the data distributions and sample novel points within them. Such methods output more diverse data and satisfy more needs of downstream tasks based on artificial heuristics and trained models.

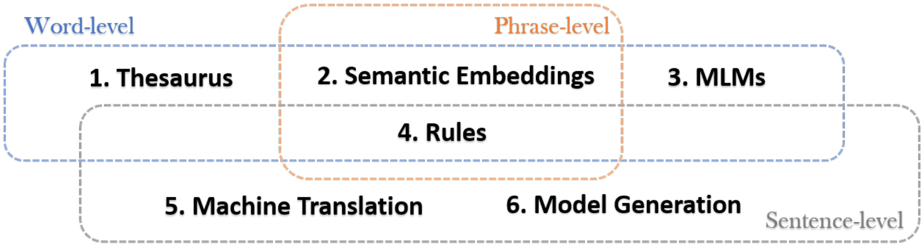


Figure 3: data augmentation techniques by paraphrasing include three levels: word-level, phrase-level, and sentence-level.

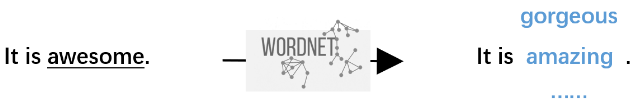


Figure 4: Paraphrasing by using thesauruses.

As shown in the examples and diagrams in Figure 2, the paraphrasing, noising, and sampling-based methods provide more diversity in turn. In this section, we will introduce and analyze them in detail.[2]

2.1. Paraphrasing-based Methods

As common phenomena in natural language, paraphrases are alternative ways to convey the same information as the original form [37, 38]. Naturally, the generation of paraphrases is a suitable solution for data augmentation. Paraphrases may occur at several levels including lexical paraphrases, phrasal paraphrases, and sentential paraphrases (Figure 3). Thus, data augmentation techniques by paraphrases generation also include these three types of rewriting.

2.1.1. Thesauruses

Some works replace words in the original text with their true synonyms and hyperonyms,[3] so as to obtain a new way of expression while keeping the semantics of the original text as unchanged as possible. As shown in Figure 4, thesauruses like WordNet [40] contain lexical triplets of words and are often used as external resources.

Zhang et al. [5] are the first to apply thesaurus in data augmentation. They use a thesaurus derived from WordNet,[4] which sorts the synonyms of words according to their similarity. For each sentence, they retrieve all replaceable words and randomly choose r of them to be replaced. The probability of number

[2] The specific classification is shown in Figure 12

[3] Replacing a word by an antonym or a hyponym (more specific word) is usually not a semantically invariant transformation. [39]

[4] The thesaurus is obtained from the Mytheas component used in LibreOffice project.

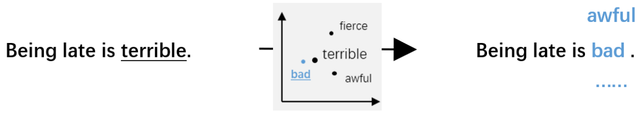


Figure 5: Paraphrasing by using semantic embeddings.

r is determined by a geometric distribution with parameter p in which $P[r] \sim p^r$. The index s of the synonym chosen given a word is also determined by a another geometric distribution in which $P[s] \sim p^s$. This method ensures synonyms that are more similar to the original word are selected with a greater probability. Some methods [41, 42, 43] apply a similar method.

A widely used text augmentation method called EDA (**E**asy **D**ata **A**ugmentation Techniques) [6] also replaces the original words with their synonyms using WordNet: they randomly choose n words from the sentence that are not stop words, and replace each of these words with one of its synonyms chosen at random, instead of following the geometric distribution.[5] Zhang et al. [44] apply a similar method in extreme multi-label classification.

In addition to synonyms, Coulombe et al. [7] propose to use hypernyms to replace the original words. They also recommend the types of words that are candidates for lexical substitution in order of increasing difficulty: adverbs, adjectives, nouns and verbs. Zuo et al. [45] use WordNet and VerbNet [46] to retrieve synonyms, hypernyms, and words of the same category.

> **Thesauruses**
> **Advantage(s):**
> 1. Easy to use.
>
> **Limitation(s):**
> 1. The scope and part-of-speech of replacement words are limited.
> 2. This method cannot solve the problem of ambiguity.
> 3. The sentence semantics may be affected if too many replacements occur.

2.1.2. Semantic Embeddings

This method overcomes the limitation of the replacement range and word part-of-speech in the Thesaurus-based method. It uses pre-trained word vectors, such as Glove, Word2Vec, FastText, etc., and replaces them with the word closest to the original word in the vector space, as shown in Figure 5.

In the Twitter message classification task, Wang et al. [8] pioneer to use both word embeddings and frame embeddings instead of discrete words.[6] As for word embeddings, each original word in the tweet is replaced with one of the k-nearest-neighbor words using cosine similarity. For example, "Being late is terrible" becomes "Being behind are bad". As for frame semantic embeddings,

[5] n is proportional to the length of the sentence.
[6] The frame embeddings refer to the continuous embeddings of semantic frames [47].

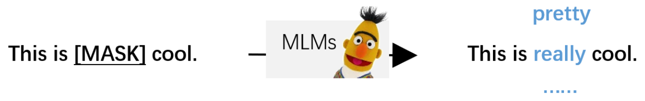


Figure 6: Paraphrasing by using language models.

the authors semantically parse 3.8 million tweets and build a continuous bag-of-frame model to represent each semantic frame using Word2Vec [48]. The same data augmentation approach as words is then applied to semantic frames.

Compared to Wang et al. [8], Liu et al. [49] only use word embeddings to retrieve synonyms. In the meanwhile, they edit the retrieving result with a thesaurus for balance. RamirezEchavarria et al. [50] create the dictionary of embeddings for selection.

> **Semantic Embeddings**
> **Advantage(s):**
> 1. Easy to use.
> 2. Higher replacement hit rate and wider replacement range.
>
> **Limitation(s):**
> 1. This method cannot solve the problem of ambiguity.[7]
> 2. The sentence semantics may be affected if too many replacements occur.

2.1.3. Language Models

The pre-trained language model has become the mainstream model in recent years due to its excellent performance. Masked language models (MLMs) such as BERT and BoBERTa have obtained the ability to predict the masked words in the text based on the context through pre-training, which can be used for text data augmentation (as shown in Figure 6). Moreover, this method alleviates the problem of ambiguity since MLMs consider the whole context.

Jiao et al. [9] use data augmentation to obtain task-specific distillation training data. They apply the tokenizer of BERT to tokenize words into multiple word pieces and form a candidate set for each word piece. Both word embeddings and masked language models are used for word replacement. Specifically, if a word piece is not a complete word ("est" for example), the candidate set is made up of its K-nearest-neighbor words by Glove. If the word piece is a complete word, the authors replace it with [MASK] and employ BERT to predict K Words to form a candidate set. Finally, a probability of 0.4 is used to determine whether each word piece is replaced by a random word in the candidate set.

Regina et al. [10], Tapia-Téllez et al. [51], Lowell et al. [52] and Palomino et al. [53] apply a similar method. They mask multiple words in a sentence and generate new sentences by filling these masks to generate more varied sentences. In addition, RNNs are also used for replacing the original word based on the context ([54, 55]).

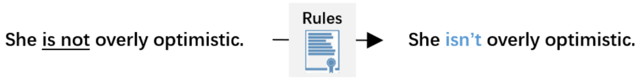


Figure 7: Paraphrasing by using rules.

Language Models
Advantage(s):
1. This method alleviates the problem of ambiguity.
2. This method considers context semantics.
Limitation(s):
1. Still limited to the word level.
2. The sentence semantics might be affected if too many replacements occur.

2.1.4. Rules

This method requires some heuristics about natural language that ensure the maintaining of sentence semantics, as shown in Figure 7.

On the one hand, some works rely on existing dictionaries or fixed heuristics to generate word-level and phrase-level paraphrases. Coulombe et al. [7] introduce the use of regular expressions to transform the form without changing sentence semantics, such as the abbreviations and prototypes of verbs, modal verbs, and negation. For example, replace "is not" with "isn't". Similarly, Regina et al. [10] perform replacements from expanded to abbreviated form and inversely between a group of words and the corresponding abbreviation, relying on word-pair dictionaries.

On the other hand, some works generate sentence-level paraphrases for original sentences with some rules, e.g. dependency trees. Coulombe et al. [7] first introduce a method via dependency trees. They use a syntactic parser to build a dependency tree for the original sentence. Then the paraphrases generator transforms this dependency tree to create a transformed dependency tree guided by a transformation grammar. For example, replace "Sally embraced Peter excitedly." with "Peter was embraced excitedly by Sally.". The transformed dependency tree is then used to generate a paraphrase as the augmented data. Dehouck et al. [56] apply a similar method. Louvan et al. [11] crop particular fragments on the dependency tree to create a smaller sentence. They also rotate the target fragment around the root of the dependency parse structure, without harming the original semantics.

Rules
Advantage(s):
1. Easy to use.
2. This method preserves the original sentence semantics.
Limitation(s):
1. This method requires artificial heuristics.
2. Low coverage and limited variation.

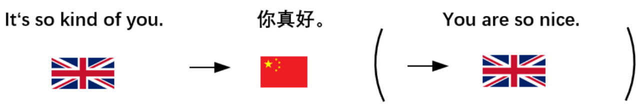


Figure 8: Paraphrasing by machine translation.

2.1.5. Machine Translation

Translation is a natural means of paraphrasing. With the development of machine translation models and the availability of online APIs, machine translation is popular as the augmentation method in many tasks, as shown in Figure 8.

Back-translation. This method means that the original document is translated into other languages, and then translated back to obtain the new text in the original language. Different from word-level methods, back-translation does not directly replace individual words but rewrites the whole sentence in a generated way.

Xie et al. [12], Yu et al. [57], and Fabbri et al. [58] use English-French translation models (in both directions) to perform back-translation on each sentence and obtain their paraphrases. Lowell et al. [52] also introduce this method as one of the unsupervised data augmentation methods. Zhang et al. [13] leverage back-translation to obtain the formal expression of the original data in the style transfer task.

In addition to some trained machine translation models, Google's Cloud Translation API service is a common tool for back-translation widely applied by some works like [7, 19, 59, 42, 60, 61, 10, 62, 63].[8]

Some works add additional features based on vanilla back-translation. Nugent et al. [64] propose a range of softmax temperature settings to ensure diversity while preserving semantic meaning. Qu et al. [65] combine back-translation with adversarial training, to synthesize diverse and informative augmented examples by organically integrating multiple transformations. Zhang et al. [13] employ a discriminator to filter the sentences in the back-translation results. This method greatly improves the quality of the augmented data as a threshold.

Unidirectional Translation. Different from back-translation, the unidirectional translation method directly translates the original text into other languages once, without translating it back to the original language. This method usually occurs in a multilingual scene.

In the task of unsupervised cross-lingual word embeddings (CLWEs), Nishikawa et al. [14] build pseudo-parallel corpus with an unsupervised machine translation model. The authors first train unsupervised machine translation (UMT) models using the source/target training corpora and then translate the corpora using

[8] The link of Google's Cloud Translation API service is: `https://cloud.google.com/translate/docs/apis`

the UMT models. The machine-translated corpus is concatenated with the original corpus for the learning of monolingual word embeddings independently for each language. Finally, the learned monolingual word embeddings are mapped to a shared CLWE space. This method both facilitates the structural similarity of two monolingual embedding spaces and improves the quality of CLWEs in the unsupervised mapping method.

Bornea et al. [15], Barrire et al. [66] and Aleksandr et al. [62] translate the original English corpus into several other languages and obtain multiplied data. Correspondingly, they use multilingual models.

> **Machine Translation**
> **Advantage(s):**
> 1. Easy to use.
> 2. Strong applicability.
> 3. This method ensures correct grammar and unchanged semantics.
> **Limitation(s):**
> 1. Poor controllability and limited diversity because of the fixed machine translation models.

2.1.6. Model Generation

Some methods employ Seq2Seq models to generate paraphrases directly. Such models output more diverse sentences given proper training objects, as shown in Figure 9.

Hou et al. [16] proposed a Seq2Seq data augmentation model for the language understanding module of task-based dialogue systems. They feed the delexicalized input utterance and the specified diverse rank k (e.g. 1, 2, 3) into the Seq2Seq model as input to generate a new utterance. Similarly, Hou et al. [67] encodes the concatenated multiple input utterances by an L-layer transformer. The proposed model uses duplication-aware attention and diverse-oriented regularization to generate more diverse sentences.

In Aspect Term Extraction, Li et al. [17] adopt Transformer as the basic structure. The masked original sentences as well as their label sequences are used to train a model M that reconstructs the masked fragment as the augmented data.[9] Kober et al. [68] use GAN to generate samples that are very similar to the original data. Liu et al. [18] employ a pre-trained model to share the question embeddings and the guidance for the proposed Transformer-based model. Then the proposed model could generate both context-relevant answerable questions and unanswerable questions.

[9] Half of the words in original sentences whose sequence labels are not 'O' are masked.

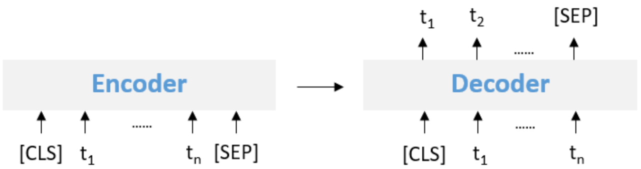


Figure 9: Paraphrasing by model generation.

Model Generation
Advantage(s):
1. Strong diversity.
2. Strong application.
Limitation(s):
1. Require training data.
2. High training difficulty.

2.2. Noising-based Methods

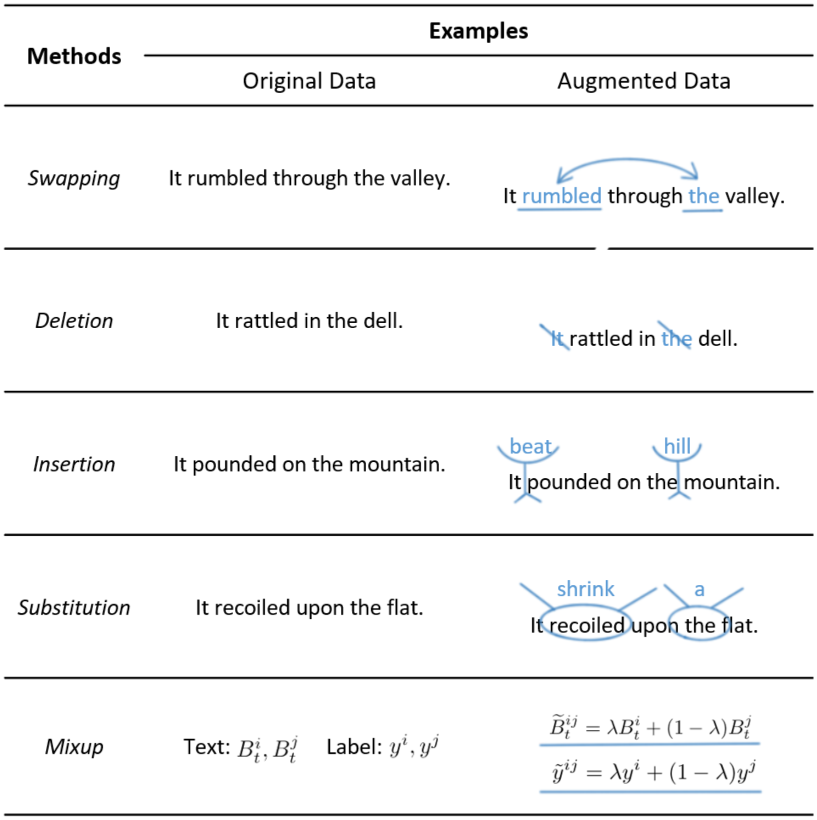
The focus of paraphrasing is to make the semantics of the augmented data as similar to the original data as possible. In contrast, the noising-based methods add faint noise that does not seriously affect the semantics, so as to make it appropriately deviate from the original data. Humans can greatly reduce the impact of weak noise on semantic understanding through the mastery of language phenomena and prior knowledge, but this noise may bring challenges to the model. Thus, this method not only expands the amount of training data but also improves model robustness.

2.2.1. Swapping

The semantics of natural language is sensitive to text order information, while slight order change is still readable for humans [69]. Therefore, the random swapping between words even sentences within a reasonable range can be used as a data augmentation method.

Wei et al. [6] randomly choose two words in the sentence and swap their positions. This work repeat this process n times, in which n is proportional to the sentence length l. Longpre et al. [60], Rastogi et al. [61], and Zhang et al. [44] also apply the same method. Dai et al. [43] first split the token sequence into segments according to labels, then randomly choose some segments to shuffle the order of the tokens inside, with the label order unchanged.

In addition to word-level swapping, some works also propose instance-level and sentence-level swapping. In the task of tweet sentiment analysis, Luque et al. [19] divide tweets into two halves. They randomly sample and combine first halves with second halves that have the same label. Although the data generated in this way may be ungrammatical and semantically unsound, it still carries relatively complete semantics and emotional polarity compared to a single word. Yan et al. [20] perform sentence-level random swapping on legal documents classification. Since sentences independently contain relatively complete semantics



Methods	Examples	
	Original Data	Augmented Data
Swapping	It rumbled through the valley.	It rumbled through the valley.
Deletion	It rattled in the dell.	~~It~~ rattled in ~~the~~ dell.
Insertion	It pounded on the mountain.	beat hill It pounded on the mountain.
Substitution	It recoiled upon the flat.	shrink a It recoiled upon the flat.
Mixup	Text: B_t^i, B_t^j Label: y^i, y^j	$\widetilde{B}_t^{ij} = \lambda B_t^i + (1-\lambda)B_t^j$ $\tilde{y}^{ij} = \lambda y^i + (1-\lambda)y^j$

Figure 10: The example of five noising-based methods.

comparing to words, the sentence order in the legal document has little effect on the meaning of the original text. Consequently, the authors shuffle the sentences to obtain the augmented text.

2.2.2. Deletion

This method means randomly deleting words in a sentence or deleting sentences in a document.

As for word-level deletion, Wei et al. [6] randomly remove each word in the sentence with probability p. Longpre et al. [60], Rastogi et al. [61], and Zhang et al. [44] also apply the same method. In the task of Spoken Language Understanding, Peng et al. [21] augment input dialogue acts by deleting slot values to obtain more combinations.

As for sentence-level deletion, Yan et al. [20] randomly delete each sentence in a legal document according to a certain probability. They do this because there exist many irrelevant statements and deleting them will not affect the understanding of the legal case. Yu et al. [22] employ the attention mechanism for both word-level and sentence-level random deletion.

2.2.3. Insertion

This method means randomly inserting words into a sentence or inserting sentences into a document.

As for word-level deletion, Wei et al. [6] select a random synonym of a random word in a sentence that is not a stop word, then insert that synonym into a random position in the sentence. The work repeats this process n times. In the task of Spoken Language Understanding, Peng et al. [21] augment input dialogue acts by inserting slot values to obtain more combinations.

In legal documents classification, since documents with the same label may have similar sentences, Yan et al. [20] employ sentence-level random insertion. They randomly select sentences from other legal documents with the same label to get augmented data.

> ℹ Random insertion introduces new noisy information that may change the original label. Tips to avoid this problem:
> 1. Use label-independent external resources at the word level.
> 2. Use other samples with the same label as the original data at the sentence level.

2.2.4. Substitution

This method means randomly replacing words or sentences with other strings. Different from the above paraphrasing methods, this method usually avoids using strings that are semantically similar to the original data.

Some works implement substitution through existing outer resources. Coulombe et al. [7] and Regina et al. [10] introduce a list of the most common misspellings in English to generate augmented texts containing common misspellings.[10] For example, “across” is easily misspelled as “accross”. Xie et al. [23] borrow from the idea of “word-dropout” and improve generalization by reducing the information in the sentence. This work uses “_” as a placeholder to replace random words, indicating that the information at that position is empty. Peng et al. [70] use pseudo-IND parallel corpus embeddings to create dictionaries and generate augmented data.

Some works use task-related resources or generate random strings for substitution. Xie et al. [12] and Xie et al. [23] replace the original words with other words in the vocabulary, and they use the TF-IDF value and the unigram frequency to choose words from the vocabulary, respectively. Lowell et al. [52] and Daval et al. [42] also explore this method as one of unsupervised data augmentation methods. Wang et al. [71] propose a method that randomly replaces words in the input and target sentences with other words in the vocabulary. In NER, Dai et al. [43] replace the original token with a random token in the training set with the same label. Qin et al. [72] propose a multi-lingual code-switching method that replaces original words in the source language with words of other languages. In the task of task-oriented dialogue, random substitution is a useful way to generate augmented data. Peng et al. [21] augment input dialogue acts by replacing slot values to obtain more combinations in spoken language understanding. In slot filling, Louvan et al. [11] do slot substitution according to the slot label. Song et al. [73] augment the training data for dialogue state tracking by copying user utterances and replace the corresponding real slot values with generated random strings.

> Random substitution introduces new noisy information that may change the original label. Tips to avoid this problem:
> 1. Use label-independent external resources at the word level.
> 2. Use other samples with the same label as the original data at the sentence level.

2.2.5. Mixup

The idea of Mixup first appears in the image field by Zhang et al. [74]. Inspired by this work, Guo et al. [24] propose two variants of Mixup for sentence classification. The first one called wordMixup conducts sample interpolation in the word embedding space, and the second one called senMixup interpolates the hidden states of sentence encoders. The interpolated new sample through wordMixup as well as senMixup, and their common interpolated label are obtained as follows:

$$\widetilde{B}_t^{ij} = \lambda B_t^i + (1-\lambda) B_t^j \tag{1}$$

$$\widetilde{B}_{\{k\}}^{ij} = \lambda f(B^i)_{\{k\}} + (1-\lambda) f(B^j)_{\{k\}} \tag{2}$$

[10] A list of common spelling errors in English can be obtained from the online resources of Oxford Dictionaries: `https://en.oxforddictionaries.com/spelling/common-misspellings`

$$\tilde{y}^{ij} = \lambda y^i + (1-\lambda) y^j \tag{3}$$

, in which $B_t^i, B_t^j \in R^{N \times d}$ denote the t-th word in two original sentences, and $f(B^i), f(B^j)$ denote the hidden layer sentence representation. Moreover, y^i, y^j are the corresponding original labels.

Mixup is widely applied in many works recently. Given the original samples, Cheng et al. [25] firstly construct their adversarial samples following [75], and then apply two Mixup strategies named P_{adv} and P_{aut}: The former interpolates between adversarial samples, and the latter interpolates between the two corresponding original samples. Similarly, Sun et al. [76], Bari et al. [77] , and Si et al. [78] both apply such Mixup method for text classification. Sun et al. [76] propose Mixup-Transformer which combines Mixup with transformer-based pretrained architecture. They test its performance on text classification datasets. Chen et al. [79] introduce Mixup into NER, proposing both Intra-LADA and InterLADA.

> 1. Mixup introduces continuous noise instead of discrete noise, it could generate augmented data between different labels.
> 2. This method is less interpretable and more difficult than the above noising-based methods.

> **Noising**
> **Advantage(s):**
> 1. Noising-based methods improve model robustness.
> 2. Easy to use (in most cases).
> 1. Distorted syntax and semantics.
> 2. Limited diversity for every single method.

2.3. Sampling-based Methods

The sampling-based methods master the data distributions and sample novel points within them. Similar to paraphrasing-based models, they also involve rules and trained models to generate augmented data. The Difference is that the sampling-based methods are task-specific and require task information like labels and data format.[11] Such methods not only ensure validity but also increase diversity. They satisfy more needs of downstream tasks based on artificial heuristics and trained models, and can be designed according to specific task requirements. Thus, they are usually more flexible and difficult than the former two categories.

[11] Recall that paraphrasing-based methods are task-independent and only require the original sentence as input.

2.3.1. Rules

This method uses some rules to directly generate new augmented data. Heuristics about natural language and the corresponding labels are sometimes required to ensure the validity of the augmented data. The model structure is as shown in Figure 11(a). Different from the above rule-based paraphrasing method, this method constructs valid but not guaranteed to be similar to the original data (even different labels).

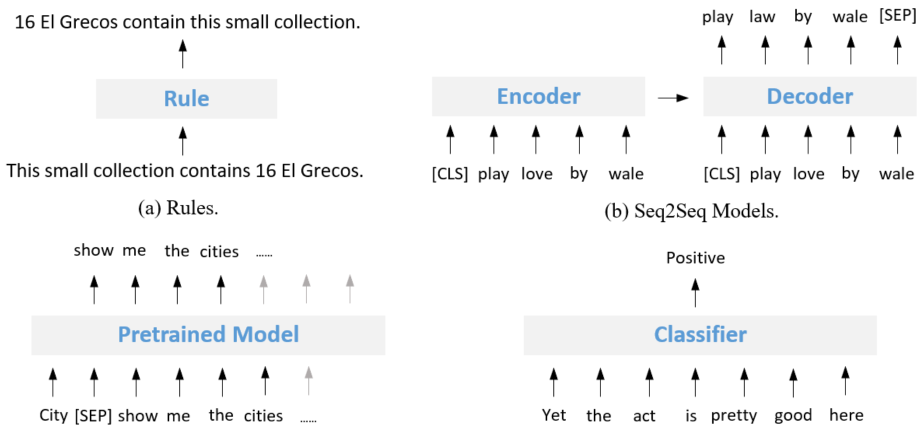
Min et al. [26] swap the subject and object of the resource sentence, and convert predicate verbs into passive form. For example, inverse "This small collection contains 16 El Grecos." into "16 El Grecos contain this small collection.". The labels of new samples are determined by rules. Liu et al. [27] apply data augmentation methods in the task of solving Math Word Problems (MWPs). They filter out some irrelevant numbers. Then some rules are used to construct new data based on the idea of double-checking, e.g., constructing augmented data describing $distance = time \times speed$ by reusing the original data describing $time = distance/speed$. The output equations of this method are computationally right. Given the training set of Audio-Video Scene-Aware Dialogue that provides 10 question-answer pairs for each video, Mou et al. [80] shuffle the first n pairs as dialogue history and take the $n + 1$-th question as what needs to be answered. In natural language inference, Kang et al. [28] apply external resources like PPDB and artificial heuristics to construct new sentences. Then they combine the new sentences with original sentences as augmented pairs according to rules, for example, *if A entails B and B entails C, then A entails C*. Kober et al. [68] define some rules to construct positive and negative pairs using adjective-noun (AN) and noun-noun (NN) compounds. For example, given $< car, car >$, they construct $< fastcar, car >$ as a positive sample and $< fastcar, redcar >$ as a negative sample. Shakeel et al. [81] construct both paraphrase annotations and non-paraphrase annotations through three properties including reflexivity, symmetry, and transitive extension. Yin et al. [82] use two kinds of rules including symmetric consistency and transitive consistency, as well as logic-guided DA methods to generate DA samples.

Rules
Advantage(s):
1. Easy to use.
Limitation(s):
1. This method requires artificial heuristics.
2. Low coverage and limited variation.

2.3.2. Seq2Seq Models

Some methods use non-pretrained models to generate augmented data. Such methods usually entail the idea of **back translation (BT)** [83],[12] which is to

[12] Note that the idea of back translation here is DIFFERENT from the above paraphrasing method called "back-translation" in Section 2.1.5.


16 El Grecos contain this small collection.
Rule
This small collection contains 16 El Grecos.
play law by wale [SEP]
Encoder
Decoder
[CLS] play love by wale
[CLS] play love by wale
show me the cities ......
Pretrained Model
City [SEP] show me the cities ......
Positive
Classifier
Yet the act is pretty good here

(a) Rules.

(b) Seq2Seq Models.

(c) Language Models.

(d) Self-training.

Figure 11: Sampling-based models.

train a target-to-source Seq2Seq model and use the model to generate source sentences from target sentences, i.e., constructing pseudo-parallel sentences [13]. Such Seq2Seq model learns the internal mapping between the distributions of the target and the source, as shown in Figure 11(b). This is different from the model generation based paraphrasing method because the augmented data of the paraphrasing method shares similar semantics with original data.

Sennrich et al. [84] train an English-to-Chinese NMT model using existing parallel corpus, and use the target English monolingual corpus to generate Chinese corpus through the above English-to-Chinese model. Kang et al. [28] train a Seq2Seq model for each label (*entailment*, *contradiction*, and *neutral*) and then generate new data using the Seq2Seq model given a sentence. Chen et al. [85] adopt the Tranformer architecture and think of the "rewrite utterance → request utterance" mapping as the machine translation process. Moreover, they enforce the optimization process of the Seq2Seq generation with a policy gradient technique for controllable rewarding. Zhang et al. [13] use Transformer as the encoder and transfer the knowledge from Grammatical Error Correction to Formality Style Transfer. Raille et al. [29] create the Edit-transformer, a Transformer-based model works cross-domain. Yoo et al. [86] propose a novel VAE model to output the semantic slot sequence and the intent label given an utterance.

> **Seq2Seq Models**
> **Advantage(s):**
> 1. Strong diversity.
> 2. Strong application.
> **Limitation(s):**
> 1. Require training data.
> 2. High training difficulty.

2.3.3. Language Models

In recent years, pretrained language models have been widely used and have been proven to contain knowledge. Thus, they are naturally used as augmentation tools, as shown in Figure 11(c).

Tavor et al. [30] propose a data augmentation method named LAMBDA. They generate labeled augmented sentences with GPT-2, which is fine-tuned on the training set in advance. Then the augmented sentences are filtered by a classifier to ensure the data quality. Kumar et al. [31] applies a similar method without the classifier for filtering.

Some works adopt masked language models to obtain augmented data. Ng et al. [32] use the masked language model to construct a corruption model and a reconstruction model. Given the input data points, they initially generate data far away from the original data manifold with the corruption model. Then the reconstruction model is used to pull the data point back to the original data manifold as the final augmented data.

Some works adopt auto-regressive models to obtain augmented data. Peng et al. [21] use the pre-trained SC-GPT and SC-GPT-NLU to generate utterances and dialogue acts respectively. The results are filtered to ensure the data quality. Abonizio et al. [87] fine-tune DistilBERT [88] on original sentences to generate synthetic sentences. Especially, GPT-2 is a popular model used for generating augmented data. Quteineh et al. [34] use label-conditioned GPT-2 to generate augmented data. Tarján et al. [89] generate augmented data with GPT-2 and retokenize them into statistically derived subwords to avoid the vocabulary explosion in a morphologically rich language. Zhang et al. [44] use GPT-2 to generate substantially diversified augmented data in extreme multi-label classification.

> **Language Models**
> **Advantage(s):**
> 1. Strong application.
> **Limitation(s):**
> 1. Require training data.

2.3.4. Self-training

In some scenarios, unlabeled raw data is easy to obtain. Thus, converting such data into valid data would greatly increase the amount of data, as shown in Figure 11(d).

Some methods train a model on the gold dataset to predict labels for unlabeled data. Thakur et al. [33] first fine-tune BERT on gold data, then use the fine-tuned BERT to label unlabeled sentence pairs. Such augmented data, as well as the gold data, are used to train SBERT together. Miao et al. [90] further introduce data distillation into the self-training process. They output the label of unlabeled data by the iteratively updated teacher model. Yang et al. [91] apply a similar self-training method in the Question Answering task; a cross-attention-based teacher model is used to determine the label of each QA pair. Du et al. [35] introduce SentAugment, a data augmentation method that computes task-specific query embeddings from labeled data to retrieve sentences from a bank of billions of unlabeled sentences crawled from the web.

Some methods directly transfer exsiting models from other tasks to generate pseudo-parallel corpus. Montella et al. [36] make use of Wikipedia to leverage a massive sentences. Then they use Stanford OpenIE package to extract the triplets given Wikipedia sentences. For example, given "*Barack Obama was born in Hawaii.*", the returned triples by Stanford OpenIE are $< BarackObama; was; born > and < BarackObama; wasbornin; Hawaii >$ Such mappings are flipped as the augmented data of RDF-to-text tasks. Aleksandr et al. [62] apply a similar method. Since BERT does well on object-property (OP) relationship prediction and object-affordance (OA) relationship prediction, Zhao et al. [92] directly use a fine-tuned BERT to predict the label of OP and OA samples.

> **Self-training**
> **Advantage(s):**
> 1. Easier than generative models.
> 2. Suitable for data-sparse scenarios.
> 1. Require for unlabeled data.
> 2. Poor application.

2.4. Analysis

As shown in Table 1, we compare the above DA methods by various aspects.

- It is easy to find that nearly all paraphrasing-based and noising-based methods are not learnable, except for *Seq2Seq* and *Mixup*. However, most sampling-based methods are learnable except for the *rules*-based ones. Learnable methods are usually more complex than non-learnable ones, thus sampling-based methods generate more diverse and fluent data than the former two.

- Among all learnable methods, *Mixup* is the only **online** one. That is to say, the process of augmented data generation is independent of downstream task model training. Thus, *Mixup* is the only one that outputs cross-label and discrete embedding from augmented data.

- Comparing the learnable column and the resource column, we could see that most non-learnable methods require external knowledge resources which go beyond the original dataset and task definition. Commonly used resources include semantic thesauruses like WordNet and PPDB, handmade resources like misspelling dictionary in [7], and artificial heuristics like the ones in [26] and [28].

- Combining the first three columns, we could see that pretrained or non-pretrained models are widely used as DA methods in addition to external resources. It is because the knowledge in pretrained models and the training objects play a similar role to external resources when guiding augmented data generation.

Table 1: Comparing a selection of DA methods by various aspects. *Learnable* denotes whether the methods involve model training; *online* and *offline* denote online learning and offline learning, respectively. *Ext.Know* refers to whether the methods require external knowledge resources to generate augmented data. *Pretrain* denotes whether the methods require a pretrained model. *Task-related* denotes whether the methods consider the label information, task format, and task requirements to generate augmented data. *Level* denotes the depth and extent to which elements of the instance/data are modified by the DA; t, e, and l denote text, embedding, and label, respectively. *Granularity* indicates the extent to which the method could augment; w, p, and s denote word, phrase, and sentence, respectively.

		Learnable	Ext.Know	Pretrain	Task-related	Level	Granularity
Paraphrasing	Thesauruses	-	✓	-	-	t	w
	Embeddings	-	✓	-	-	t	w, p
	MLMs	-	-	✓	-	t	w
	Rules	-	✓	-	-	t	w, p, s
	MT	-	-	-	-	t	s
	Seq2Seq	offline	-	-	✓	t	s
Noising	Swapping	-	-	-	-	t	w, p, s
	Deletion	-	-	-	-	t	w, p, s
	Insertion	-	✓	-	-	t	w, p, s
	Substitution	-	✓	-	-	t	w, p, s
	Mixup	online	-	-	✓	e, l	s
Sampling	Rules	-	✓	-	✓	t, l	w, p, s
	Non-pretrained	offline	-	-	✓	t, l	s
	Pretrained	offline	-	✓	✓	t, l	s
	Self-training	offline	-	-	✓	t, l	s

- Comparing the learnable column and the task-related column, we could see that in the two categories of paraphrasing and noising, almost all methods are not task-related. They could generate augmented data given only original data without labels or task definition. However, all sampling methods are task-related because they adopt heuristics and model training to satisfy the needs of specific tasks.

- Comparing the level column and the task-related column, we could see that they are relevant. The paraphrasing-based methods are at the text level. So does the noising-based methods, except for Mixup because it makes changes in the embeddings as well as the labels. All sampling-based methods are at the text and label level since the labels are also considered and constructed during augmentation.

- Comparing the learnable column and the granularity column, we could see that almost all non-learnable methods could be used for word-level and phrase-level DA, but all learnable methods could only be applied for sentence-level DA. Although learnable methods generate high-quality augmented sentences, unfortunately, they do not work for document augmentation because of their weaker processing ability for documents. Thus, document augmentation still relies on simple non-learnable methods, which is also a current situation we have observed in our research.

3. Strategies and Tricks

The three types of DA methods including paraphrasing, noising, and sampling have been introduced above, and we analyzed their application in various NLP tasks. In practical applications, the effect of the DA method is influenced by many factors. In this chapter, we introduce such factors to inspire our readers to choose and construct suitable DA methods.

3.1. Method Stacking

The methods in Section 2 are not mandatory to be applied alone. They could be combined for better performance. Common combinations include:

The Same Type of Methods. Some works combine different paraphrasing-based methods and obtain different paraphrases, to increase the richness of augmented data. For example, Liu et al. [49] use both thesauruses and semantic embeddings, and Jiao et al. [9] use both semantic embeddings and MLMs. As for noising-based methods, the former unlearnable ways are usually used together like [21]. It is because these methods are simple, effective, and complementary. Some methods also adopt different sources of noising or paraphrasing like [10] and [23]. The combination of different resources could also improve the robustness of the model.

Unsupervised Methods. In some scenarios, the simple and task-independent unsupervised DA methods could meet the demand. Naturally, they are grouped together and widely used. Wei et al. [93] introduce a DA toolkit called EDA that consists of synonym replacement, random insertion, random swap, and random deletion. EDA is very popular and used for many tasks ([60, 61]). UDA by Xie et al [12] includes back-translation and unsupervised noising-based methods; it is also used in many tasks like [42].

Multi-granularity. Some works apply the same method at different levels to enrich the augmented data with changes of different granularities and improve the robustness of the model. For example, Wang et al. [8] train both word embeddings and frame embeddings by Word2Vec; Guo et al. [24] apply Mixup at the word and sentence level, and Yu et al. [22] use a series of noising-based methods at both the word and the sentence level.

3.2. Optimization

The optimization process of DA methods directly influences the quality of augmented data. We introduce it through four angles: the use of augmented data, hyperparameters, training strategies, and training objects.

3.2.1. The Use of Augmented Data

The way of using augmented data directly influences the final effect. From the perspective of data quality, the augmented data could be used to pre-train a model if it is not of high quality; otherwise, it could be used to train a model directly. From the perspective of data amount, if the amount of the augmented data is much higher than the original data, they are usually not directly used together for model training. Instead, some common practices include (1) oversampling the original data before training the model (2) pre-training the model with the augmented data and fine-tuning it on the original data.

3.2.2. Hyperparameters

All the above methods involve hyperparameters that largely affect the augmentation effect. We list some common hyperparameters in Figure 12:

3.2.3. Training Strategies

Some works apply training strategies based on the basic data augmentation methods. For example, Qu et al. [65] combine back-translation with adversarial training. Similarly, Quteineh et al. [34] transform the basic pre-trained model into an optimization problem [13] to maximize the usefulness of the generated output. Hu et al. [94] and Liu et al. [95] use pre-trained language models to generate augmented data, and transfer such progress into reinforcement learning. Some works ([61, 96]) take the idea of Generative Adversarial Networks to generate challenging augmented data.

[13] Monte Carlo Tree Search.

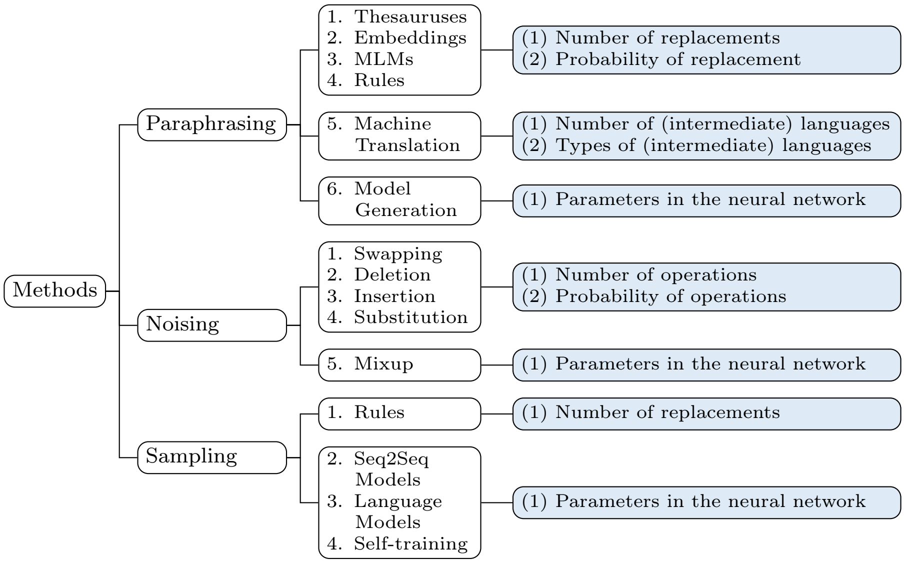

Methods
Paraphrasing
1. Thesauruses
2. Embeddings
3. MLMs
4. Rules
(1) Number of replacements
(2) Probability of replacement
5. Machine Translation
(1) Number of (intermediate) languages
(2) Types of (intermediate) languages
6. Model Generation
(1) Parameters in the neural network
Noising
1. Swapping
2. Deletion
3. Insertion
4. Substitution
(1) Number of operations
(2) Probability of operations
5. Mixup
(1) Parameters in the neural network
Sampling
1. Rules
(1) Number of replacements
2. Seq2Seq Models
3. Language Models
4. Self-training
(1) Parameters in the neural network

Figure 12: Hyperparameters that affect the augmentation effect in each DA method.

3.2.4. Training Objects

Training objects are essential for model training, especially for the learnable DA methods. Nugent et al. [64] propose a range of softmax temperature settings to ensure diversity while preserving semantic meaning. Hou et al. [67] use duplication-aware attention and diverse-oriented regularization to generate more diverse sentences. Cheng et al. [25] employ curriculum learning to encourage the model to focus on the difficult training examples.

3.3. Filtering

Sometimes the progress of data augmentation inevitably introduces some noise even errors, thus filtering mechanisms are introduced to avoid this problem.

Some works filter some input data in the initial stage to avoid inappropriate input affecting the augmentation effect. A typical example is sentence length, i.e., filter sentences that are too short ([17]). Liu et al. [27] filter out irrelevant numbers without augmenting them in solving Math Word Problems, to ensure the generated data is computationally right.

In addition, some works filter the synthetic augmented data at the end-stage. This is usually achieved through a model. For example, Zhang et al. [13] employ a discriminator to filter the back-translation results. Tavor et al. [30] and Peng et al. [21] both apply a classifier to filter the augmented sentences generated by pre-trained models to ensure the data quality.

Table 2: The application of DA methods in NLP tasks. Note that if a paper involves multiple methods, we count it multiple times.

		Text classification	Text generation	Structure prediction
Paraphrasing	Thesauruses	[5], [93], [49], [7], [42], [60], [44], [45], [98]	-	[42], [43]
	Embeddings	[8], [49]	-	-
	MLMs	[10], [51], [54]	[55]	-
	Rules	[10], [7], [11]	-	[99]
	MT	[42], [60], [10], [12], [59], [61], [63], [7], [19], [66], [100], [98]	[13], [58]	[42], [57], [15]
	Seq2Seq	[18], [68], [101]	[18], [102]	[18], [16], [67], [17], [103], [82]
Noising	Swapping	[93], [60], [44], [61], [20], [19]	-	[43]
	Deletion	[93], [60], [44], [61], [20], [22]	[21]	-
	Insertion	[93], [60], [44], [61]	[21]	-
	Substitution	[42], [10], [12], [7], [100]	[23], [71], [21]	[42], [11], [43], [104]
	Mixup	[24], [76], [78]	[25]	[79]
Sampling	Rules	[26], [28], [68], [81], [100], [105]	[80], [106], [107]	[108]
	Seq2Seq	[28], [29], [86], [109]	[13], [85], [110], [111], [84]	[86]
	Pretrained	[44], [32], [31], [34], [95], [30], [87], [112]	[21], [32], [89]	[21]
	Self-training	[35], [91], [62], [90]	[36]	[91]

4. Applications on NLP Tasks

Although a variety of data augmentation methods have emerged in the field of NLP in recent years, it is difficult to directly compare their performance. This is because different tasks, evaluation metrics, datasets, model architectures, and experimental settings make direct comparisons meaningless. Therefore, based on the work introduced above, we analyze the data augmentation methods from the perspective of different NLP tasks including text classification, text generation, and structured prediction [97].

- Text classification is the simplest and most basic natural language processing problem. That is, for a piece of text input, output the category to which the text belongs, where the category is a pre-defined closed set.[14]

[14] Text matching tasks such as Natural Language Inference can also be transformed into text classification.

- Text generation, as the name implies, is to generate the corresponding text given the input data. The most classic example is machine translation.

- The structured prediction problem is usually unique to NLP. Different from the text classification, there are strong correlation and format requirements between the output categories in the structured prediction problem.

In this section, we try to analyze the features as well as the development status of DA in these tasks. Some statistical results are shown in Table 2 and Table 3.

DA methods are applied more widely in text classification than other NLP tasks in general and in each category. Moreover, each individual DA method could be applied to text classification. Such application advantage is because of the simple form of text classification: given the input text, it directly investigates the model's understanding of semantics by label prediction. Therefore, it is relatively simple for data augmentation to only consider retaining the semantics of words that are important for classification.

As for text generation, it prefers sampling-based methods to bring more semantic diversity. And structured prediction prefers paraphrasing-based methods because it is sensitive to data format. Thus, it has higher requirements for data validity.

By comparing each DA method, we can see that simple and effective unsupervised methods, including machine translation, thesaurus-based paraphrasing, and random substitution, are quite popular. In addition, learnable methods like Seq2Seq paraphrasing models, pre-trained models, and self-training, also gain a lot of attention because of their diversity and effectiveness.

We also show the development process of the DA method on three types of tasks through a timeline (Table 3). On the whole, the number of applications of DA in these tasks has increased year by year. Text classification is the first task to use DA, and the number of corresponding papers is also larger than the other two tasks. In terms of text generation and structured prediction, DA has received more attention. Paraphrasing-based methods have always been a popular method, and in recent years sampling-based methods have also proven effective in text classification and text generation, but people still tend to use paraphrasing and noising-based methods in structured prediction.

Table 3 Timeline of DA methods applied in three kinds of NLP tasks. The time for each paper is based on its first arXiv version (if exists) or estimated submission time. P considers paraphrasing-based methods; N considers noising-based methods; S considers sampling-based methods.

	Text Classification	**Text Generation**	**Structured Prediction**
2015.09	Zhang et al. [5] P		
	Wang et al. [8] P		
2015.11		Sennrich et al. [84] S	
2016.01	Xu et al. [105] S		
...			
2017.03		Xie et al. [23] N	
2017.05		Fadaee et al. [55] P	
...			
2018.04			Yu et al. [57] P
2018.05	Kang et al. [28] S		
2018.06	Kobayashi et al. [54] P		
2018.07			Hou et al. [16] P
2018.08	Aroyehun et al. [63] P	Wang et al. [71] N	
2018.09	Yoo et al. [86] S		Yoo et al. [86] S
2018.10			Sahin et al. [99] P
2018.12	Coulombe et al. [7] P, N		
2019.01	Wei et al. [93] P, N		
2019.04	Xie et al. [12] P, N		
2019.05	Guo et al. [24] N	Gao et al. [113] N	
2019.06		Xia et al. [114] S	
2019.07	Yu et al. [22] N		
2019.08			Yin et al. [82] P
2019.09	Luque et al. [19] P, N		
	Yan et al. [20] N		

	Text Classification	Text Generation	Structured Prediction
2019.11	Anaby et al. [30] S		
	Malandrakis et al. [115] P		
2019.12	Shakeel et al. [81] S		
2020.01			Yoo et al. [103] P
2020.03	Kumar et al. [31] S		
	Raille et al. [29] S		
2020.04	Lun et al. [100] P, N, S	Peng et al. [21] N, S	Li et al. [17] P
			Peng et al. [21] S
2020.05	Kober et al. [68] P, S	Zhang et al. [13] P, S	
	Cao et al. [116] S		
2020.06	Liu et al. [49] P	Cheng et al. [25] N	
	Qin et al. [72] N		Qin et al. [72] N
2020.07	Min et al. [26] S	Chen et al. [111] S	
	Rastogi et al. [61] P, N	Tarjan et al. [89] S	
	Regina et al. [10] P, N	Mou et al. [80] S	
	Asai et al. [107] S		
2020.09	Ng et al. [32] S	Ng et al. [32] S	Yang et al. [91] S
	Zhang et al. [44] P,N, S	Zhang et al. [106] S	
2020.10	Barrire et al. [66] P	Fabbri et al. [58] P	Liu et al. [18] P
	Louvan et al. [11] P		Louvan et al. [11] N
	Tapia-Téllez et al. [51] P		Chen et al. [79] N
	Sun et al. [76] N		Dai et al. [43] P, N
	Abonizio et al. [87] S		
	Zuo et al. [45] P		
2020.11	Longpre et al. [60] P, N		
	Quteineh et al. [34] S		
2020.12	Miao et al. [90] S	Wan et al. [102] P	Bornea et al. [15] P
	Daval et al. [42] P ,N	Yao et al. [110]	Hou et al. [67] P
	Liu et al. [95] S	Montella et al. [36] S S	Daval et al. [42] P ,N
	Aleksandr et al. [62] S	Chen et al. [85] S	
	Si et al. [78] N		
	Xu et al. [101] P		
	Liu et al. [98] P		
2021.01	Shi et al. [104] N		Shi et al. [104] N
	Staliunaite et al. [112] S		

5. Related Topics

How does data augmentation relate to other learning methods? In this section, we connect data augmentation with other similar topics.

5.1. Pretrained Language Models

The training of most pre-trained language models (PLMs) is based on self-supervised learning. Self-supervised learning mainly uses auxiliary tasks to mine its supervised information from large-scale unsupervised data, and trains the network through this constructed supervised information, so that it can learn valuable representations for downstream tasks. From this perspective, PLMs also introduce more training data into downstream tasks, in an implicit way. On the other hand, the general large-scale unsupervised data of PLMs may be out-of-domain for specific tasks. Differently, the task-related data augmentation methods essentially focus on specific tasks.

5.2. Contrastive Learning

Contrastive learning is to learn an embedding space in which similar samples are close to each other while dissimilar ones are far apart. It focuses on learning the common features between similar samples and distinguishing the differences between dissimilar ones. The first step of contrastive learning is applying data augmentation to construct similar samples with the same label, and the second step is to randomly choose instances as the negative samples. Thus, contrastive learning is one of the applications of data augmentation.

5.3. Other Data Manipulation Methods

In addition to DA, there are some other data manipulation methods to improve model generalization [117, 94]. *Oversampling* is usually used in data imbalance scenarios. It simply samples original data from the minority group as new samples, instead of generating augmented data. *Data cleaning* is additionally applied to the original data to improve data quality and reduce data noise. It usually includes lowercasing, stemming, lemmatization, etc. *Data weighting* assigns different weights to different samples according to their importance during training, without generating new data. *Data synthesis* provides entire labeled artificial examples instead of augmented data generated by models or rules.

5.4. Generative Adversarial Networks

Generative Adversarial Networks (GANs) are first introduced by Goodfellow et al. [118]. As a type of semi-supervised method, GANs include the generative model, which is mainly used to challenge the discriminator of GANs, while the generative models in some DA methods are directly used to augment training data. Moreover, the generative model of GANS is applied as a DA method in some scenes like [61, 119, 96, 68, 109, 116], and have demonstrated to be effective for data augmentation purposes.

5.5. Adversarial Attacks

Adversarial attacks are techniques to generate adversarial examples attacking a machine learning model, i.e., causing the model to make a mistake. Some works use DA methods like code-switch substitution to generate adversarial examples as consistency regularization [120].

6. Challenges and Opportunities

Data augmentation has seen a great process over the last few years, and it has provided a great contribution to large-scale model training as well as the development of downstream tasks. Despite the process, there are still challenges to be addressed. In this section, we discuss some of these challenges and future directions that could help advance the field.

Theoretical Narrative. At this stage, there appears to be a lack of systematic probing work and theoretical analysis of DA methods in NLP. Most previous works propose new methods or prove the effectiveness of the DA method on downstream tasks, but do not explore the reasons and laws behind it, e.g., from the perspective of mathematics. The discrete nature of natural language makes theoretical narrative essential since narrative helps us understand the nature of DA, without being limited to determining effectiveness through experiments.

More Exploration on Pretrained Language Models. In recent years, pre-trained language models have been widely applied in NLP, which contain rich knowledge through self-supervision on a huge scale of corpora. There are works using pre-trained language models for DA, but most of them are limited to [MASK] completion, direct generation after fine-tuning, or self-training. Is DA still helpful in the era of pre-trained language models? Or, how to further use the information in pre-trained models to generate more diverse and high-quality data with less cost? The above are directions worth considering.

More Generalized Methods for NLP. Natural language is most different from image or sound in that its representation is discrete. At the same time, NLP includes specific tasks such as structured prediction that are not available in other modalities. Therefore, unlike general methods such as *clipping* for image augmentation or *speed perturbation* for audio augmentation, there is currently no DA method that can be effective for all NLP tasks. This means that there is still a gap for DA methods between different NLP tasks. With the development of pre-trained models, this seems to have some possibilities. Especially the proposal of T5 [121] and GPT3 [122], as well as the emergence of prompting learning further verify that the formalization of tasks in natural language can be independent of the traditional categories, and a more generalized model could be obtained by unifying task definitions.

***Working with Long Texts and Low Resources Languages*.** The existing methods have made significant progress in short texts and common languages. However, limited by model capabilities, DA methods on long texts still struggle with the simplest methods of paraphrasing and noising [49, 20, 22] (as shown in Table 1). At the same time, limited by data resources, augmentation methods of low resource languages are scarce [31], although they have more demand for data augmentation. Obviously, exploration in these two directions is still limited, and they could be promising directions.

7. Conclusion

In this paper, we presented a comprehensive and structured survey of data augmentation for natural language processing. In order to inspect the nature of DA, we framed DA methods into three categories according to **diversity** of augmented data, including paraphrasing, noising, and sampling. Such categories can help to understand and develop DA methods. We also present the application of DA methods in NLP tasks and analyzed them through a timeline. In addition, we introduced some tricks and strategies so that researchers and practitioners can refer to obtain better model performance. Finally, we distinguish DA with some related topics and outlined current challenges as well as opportunities for future research.

References

References

[1] C. Shorten, T. M. Khoshgoftaar, A survey on image data augmentation for deep learning, Journal of Big Data 6 (1) (2019) 1–48.

[2] P. Liu, X. Wang, C. Xiang, W. Meng, A survey of text data augmentation, 2020 International Conference on Computer Communication and Network Security (CCNS) (2020) 191–195.

[3] S. Y. Feng, V. Gangal, J. Wei, S. Chandar, S. Vosoughi, T. Mitamura, E. Hovy, A survey of data augmentation approaches for NLP, in: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, Association for Computational Linguistics, Online, 2021, pp. 968–988. `doi:10.18653/v1/2021.findings-acl.84`.
URL `https://aclanthology.org/2021.findings-acl.84`

[4] M. Bayer, M.-A. Kaufhold, C. Reuter, A survey on data augmentation for text classification, ArXiv abs/2107.03158 (2021).

[5] X. Zhang, J. Zhao, Y. LeCun, Character-level convolutional networks for text classification, in: NIPS, 2015.

[6] J. Wei, K. Zou, Eda: Easy data augmentation techniques for boosting performance on text classification tasks, ArXiv abs/1901.11196 (2019).

[7] C. Coulombe, Text data augmentation made simple by leveraging nlp cloud apis, ArXiv abs/1812.04718 (2018).

[8] W. Y. Wang, D. Yang, 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, 2015, pp. 2557–2563.

[9] X. Jiao, Y. Yin, L. Shang, X. Jiang, X. Chen, L. Li, F. Wang, Q. Liu, Tinybert: Distilling bert for natural language understanding, ArXiv abs/1909.10351 (2020).

[10] M. Regina, M. Meyer, S. Goutal, Text data augmentation: Towards better detection of spear-phishing emails (07 2020).

[11] S. Louvan, B. Magnini, Simple is better! lightweight data augmentation for low resource slot filling and intent classification, in: Proceedings of the 34th Pacific Asia Conference on Language, Information and Computation, Association for Computational Linguistics, Hanoi, Vietnam, 2020, pp. 167–177.
URL `https://aclanthology.org/2020.paclic-1.20`

[12] Q. Xie, Z. Dai, E. Hovy, M.-T. Luong, Q. V. Le, Unsupervised data augmentation for consistency training, arXiv preprint arXiv:1904.12848 (2019).

[13] Y. Zhang, T. Ge, X. Sun, Parallel data augmentation for formality style transfer, in: ACL, 2020.

[14] S. Nishikawa, R. Ri, Y. Tsuruoka, Data augmentation with unsupervised machine translation improves the structural similarity of cross-lingual word embeddings, 2020.

[15] M. A. Bornea, L. Pan, S. Rosenthal, R. Florian, A. Sil, Multilingual transfer learning for qa using translation as data augmentation, in: AAAI, 2021.

[16] Y. Hou, Y. Liu, W. Che, T. Liu, Sequence-to-sequence data augmentation for dialogue language understanding, ArXiv abs/1807.01554 (2018).

[17] K. Li, C. Chen, X. Quan, Q. Ling, Y. Song, Conditional augmentation for aspect term extraction via masked sequence-to-sequence generation, arXiv preprint arXiv:2004.14769 (2020).

[18] D. Liu, Y. Gong, J. Fu, Y. Yan, J. Chen, J. Lv, N. Duan, M. Zhou, Tell me how to ask again: Question data augmentation with controllable rewriting in continuous space, ArXiv abs/2010.01475 (2020).

[19] F. M. Luque, Atalaya at tass 2019: Data augmentation and robust embeddings for sentiment analysis, in: IberLEF@SEPLN, 2019.

[20] G. Yan, Y. Li, S. Zhang, Z. Chen, Data augmentation for deep learning of judgment documents, in: IScIDE, 2019.

[21] B. Peng, C. Zhu, M. Zeng, J. Gao, Data augmentation for spoken language understanding via pretrained models, ArXiv abs/2004.13952 (2020).

[22] S. Yu, J. Yang, D. Liu, R. Li, Y. Zhang, S. Zhao, Hierarchical data augmentation and the application in text classification, IEEE Access 7 (2019) 185476–185485.

[23] Z. Xie, S. I. Wang, J. Li, D. Lévy, A. Nie, D. Jurafsky, A. Ng, Data noising as smoothing in neural network language models, ArXiv abs/1703.02573 (2017).

[24] H. Guo, Y. Mao, R. Zhang, Augmenting data with mixup for sentence classification: An empirical study, arXiv preprint arXiv:1905.08941 (2019).

[25] Y. Cheng, L. Jiang, W. Macherey, J. Eisenstein, Advaug: Robust adversarial augmentation for neural machine translation, arXiv preprint arXiv:2006.11834 (2020).

[26] J. Min, R. T. McCoy, D. Das, E. Pitler, T. Linzen, Syntactic data augmentation increases robustness to inference heuristics, in: ACL, 2020.

[27] Q. Liu, W. Guan, S. Li, F. Cheng, D. Kawahara, S. Kurohashi, Reverse operation based data augmentation for solving math word problems, ArXiv abs/2010.01556 (2020).

[28] D. Kang, T. Khot, A. Sabharwal, E. Hovy, Adventure: Adversarial training for textual entailment with knowledge-guided examples, in: ACL, 2018.

[29] G. Raille, S. Djambazovska, C. Musat, Fast cross-domain data augmentation through neural sentence editing, ArXiv abs/2003.10254 (2020).

[30] A. Anaby-Tavor, B. Carmeli, E. Goldbraich, A. Kantor, G. Kour, S. Shlomov, N. Tepper, N. Zwerdling, Not enough data? deep learning to the rescue!, arXiv preprint arXiv:1911.03118 (2019).

[31] V. Kumar, A. Choudhary, E. Cho, Data augmentation using pre-trained transformer models, arXiv preprint arXiv:2003.02245 (2020).

[32] N. Ng, K. Cho, M. Ghassemi, Ssmba: Self-supervised manifold based data augmentation for improving out-of-domain robustness, ArXiv abs/2009.10195 (2020).

[33] N. Thakur, N. Reimers, J. Daxenberger, I. Gurevych, Augmented sbert: Data augmentation method for improving bi-encoders for pairwise sentence scoring tasks, in: NAACL, 2021.

[34] H. Quteineh, S. Samothrakis, R. Sutcliffe, Textual data augmentation for efficient active learning on tiny datasets, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Online, 2020, pp. 7400–7410. `doi:10.18653/v1/2020.emnlp-main.600`.
URL `https://aclanthology.org/2020.emnlp-main.600`

[35] J. Du, E. Grave, B. Gunel, V. Chaudhary, O. Çelebi, M. Auli, V. Stoyanov, A. Conneau, Self-training improves pre-training for natural language understanding, in: NAACL, 2021.

[36] S. Montella, B. Fabre, T. Urvoy, J. Heinecke, L. Rojas-Barahona, Denoising pre-training and data augmentation strategies for enhanced rdf verbalization with transformers, ArXiv abs/2012.00571 (2020).

[37] R. Barzilay, K. McKeown, Extracting paraphrases from a parallel corpus, in: Proceedings of the 39th annual meeting of the Association for Computational Linguistics, 2001, pp. 50–57.

[38] N. Madnani, B. J. Dorr, Generating phrasal and sentential paraphrases: A survey of data-driven methods, Computational Linguistics 36 (3) (2010) 341–387. `doi:10.1162/coli_a_00002`.
URL `https://www.aclweb.org/anthology/J10-3003`

[39] C. Coulombe, Text data augmentation made simple by leveraging nlp cloud apis, arXiv preprint arXiv:1812.04718 (2018).

[40] G. Miller, Wordnet: a lexical database for english, Commun. ACM 38 (1995) 39–41.

[41] J. Mueller, A. Thyagarajan, Siamese recurrent architectures for learning sentence similarity, in: AAAI, 2016.

[42] G. Daval-Frerot, Y. Weis, WMD at SemEval-2020 tasks 7 and 11: Assessing humor and propaganda using unsupervised data augmentation, in: Proceedings of the Fourteenth Workshop on Semantic Evaluation, International Committee for Computational Linguistics, Barcelona (online), 2020, pp. 1865–1874.
URL `https://www.aclweb.org/anthology/2020.semeval-1.246`

[43] X. Dai, H. Adel, An analysis of simple data augmentation for named entity recognition, in: COLING, 2020.

[44] D. Zhang, T. Li, H. Zhang, B. Yin, On data augmentation for extreme multi-label classification, ArXiv abs/2009.10778 (2020).

[45] X. Zuo, Y. Chen, K. Liu, J. Zhao, Knowdis: Knowledge enhanced data augmentation for event causality detection via distant supervision, in: COLING, 2020.

[46] K. Schuler, M. Palmer, Verbnet: a broad-coverage, comprehensive verb lexicon, 2005.

[47] C. F. Baker, C. J. Fillmore, J. B. Lowe, The Berkeley FrameNet project, in: 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1, Association for Computational Linguistics, Montreal, Quebec, Canada, 1998, pp. 86–90. doi:10.3115/980845.980860.
URL https://aclanthology.org/P98-1013

[48] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, arXiv preprint arXiv:1310.4546 (2013).

[49] S. Liu, K. Lee, I. Lee, Document-level multi-topic sentiment classification of email data with bilstm and data augmentation, Knowl. Based Syst. 197 (2020) 105918.

[50] D. Ramirez-Echavarria, A. Bikakis, L. Dickens, R. Miller, A. Vlachidis, On the effects of knowledge-augmented data in word embeddings, ArXiv abs/2010.01745 (2020).

[51] J. M. Tapia-Téllez, H. Escalante, Data augmentation with transformers for text classification, in: MICAI, 2020.

[52] D. Lowell, B. Howard, Z. C. Lipton, B. C. Wallace, Unsupervised data augmentation with naive augmentation and without unlabeled data, ArXiv abs/2010.11966 (2020).

[53] D. Palomino, J. E. O. Luna, Palomino-ochoa at tass 2020: Transformer-based data augmentation for overcoming few-shot learning, in: IberLEF@SEPLN, 2020.

[54] S. Kobayashi, Contextual augmentation: Data augmentation by words with paradigmatic relations, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 452–457. doi:10.18653/v1/N18-2072.
URL https://aclanthology.org/N18-2072

[55] M. Fadaee, A. Bisazza, C. Monz, Data augmentation for low-resource neural machine translation, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 567–573. doi:10.18653/v1/P17-2090.
URL https://aclanthology.org/P17-2090

[56] M. Dehouck, C. Gómez-Rodríguez, Data augmentation via subtree swapping for dependency parsing of low-resource languages, in: COLING, 2020.

[57] A. W. Yu, D. Dohan, M.-T. Luong, R. Zhao, K. Chen, M. Norouzi, Q. V. Le, Qanet: Combining local convolution with global self-attention for reading comprehension, ArXiv abs/1804.09541 (2018).

[58] A. R. Fabbri, S. Han, H. Li, H. Li, M. Ghazvininejad, S. R. Joty, D. Radev, Y. Mehdad, Improving zero and few-shot abstractive summarization with intermediate fine-tuning and data augmentation, ArXiv abs/2010.12836 (2021).

[59] M. Ibrahim, M. Torki, N. El-Makky, Alexu-backtranslation-tl at semeval-2020 task 12: Improving offensive language detection using data augmentation and transfer learning, in: SEMEVAL, 2020.

[60] S. Longpre, Y. Wang, C. DuBois, How effective is task-agnostic data augmentation for pretrained transformers?, in: Findings of the Association for Computational Linguistics: EMNLP 2020, Association for Computational Linguistics, Online, 2020, pp. 4401–4411. `doi:10.18653/v1/2020.findings-emnlp.394`.
URL `https://www.aclweb.org/anthology/2020.findings-emnlp.394`

[61] C. Rastogi, N. Mofid, F.-I. Hsiao, Can we achieve more with less? exploring data augmentation for toxic comment classification, ArXiv abs/2007.00875 (2020).

[62] A. Perevalov, A. Both, Augmentation-based answer type classification of the smart dataset, 2020.

[63] S. T. Aroyehun, A. Gelbukh, Aggression detection in social media: Using deep neural networks, data augmentation, and pseudo labeling, in: Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018), Association for Computational Linguistics, Santa Fe, New Mexico, USA, 2018, pp. 90–97.
URL `https://aclanthology.org/W18-4411`

[64] T. Nugent, N. Stelea, J. L. Leidner, Detecting esg topics using domain-specific language models and data augmentation approaches, ArXiv abs/2010.08319 (2020).

[65] Y. Qu, D. Shen, Y. Shen, S. Sajeev, J. Han, W. Chen, Coda: Contrast-enhanced and diversity-promoting data augmentation for natural language understanding, ArXiv abs/2010.08670 (2021).

[66] V. Barrière, A. Balahur, Improving sentiment analysis over non-english tweets using multilingual transformers and automatic translation for data-augmentation, in: COLING, 2020.

[67] Y. Hou, S. Chen, W. Che, C. Chen, T. Liu, C2c-genda: Cluster-to-cluster generation for data augmentation of slot filling, ArXiv abs/2012.07004 (2020).

[68] T. Kober, J. Weeds, L. Bertolini, D. J. Weir, Data augmentation for hypernymy detection, in: EACL, 2021.

[69] J. Wang, H.-C. Chen, R. Radach, A. Inhoff, Reading Chinese script: A cognitive analysis, Psychology Press, 1999.

[70] W. Peng, C. Huang, T. Li, Y. Chen, Q. Liu, Dictionary-based data augmentation for cross-domain neural machine translation, ArXiv abs/2004.02577 (2020).

[71] X. Wang, H. Pham, Z. Dai, G. Neubig, SwitchOut: an efficient data augmentation algorithm for neural machine translation, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 856–861. `doi:10.18653/v1/D18-1100`.
URL `https://aclanthology.org/D18-1100`

[72] L. Qin, M. Ni, Y. Zhang, W. Che, Cosda-ml: Multi-lingual code-switching data augmentation for zero-shot cross-lingual nlp, ArXiv abs/2006.06402 (2020).

[73] X. Song, L. Zang, Y. Su, X. Wu, J. Han, S. Hu, Data augmentation for copy-mechanism in dialogue state tracking, in: ICCS, 2021.

[74] H. Zhang, M. Cisse, Y. N. Dauphin, D. Lopez-Paz, mixup: Beyond empirical risk minimization, arXiv preprint arXiv:1710.09412 (2017).

[75] Y. Cheng, L. Jiang, W. Macherey, Robust neural machine translation with doubly adversarial inputs, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 4324–4333. `doi:10.18653/v1/P19-1425`.
URL `https://aclanthology.org/P19-1425`

[76] L. Sun, C. Xia, W. Yin, T. Liang, P. S. Yu, L. He, Mixup-transformer: Dynamic data augmentation for nlp tasks, in: COLING, 2020.

[77] M. S. BARI, M. Mohiuddin, S. R. Joty, Multimix: A robust data augmentation strategy for cross-lingual nlp, ArXiv abs/2004.13240 (2020).

[78] C. Si, Z. Zhang, F. Qi, Z. Liu, Y. Wang, Q. Liu, M. Sun, Better robustness by more coverage: Adversarial training with mixup augmentation for robust fine-tuning, ArXiv abs/2012.15699 (2020).

[79] J. Chen, Z. Wang, R. Tian, Z. Yang, D. Yang, Local additivity based data augmentation for semi-supervised ner, in: EMNLP, 2020.

[80] X. Mou, B. Sigouin, I. Steenstra, H. Su, Multimodal dialogue state tracking by qa approach with data augmentation, ArXiv abs/2007.09903 (2020).

[81] M. Shakeel, A. Karim, I. Khan, A multi-cascaded model with data augmentation for enhanced paraphrase detection in short texts, ArXiv abs/1912.12068 (2020).

[82] Y. Yin, L. Shang, X. Jiang, X. Chen, Q. Liu, Dialog state tracking with reinforced data augmentation, in: AAAI, 2020.

[83] R. Sennrich, B. Haddow, A. Birch, 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), Association for Computational Linguistics, Berlin, Germany, 2016, pp. 86–96. doi:10.18653/v1/P16-1009.
URL https://aclanthology.org/P16-1009

[84] R. Sennrich, B. Haddow, A. Birch, Improving neural machine translation models with monolingual data, ArXiv abs/1511.06709 (2016).

[85] Y. Chen, S. Lu, F. Yang, X. Huang, X. Fan, C. Guo, Pattern-aware data augmentation for query rewriting in voice assistant systems, ArXiv abs/2012.11468 (2020).

[86] K. M. Yoo, Y. Shin, S. goo Lee, Data augmentation for spoken language understanding via joint variational generation, in: AAAI, 2019.

[87] H. Q. Abonizio, S. B. Junior, Pre-trained data augmentation for text classification, in: BRACIS, 2020.

[88] V. Sanh, L. Debut, J. Chaumond, T. Wolf, Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, ArXiv abs/1910.01108 (2019).

[89] B. Tarján, G. Szaszák, T. Fegyó, P. Mihajlik, Deep transformer based data augmentation with subword units for morphologically rich online asr, arXiv preprint arXiv:2007.06949 (2020).

[90] L. Miao, M. Last, M. Litvak, Twitter data augmentation for monitoring public opinion on covid-19 intervention measures, in: NLP4COVID@EMNLP, 2020.

[91] Y. Yang, N. Jin, K. Lin, M. Guo, D. M. Cer, Neural retrieval for question answering with cross-attention supervised data augmentation, in: ACL/I-JCNLP, 2020.

[92] Z. Zhao, E. Papalexakis, X. Ma, Learning physical common sense as knowledge graph completion via bert data augmentation and constrained tucker factorization, in: EMNLP, 2020.

[93] J. Wei, K. Zou, Eda: Easy data augmentation techniques for boosting performance on text classification tasks, ArXiv abs/1901.11196 (2019).

[94] Z. Hu, B. Tan, R. Salakhutdinov, T. M. Mitchell, E. Xing, Learning data manipulation for augmentation and weighting, ArXiv abs/1910.12795 (2019).

[95] R. Liu, G. Xu, C. Jia, W. Ma, L. Wang, S. Vosoughi, Data boost: Text data augmentation through reinforcement learning guided conditional generation, in: EMNLP, 2020.

[96] S. Shehnepoor, R. Togneri, W. Liu, M. Bennamoun, Gangster: A fraud review detector based on regulated gan with data augmentation, ArXiv abs/2006.06561 (2020).

[97] Che,Wanxiang and Guo,Jiang and Cui,Yiming, Natural language processing: methods based on pre-trained models, Electronic Industry Press, 2021.

[98] C. Liu, D. Yu, BLCU-NLP at SemEval-2020 task 5: Data augmentation for efficient counterfactual detecting, in: Proceedings of the Fourteenth Workshop on Semantic Evaluation, International Committee for Computational Linguistics, Barcelona (online), 2020, pp. 633–639.
URL https://aclanthology.org/2020.semeval-1.81

[99] G. G. Sahin, M. Steedman, Data augmentation via dependency tree morphing for low-resource languages, in: EMNLP, 2018.

[100] J. Lun, J. Zhu, Y. Tang, M. Yang, Multiple data augmentation strategies for improving performance on automatic short answer scoring, in: AAAI, 2020.

[101] B. Xu, S. Qiu, J. Zhang, Y. Wang, X. Shen, G. de Melo, Data augmentation for multiclass utterance classification – a systematic study, in: COLING, 2020.

[102] Z. Wan, X. Wan, W. Wang, Improving grammatical error correction with data augmentation by editing latent representation, in: COLING, 2020.

[103] K. M. Yoo, H. Lee, F. Dernoncourt, T. Bui, W. Chang, S. goo Lee, Variational hierarchical dialog autoencoder for dialog state tracking data augmentation, in: EMNLP, 2020.

[104] H. Shi, K. Livescu, K. Gimpel, Substructure substitution: Structured data augmentation for nlp, ArXiv abs/2101.00411 (2021).

[105] Y. Xu, R. Jia, L. Mou, G. Li, Y. Chen, Y. Lu, Z. Jin, Improved relation classification by deep recurrent neural networks with data augmentation, in: COLING, 2016.

[106] R. Zhang, Y. Zheng, J. Shao, X.-X. Mao, Y. Xi, M. Huang, Dialogue distillation: Open-domain dialogue augmentation using unpaired data, in: EMNLP, 2020.

[107] A. Asai, H. Hajishirzi, Logic-guided data augmentation and regularization for consistent question answering, in: ACL, 2020.

[108] R. Zmigrod, S. J. Mielke, H. Wallach, R. Cotterell, Counterfactual data augmentation for mitigating gender stereotypes in languages with rich morphology, in: ACL, 2019.

[109] Y. Zhou, F. Dong, Y. Liu, Z. Li, J. Du, L. Zhang, Forecasting emerging technologies using data augmentation and deep learning, Scientometrics 123 (2020) 1–29.

[110] L. Yao, B. Yang, H. Zhang, B. Chen, W. Luo, Domain transfer based data augmentation for neural query translation, in: COLING, 2020.

[111] G. Chen, Y. Chen, Y. Wang, V. O. Li, Lexical-constraint-aware neural machine translation via data augmentation, in: C. Bessiere (Ed.), Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20, International Joint Conferences on Artificial Intelligence Organization, 2020, pp. 3587–3593, main track. `doi:10.24963/ijcai.2020/496`.
URL `https://doi.org/10.24963/ijcai.2020/496`

[112] I. Staliunaite, P. Gorinski, I. Iacobacci, Improving commonsense causal reasoning by adversarial training and data augmentation, in: AAAI, 2021.

[113] F. Gao, J. Zhu, L. Wu, Y. Xia, T. Qin, X. Cheng, W. Zhou, T.-Y. Liu, Soft contextual data augmentation for neural machine translation, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 5539–5544. `doi:10.18653/v1/P19-1555`.
URL `https://aclanthology.org/P19-1555`

[114] M. Xia, X. Kong, A. Anastasopoulos, G. Neubig, Generalized data augmentation for low-resource translation, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 5786–5796. `doi:10.18653/v1/P19-1579`.
URL `https://aclanthology.org/P19-1579`

[115] N. Malandrakis, M. Shen, A. Goyal, S. Gao, A. Sethi, A. Metallinou, Controlled text generation for data augmentation in intelligent artificial agents, in: Proceedings of the 3rd Workshop on Neural Generation and Translation, Association for Computational Linguistics, Hong Kong, 2019, pp. 90–98. `doi:10.18653/v1/D19-5609`.
URL `https://aclanthology.org/D19-5609`

[116] R. Cao, R. K. Lee, Hategan: Adversarial generative-based data augmentation for hate speech detection, in: COLING, 2020.

[117] J. Kukačka, V. Golkov, D. Cremers, Regularization for deep learning: A taxonomy, arXiv preprint arXiv:1710.10686 (2017).

[118] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, Y. Bengio, Generative adversarial networks, ArXiv abs/1406.2661 (2014).

[119] J. Morris, E. Lifland, J. Y. Yoo, J. Grigsby, D. Jin, Y. Qi, TextAttack: A framework for adversarial attacks, data augmentation, and adversarial training in NLP, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Association for Computational Linguistics, Online, 2020, pp. 119–126. doi:10.18653/v1/2020.emnlp-demos.16.
URL https://aclanthology.org/2020.emnlp-demos.16

[120] B. Zheng, L. Dong, S. Huang, W. Wang, Z. Chi, S. Singhal, W. Che, T. Liu, X. Song, F. Wei, Consistency regularization for cross-lingual finetuning, in: ACL/IJCNLP, 2021.

[121] C. Raffel, N. M. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, ArXiv abs/1910.10683 (2020).

[122] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language models are few-shot learners, ArXiv abs/2005.14165 (2020).