Data Augmentation in NLP

2020-03-21

Xiachong Feng

Outline

- Why we need Data Augmentation?
- Data Augmentation in CV
- Widely Used Methods
  - EDA
  - Back-Translation
  - Contextual Augmentation
- **Methods based on Pre-trained Language Models.**
  - BERT
  - GPT
  - Seq2Seq (BART)
- Conclusion

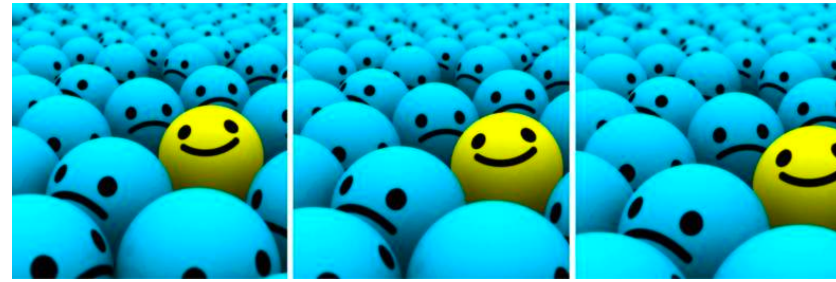
Why we need Data Augmentation?

- Few-shot Learning
- Imbalance labeled data
- Semi-supervise Learning
- ......

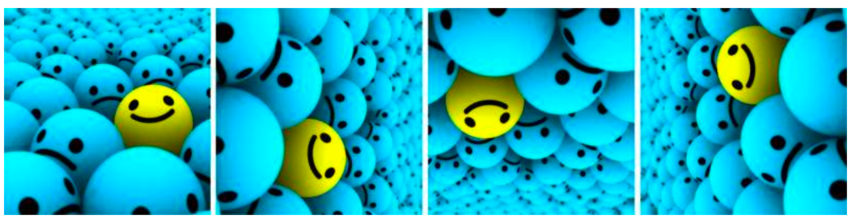
https://mp.weixin.qq.com/s/CHSDi2LpDOLMjWOLXIvSAg

Data Augmentation in CV

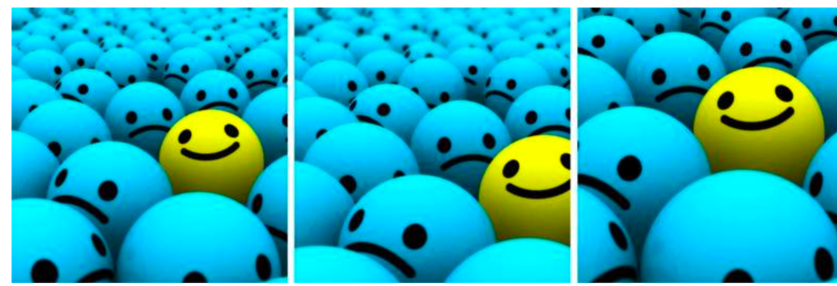
Flip : flip images horizontally and vertically.



Scale



Rotation



Crop : randomly sample a section from the original image

Gaussian Noise

https://medium.com/nanonets/how-to-use-deep-learning-when-you-have-limited-data-part-2-data-augmentation-c26971dc8ced

IF we apply them to NLP

I hate you !

! you hate I

Flip : flip horizontally and vertically.

I hate you !

Crop : randomly sample a section

Language is Discrete.

Widely Used Methods

- EDA
- Back-Translation
- Contextual Augmentation

EDA

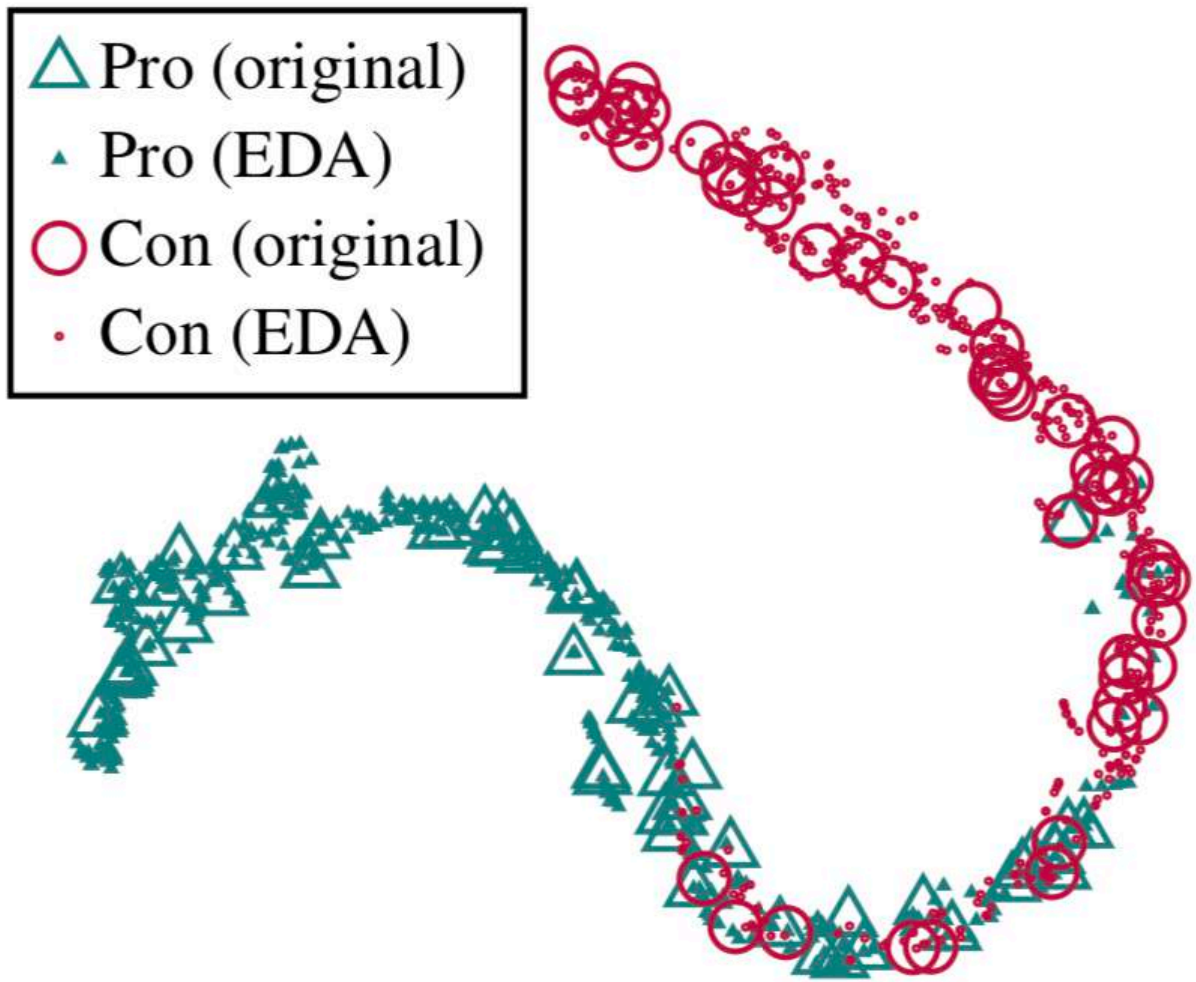
- EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks

1. **Synonym Replacement (SR):** Randomly choose n words from the sentence that are not stop words. Replace each of these words with one of its synonyms chosen at random.
2. **Random Insertion (RI):** Find a random synonym of a random word in the sentence that is not a stop word. Insert that synonym into a random position in the sentence. Do this n times.
3. **Random Swap (RS):** Randomly choose two words in the sentence and swap their positions. Do this n times.
4. **Random Deletion (RD):** Randomly remove each word in the sentence with probability p.

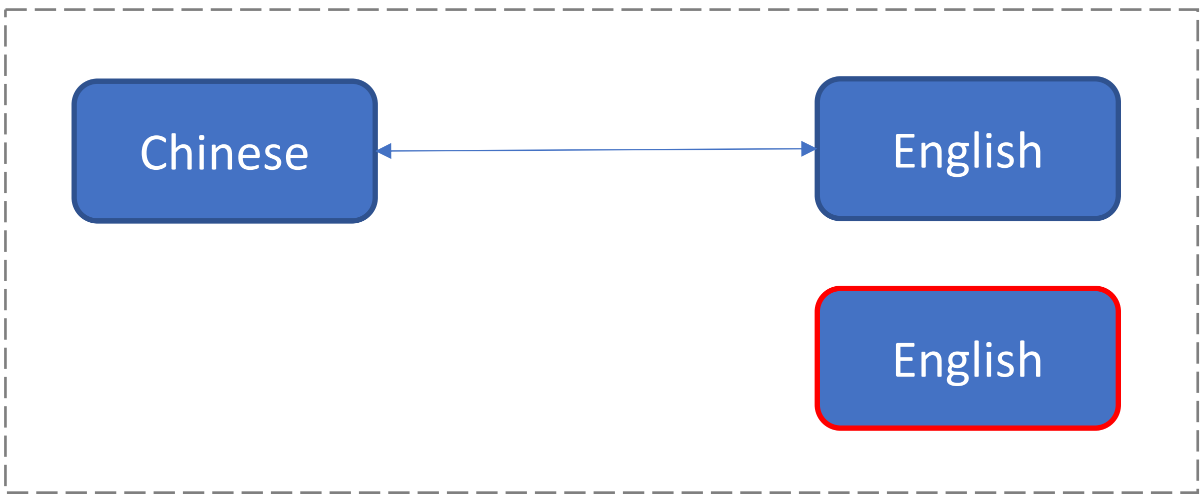
EDA Examples

Operation	Sentence
None	A sad, superior human comedy played out on the back roads of life.
SR	A ***lamentable***, superior human comedy played out on the ***backward*** road of life.
RI	A sad, superior human comedy played out on ***funniness*** the back roads of life.
RS	A sad, superior human comedy played out on ***roads*** back ***the*** of life.
RD	A sad, superior human out on the roads of life.

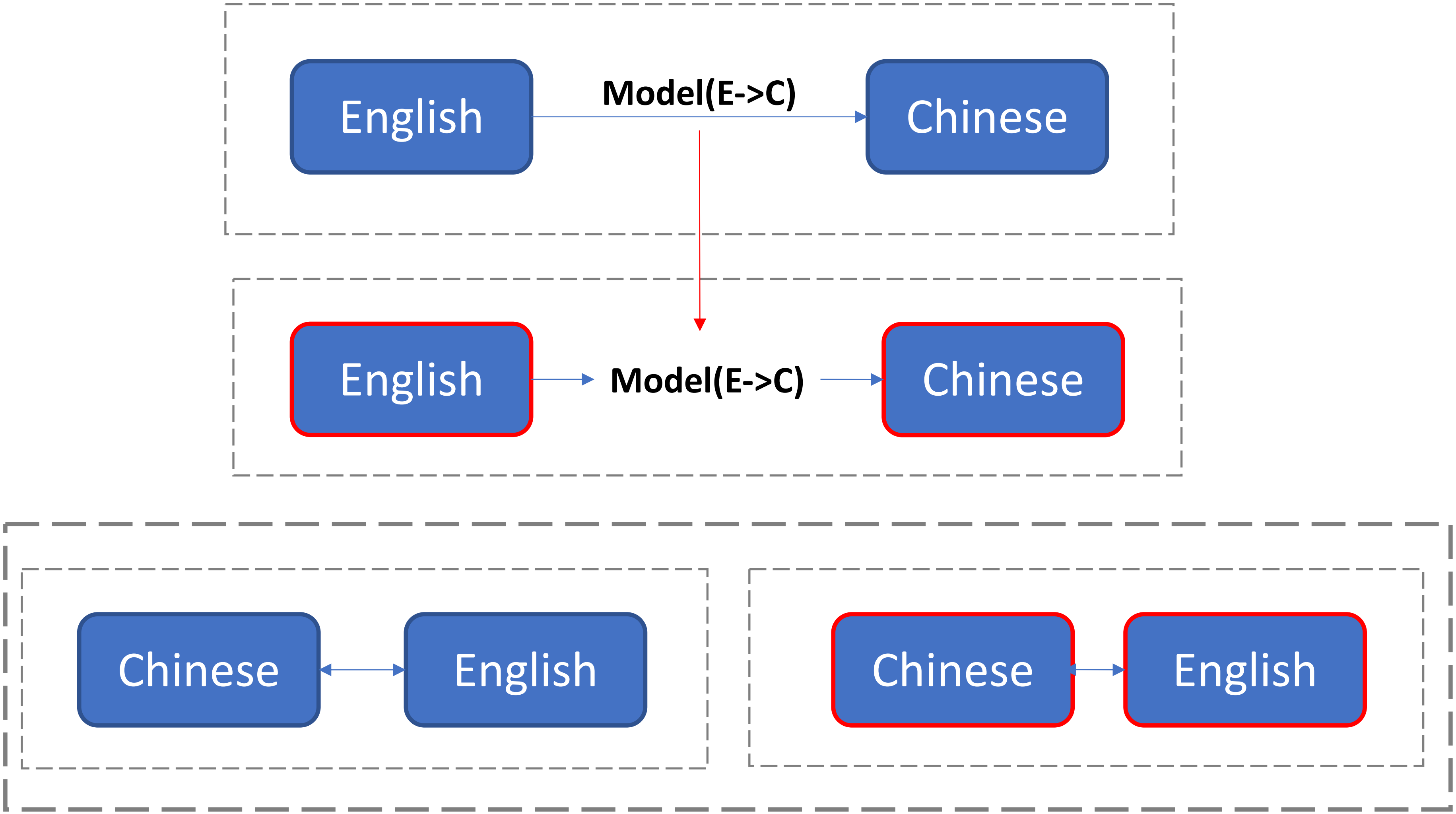
Conserving True Labels ?



Back-Translation



Back-Translation



Contextual Augmentation

- Contextual Augmentation: Data Augmentation by Words with Paradigmatic Relations *NAACL18*

- Disadvantages of the **Synonym Replacement**
  - Snonyms are very limited.
  - Synonym-based augmentation cannot produce numerous different patterns from the original texts.

Contextual Augmentation

the ***performances*** are fantastic
the ***films*** are fantastic
the ***movies*** are fantastic
the ***stories*** are fantastic

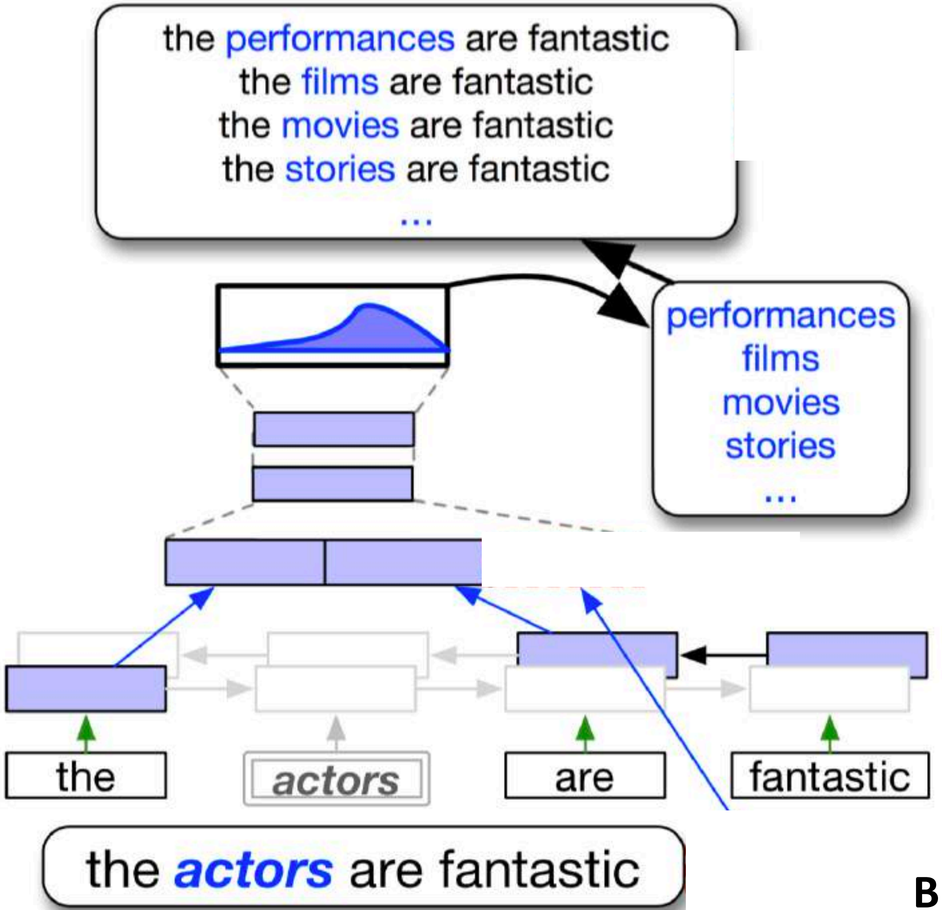
the ***performer*** are fantastic
the ***actress*** are fantastic

Synonym Replacement

Contextual Augmentation

the ***actors*** are fantastic

Contextual Augmentation



Sample

Bi-directional LSTM-RNN

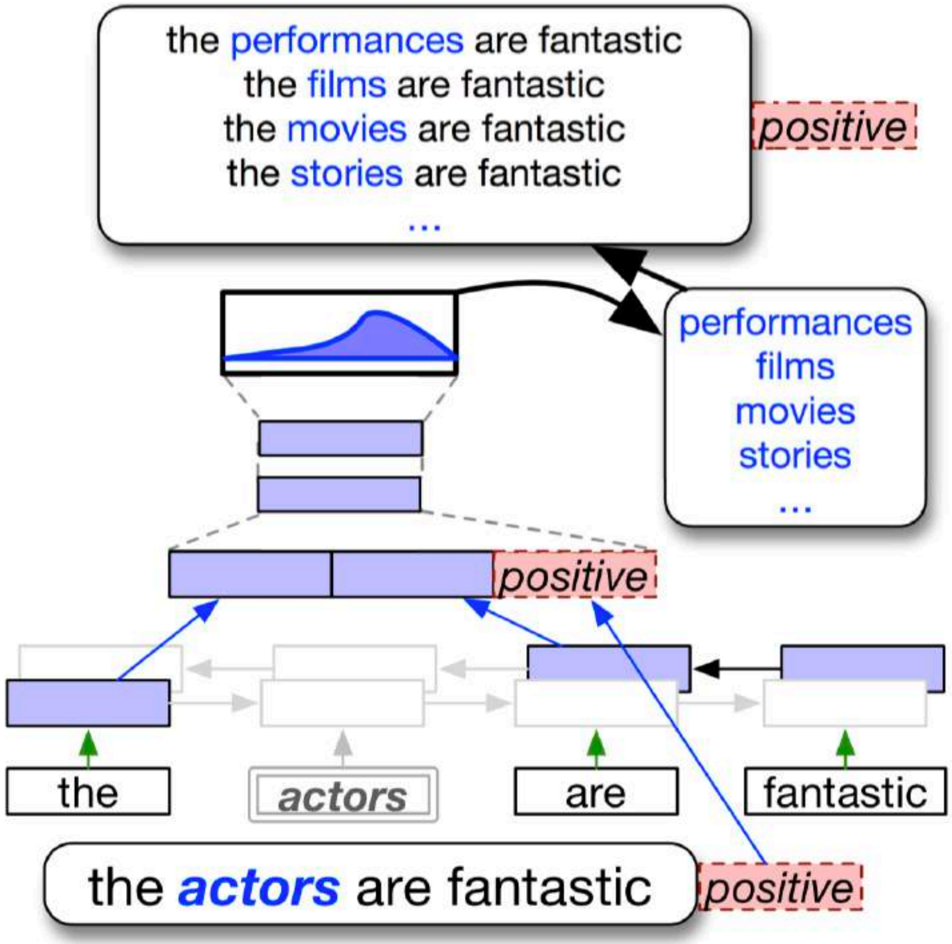
Pretrained on WikiText-103 corpus

Contextual Augmentation

the actors are ***good*** positive
the actors are ***entertaining*** positive
the actors are ***bad***
the actors are ***terrible***

the actors are ***fantastic*** positive

Contextual Augmentation



Further trained on each labeled dataset

Others

- Variational Auto Encoding (VAE)
- Paraphrasing
- Round-trip Translation
- Generative Adversarial Networks (GAN)

Methods based on Pre-trained Language Models

- Conditional BERT Contextual Augmentation *ICCS19*
- Do Not Have Enough Data? Deep Learning to the Rescue! *AAAI20*
- Data Augmentation using Pre-trained Transformer Models *Arxiv20*

Methods based on Pre-trained Language Models

(4) **Knowledge Transfer Beyond Fine-tuning** Currently, fine-tuning is the dominant method to transfer PTMs' knowledge to downstream tasks, but one deficiency is its parameter inefficiency: every downstream task has its own fine-tuned parameters. An improved solution is to fix the original parameters of PTMs and by adding small fine-tunable adaption modules for specific task [149, 61]. Thus, we can use a shared PTM to serve multiple downstream tasks. Indeed, mining knowledge from PTMs can be more flexible, such as feature extraction, knowledge distillation [195], data augmentation [185, 84], using PTMs as external knowledge [125] , and so on. More efficient methods are expected.

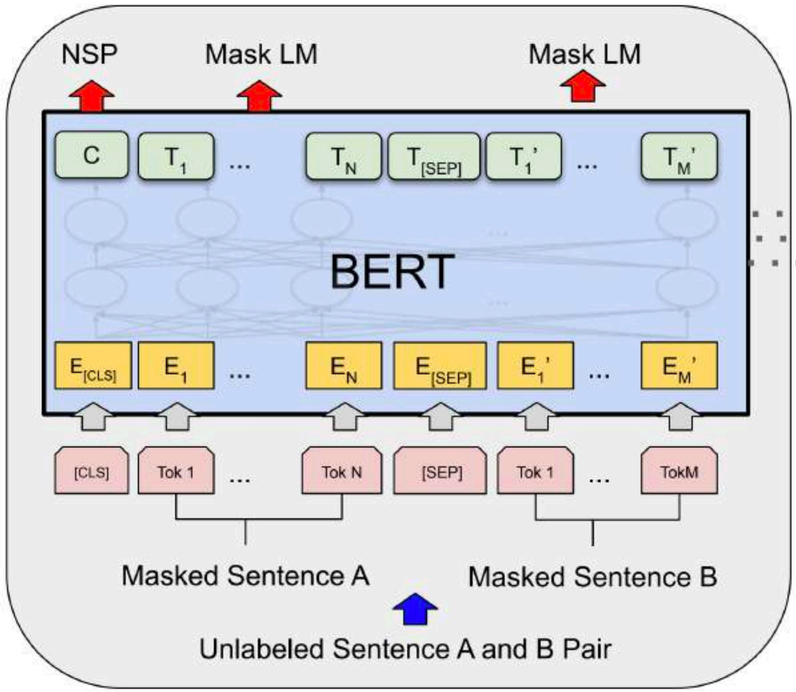
From Pre-trained Models for Natural Language Processing: A Survey

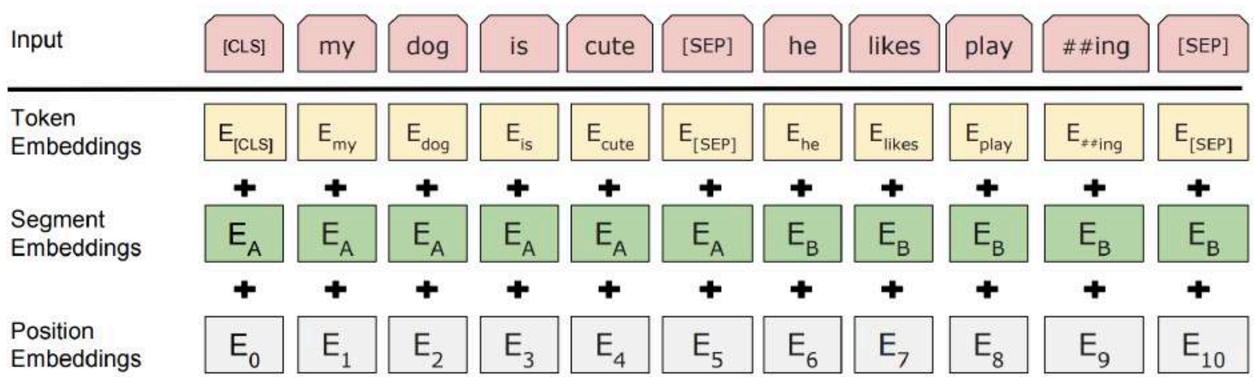
Conditional BERT Contextual Augmentation

ICCS19

Xing Wu, Shangwen Lv, Liangjun Zang, Jizhong Han, Songlin Hu,
Institute of Information Engineering, Chinese Academy of Sciences, Beijing,
China University of Chinese Academy of Sciences, Beijing, China

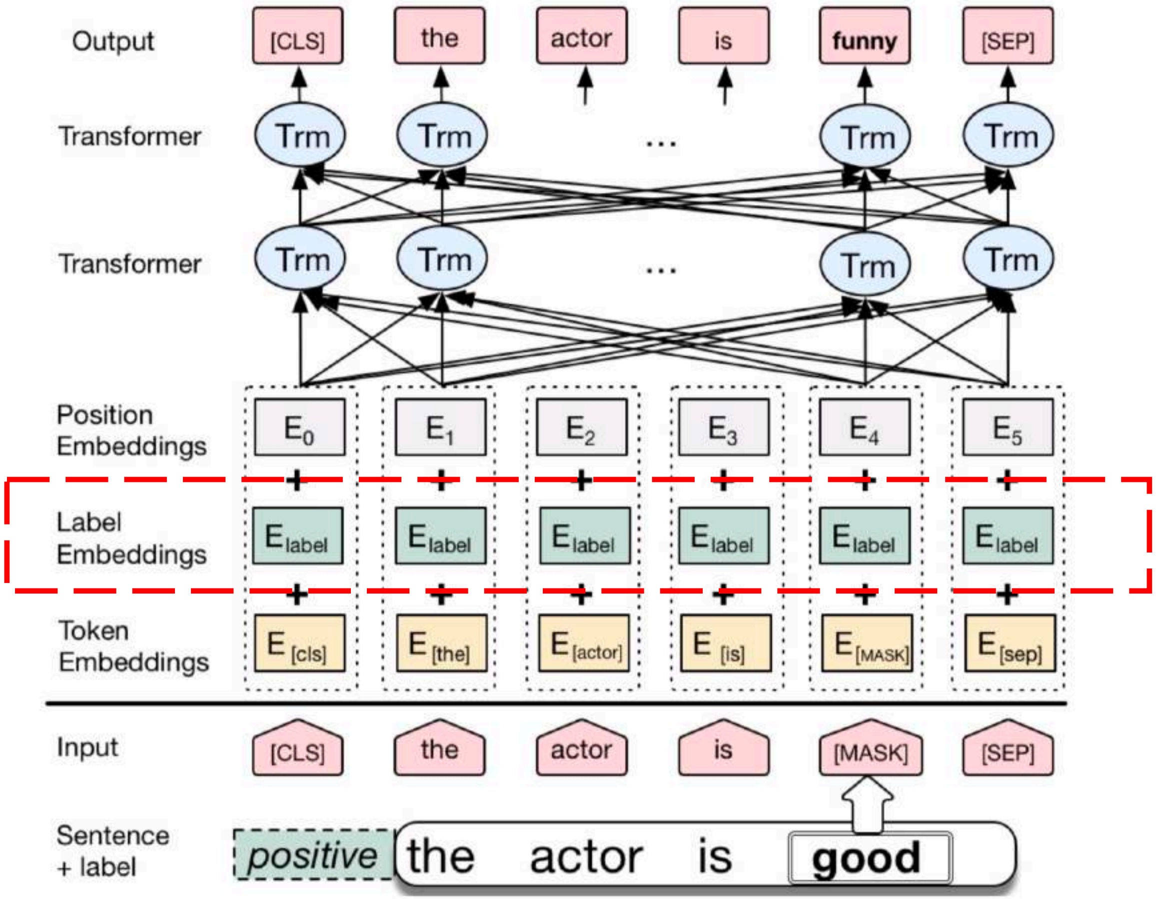
BERT





BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

C-BERT



Output
[CLS]
the
actor
is
funny
[SEP]
Transformer
Trm
...
Transformer
Trm
...
Position Embeddings
E_0
E_1
E_2
E_3
E_4
E_5
Label Embeddings
E_{label}
Token Embeddings
$E_{[cls]}$
$E_{[the]}$
$E_{[actor]}$
$E_{[is]}$
$E_{[MASK]}$
$E_{[sep]}$
Input
[CLS]
the
actor
is
[MASK]
[SEP]
Sentence + label
positive
the actor is good

Do Not Have Enough Data? Deep Learning to the Rescue !

AAAI20

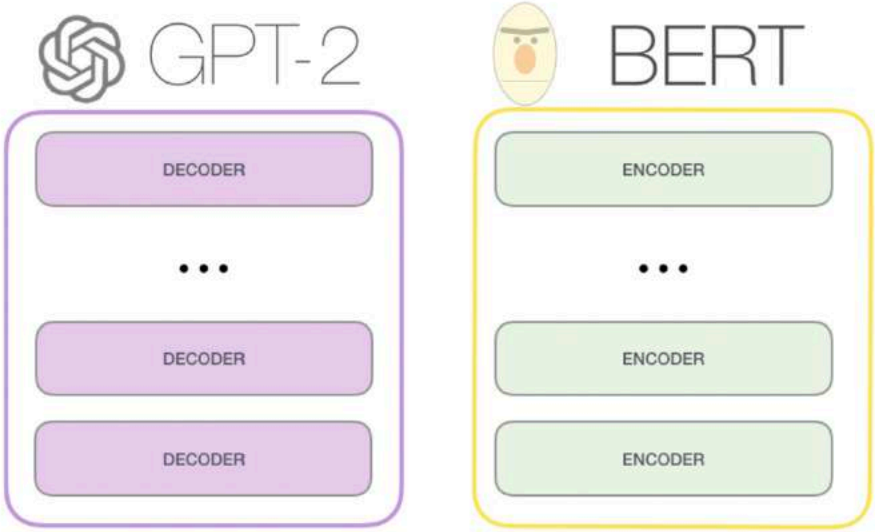
Ateret Anaby-Tavor, Boaz Carmeli, Esther Goldbraich, Amir Kantor, George Kour,
Segev Shlomov, Naama Tepper, Naama Zwerdling
IBM Research AI,
University of Haifa, Israel,
Technion - Israel Institute of Technology

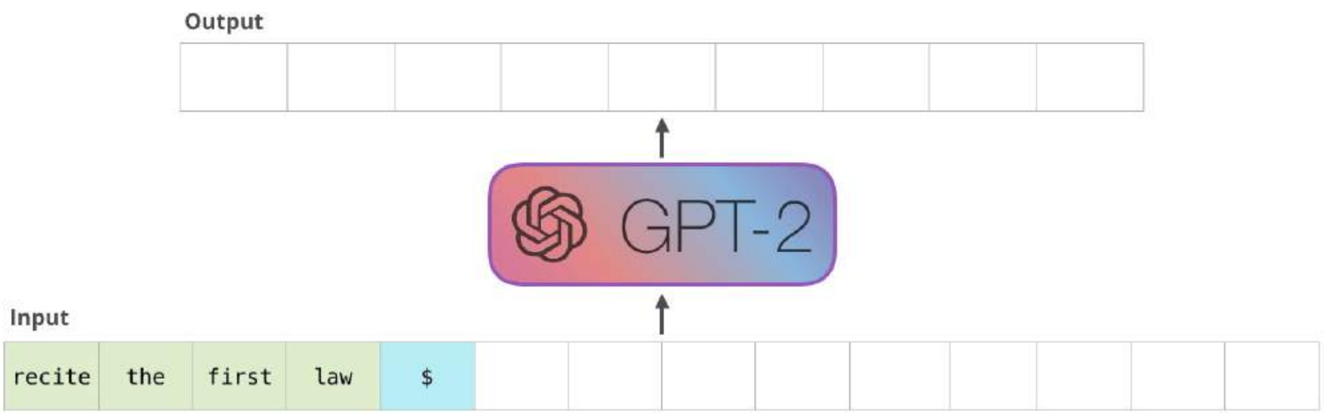
LAMBADA

- language-model-based data augmentation (LAMBADA)

- Disadvantages of **the Contextual Augmentation**
  - Presumably, methods that make only local changes will produce sentences with a structure similar to the original ones, thus yielding low corpus-level variability

GPT





https://gpt2.apps.allenai.org/?text=Joel%20is%20a

LAMBADA

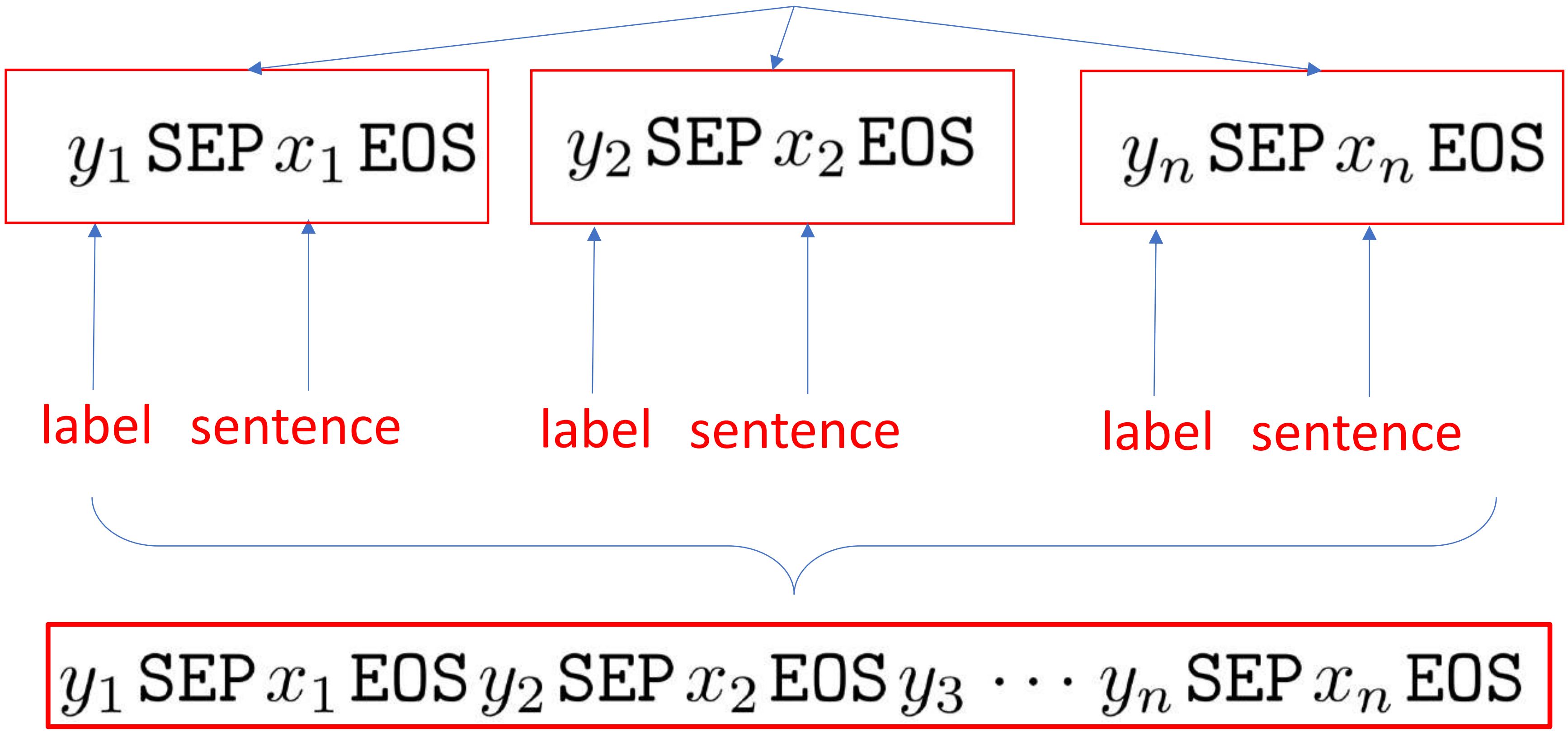
- The generative pre-training (GPT) model

Class label	Sentences
Flight time	what time is the last flight from san francisco to washington dc on continental
Aircraft	show me all the types of aircraft used flying from atl to dallas
City	show me the cities served by canadian airlines

LAMBADA

$$J_\theta = -\sum_j \log P_\theta(w^j | w^{j-k}, \ldots, w^{j-1})$$

$$D_{train} = \{(x_i, y_i)\}_{i=1}^{n}$$



LAMBADA

- Filter synthesized data

$$\mathcal{G}_{tuned} \longrightarrow D^*$$

$$D_{train} \longrightarrow \text{classifier } h$$

$$(x, y) \in D^* \longrightarrow \text{classifier } h$$

$$h(x) \neq y$$

$$h(x) = y$$

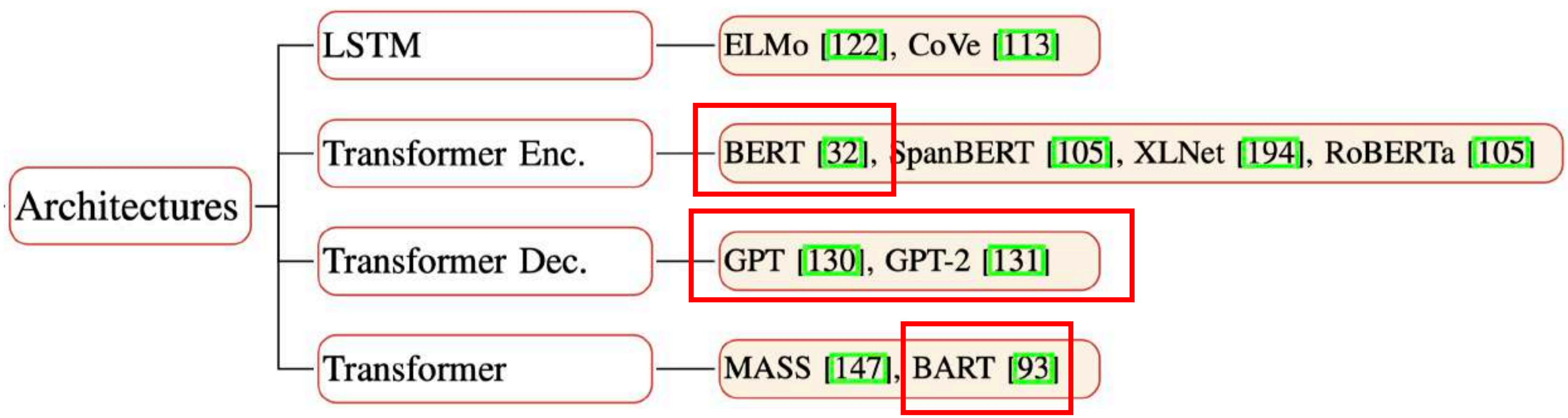
Confidence Score

Data Augmentation using Pre-trained Transformer Models

Arxiv20

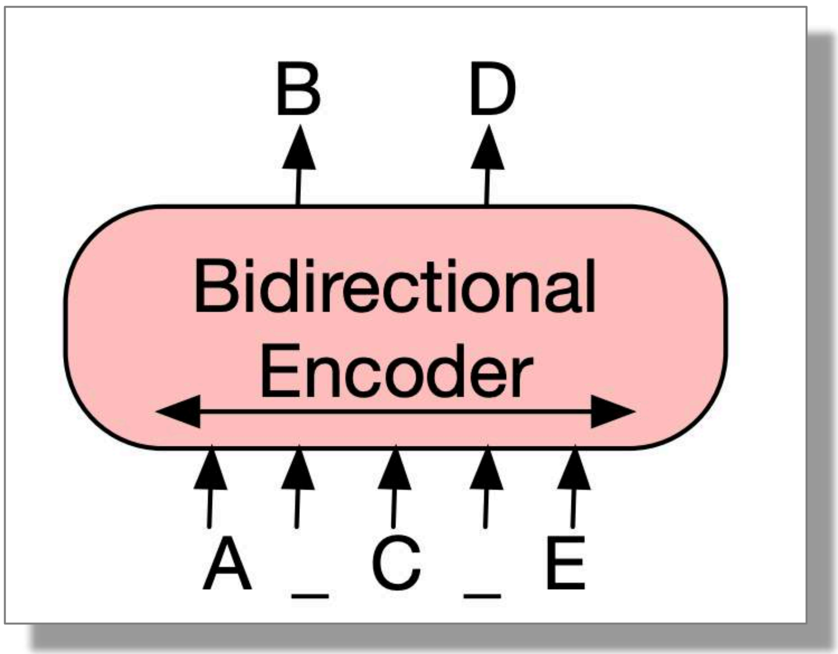
Varun Kumar, Alexa AI
Ashutosh Choudhary, Alexa AI
Eunah Cho, Alexa AI

Pre-trained Language Models

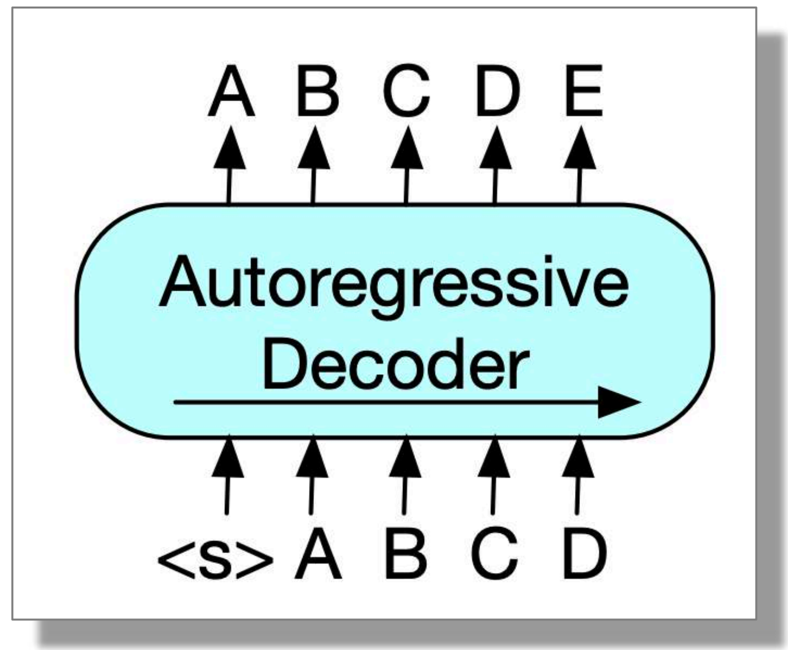


From Pre-trained Models for Natural Language Processing: A Survey

Pre-trained Language Models



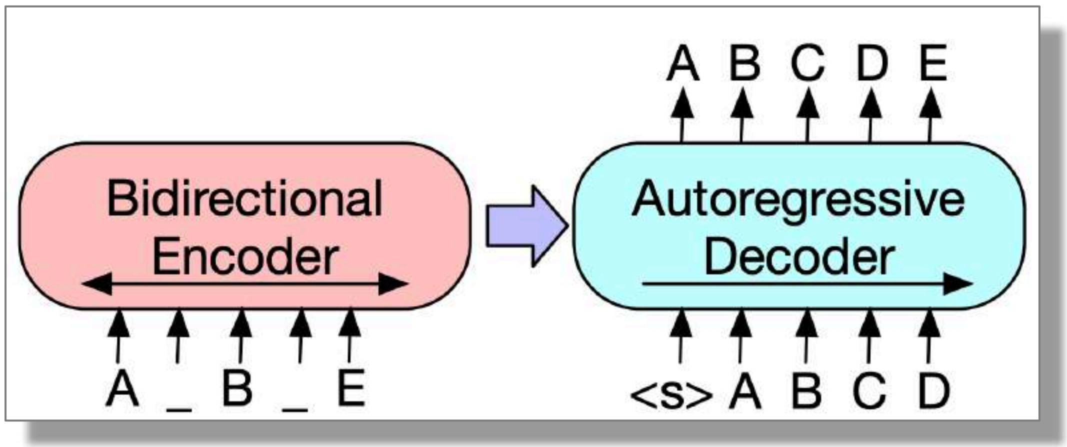
BERT

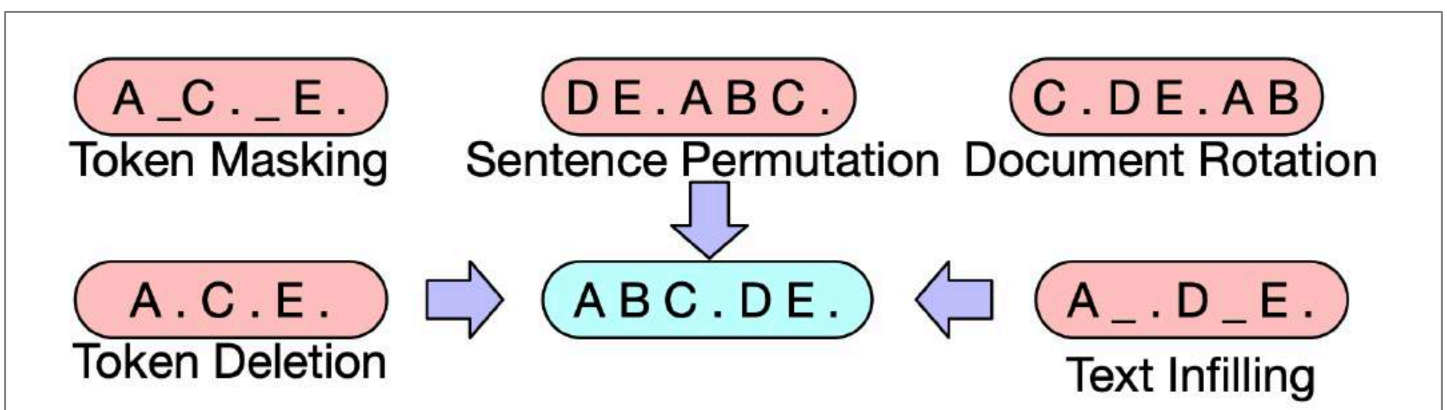


GPT-2

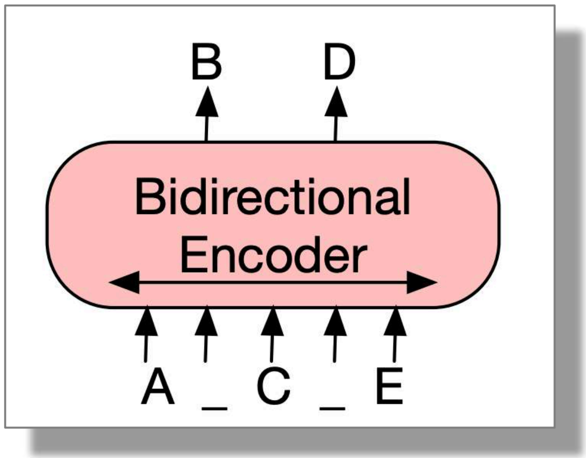
Pre-trained Language Models

- BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension

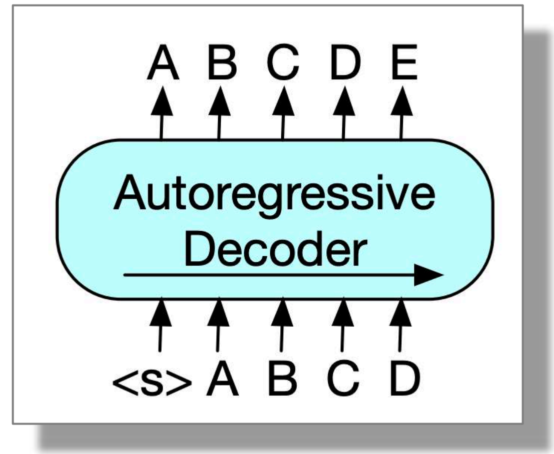




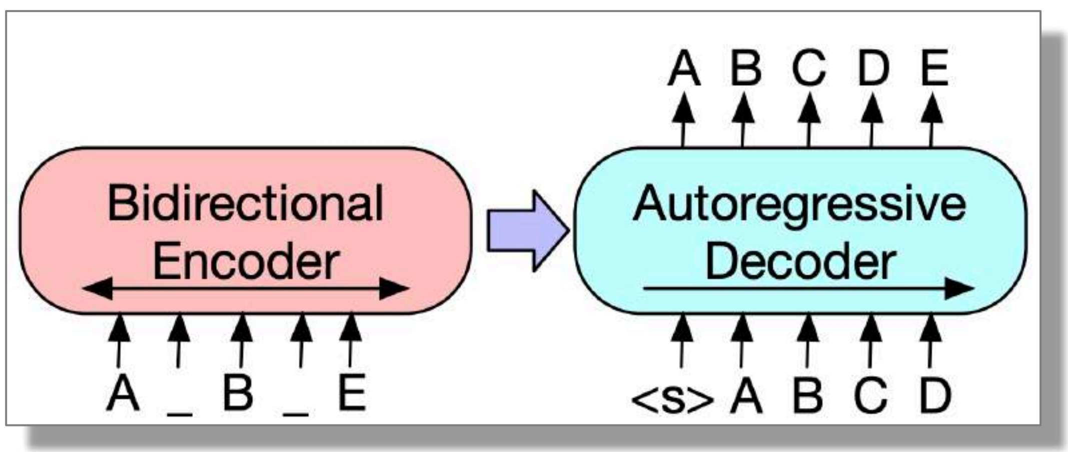
Unified Approach



autoencoder (AE) LM: BERT



auto-regressive (AR) : GPT2



seq2seq model: BART

Add Labels : Expend

`expand :` prepending label y_i to each sequence x_i in the training data and adding y_i to model vocabulary.

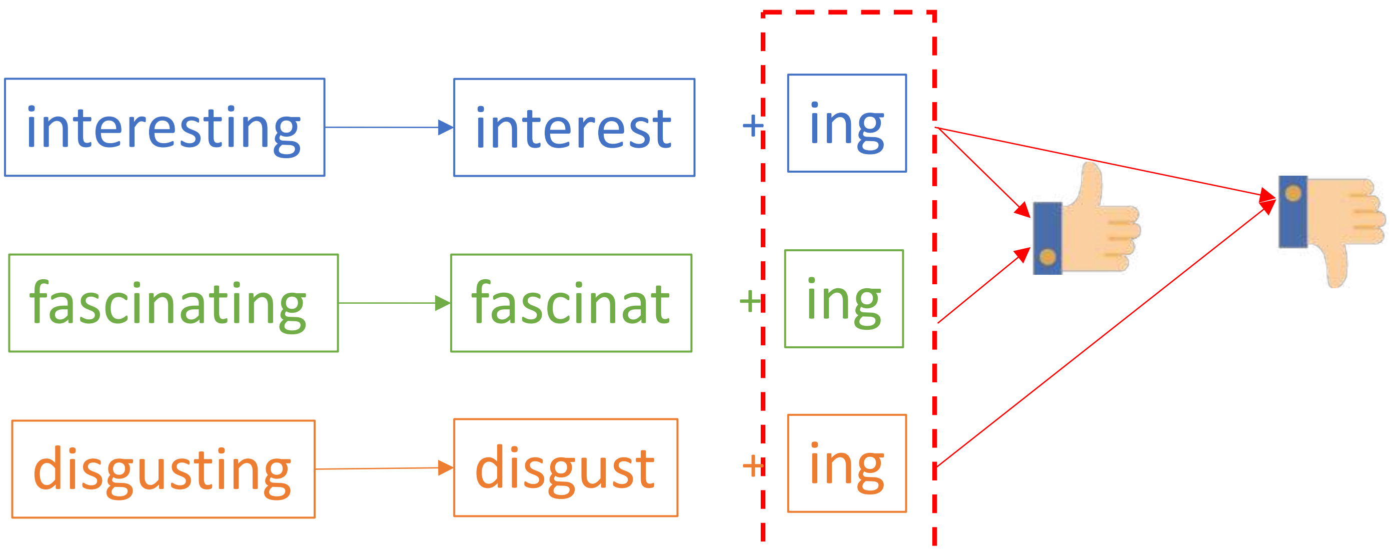
treats a label as a single token

interesting

Add Labels : Prepend

prepend : prepending label y_i to each sequence x_i in the training data without adding y_i to model vocabulary

the model may split label into multiple subword units



Fine-tuning

Type	PLM	Task	Labels	Model	Description
AE	BERT	MLM	prepend	BERT prepend	
			expand	BERT expand	

Fine-tuning

Type	PLM	Task	Labels	Model	Description
AE	BERT	MLM	prepend	BERT prepend	
			expand	BERT expand	
AR	GPT2	LM ($y_1 SEP x_1 EOS$ …)	prepend	GPT2	$y_i SEP$
				GPT2 context	$y_i SEP w_1 w_2 w_3$

Fine-tuning

Type	PLM	Task	Labels	Model	Description
AE	BERT	MLM	prepend	BERT prepend	
			expand	BERT expand	
AR	GPT2	LM ($y_1 SEP x_1 EOS$ ...)	prepend	GPT2	$y_i SEP$
				GPT2 context	$y_i SEP w_1 w_2 w_3$
Seq2Seq	BART	Denoising	prepend	BART word	Replace a token with mask
				BART span	Replace a continuous chunk words

Algorithm

Algorithm 1: Data Augmentation approach

Input : Training Dataset D_{train}

Pretrained model $G \in \{AE, AR, Seq2Seq\}$

Fine-tune G using D_{train} to obtain G_{tuned}

$D_{synthetic} \leftarrow \{\}$

foreach $\{x_i, y_i\} \in D_{train}$ **do**

  Synthesize s examples $\{\hat{x}_i, \hat{y}_i\}_p^1$ using G_{tuned}

  $D_{synthetic} \leftarrow D_{synthetic} \cup \{\hat{x}_i, \hat{y}_i\}_p^1$

end

Experiments

- **Baseline**
  - EDA
  - C-BERT
- **Task**
  - Sentiment Classification (SST2)
  - Intent Classification (SNIPS)
  - Question Classification (TREC)

Data	Label Names
SST-2	Positive, Negative
TREC	Description, Entity, Abbreviation, Human, Location, Numeric
SNIPS	PlayMusic, GetWeather, RateBook, SearchScreeningEvent, SearchCreativeWork, AddToPlaylist, BookRestaurant

	SST-2		SNIPS		TREC	
	All	1%	All	1%	All	1%
Train	6,229	61	13,084	127	5,406	51
Dev	693	10	700	35	546	30
Test	1,821		700		500	

five validation examples per class

Experiments

Extrinsic Evaluation

- Sentiment Classification
- Intent Classification
- Question Classification

Intrinsic Evaluation

- Semantic Fidelity
- Text Diversity

Extrinsic Evaluation

- Pre-trained BERT classifier

Model	SST2 (1%)	SNIPS (1%)	TREC (1%)
No Aug	59.08	57.95	30.65
EDA	59.09	77.46	29.57
CBERT	59.85	80.55	29.96
$BERT_{expand}$	61.24	79.75	31.88
$BERT_{prepend}$	61.90	81.31	30.28
GPT2	58.62	68.25	26.24
$GPT2_{context}$	59.39	77.73	31.54
$BART_{word}$	62.35	79.98	**37.48**
$BART_{span}$	**63.00**	**81.68**	37.25

Semantic Fidelity

- Training + Test dataset → BERT classifier

Model	SST2	SNIPS	TREC
CBERT	96.94	**97.32**	95.29
$BERT_{expand}$	96.17	96.80	92.68
$BERT_{prepend}$	**97.38**	**97.32**	**96.08**
GPT2	58.80	42.89	24.44
$GPT2_{context}$	69.84	85.04	73.33
$BART_{word}$	88.99	94.86	87.06
$BART_{span}$	89.39	94.87	86.80

Text Diversity

Model	SST2			SNIPS			TREC		
n-gram	1	2	3	1	2	3	1	2	3
CBERT	0.466	0.906	0.980	0.411	0.794	0.923	0.488	0.870	0.961
$BERT_{expand}$	0.490	0.914	0.983	**0.432**	0.809	0.934	0.511	0.881	0.965
$BERT_{prepend}$	0.465	0.907	0.981	0.415	0.798	0.932	0.487	0.873	0.956
GPT2	0.519	0.929	0.985	0.383	0.803	0.914	0.514	0.802	0.896
$GPT2_{context}$	0.524	0.933	0.994	0.354	0.781	0.938	**0.571**	0.872	0.954
$BART_{word}$	**0.537**	**0.941**	**0.995**	0.415	**0.813**	**0.948**	0.529	0.849	**0.971**
$BART_{span}$	0.527	0.936	**0.995**	0.408	0.798	0.934	0.502	**0.882**	0.965

Conclusion

- Data augmentation is useful.
- EDA, Back-translation,......
- PLM can be used for data augmentation.
- Generate new data is powerful than the replace-based method.
- Data Augmentation for Text Generation?

Thanks!