Data Augmentation in NLP

2020-03-21
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
- ......
Data Augmentation in CV

Flip: flip images horizontally and vertically.

Scale

Rotation

Crop: randomly sample a section from the original image

Gaussian Noise

IF we apply them to NLP

Flip: flip horizontally and vertically.

I hate you!  |  ! you hate I

Crop: randomly sample a section

I hate you!  |  I hate you!  |  I hate you!

Language is Discrete.
Widely Used Methods

• EDA
• Back-Translation
• Contextual Augmentation
• **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

<table>
<thead>
<tr>
<th>Operation</th>
<th>Sentence</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>A sad, superior human comedy played out on the back roads of life.</td>
</tr>
<tr>
<td>SR</td>
<td>A <em>lamentable</em>, superior human comedy played out on the <em>backward</em> road of life.</td>
</tr>
<tr>
<td>RI</td>
<td>A sad, superior human comedy played out on <em>funniness</em> the back roads of life.</td>
</tr>
<tr>
<td>RS</td>
<td>A sad, superior human comedy played out on <em>roads</em> back <em>the</em> of life.</td>
</tr>
<tr>
<td>RD</td>
<td>A sad, superior human out on the roads of life.</td>
</tr>
</tbody>
</table>
Conserving True Labels?
Back-Translation
Back-Translation

Diagram:

1. English → Model(E→C) → Chinese
2. Chinese → Model(E→C) → English
3. Chinese → English
4. English → Chinese
Contextual Augmentation

• Contextual Augmentation: Data Augmentation by Words with Paradigmatic Relations  NAACL18

• Disadvantages of the Synonym Replacement
  • Synonyms are very limited.
  • Synonym-based augmentation cannot produce numerous different patterns from the original texts.
Contextual Augmentation

- the *performer* are fantastic
- the *actress* are fantastic
- the performances are fantastic
- the *films* are fantastic
- the movies are fantastic
- the stories are fantastic
- the *actors* are fantastic

Syonym Replacement
Contextual Augmentation
Contextual Augmentation

Bi-directional LSTM-RNN
Pretrained on WikiText-103 corpus
the actors are *good*  
the actors are *entertaining*  
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 \textit{ICCS19}

• Do Not Have Enough Data? Deep Learning to the Rescue! \textit{AAAI20}

• Data Augmentation using Pre-trained Transformer Models \textit{Arxiv20}
(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 adaptation 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.
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: Pre-training of Deep Bidirectional Transformers for Language Understanding
C-BERT
Do Not Have Enough Data? Deep Learning to the Rescue!

AAAII20

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-2 and BERT models are shown in a diagram. The input is fed into the GPT-2 model, which then processes the data and outputs text or other information.
LAMBADA

- The generative pre-training (GPT) model

<table>
<thead>
<tr>
<th>Class label</th>
<th>Sentences</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flight time</td>
<td>what time is the last flight from san francisco to washington dc on continental</td>
</tr>
<tr>
<td>Aircraft</td>
<td>show me all the types of aircraft used flying from atl to dallas</td>
</tr>
<tr>
<td>City</td>
<td>show me the cities served by canadian airlines</td>
</tr>
</tbody>
</table>
LAMBADA

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

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

$y_1$ SEP $x_1$ EOS $y_2$ SEP $x_2$ EOS $y_n$ SEP $x_n$ EOS

$y_1$ SEP $x_1$ EOS $y_2$ SEP $x_2$ EOS $y_3 \cdots y_n$ SEP $x_n$ EOS
LAMBADA

• Filter synthesized data

\[ G_{tuned} \rightarrow D^* \]

\[ D_{train} \rightarrow \text{classifier } h \]

\[ (x, y) \in D^* \rightarrow \text{classifier } h \]

\[ h(x) \neq y \]

\[ h(x) = y \]

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

Architectures

- LSTM
  - ELMo [122], CoVe [113]
- Transformer Enc.
  - BERT [32], SpanBERT [105], XLNet [194], RoBERTa [105]
- Transformer Dec.
  - GPT [130], GPT-2 [131]
- Transformer
  - MASS [147], BART [93]
Pre-trained Language Models

BERT

Autoregressive Decoder

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
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.

- interesting → interest
- fascinating → fascinat
- disgusting → disgust
- ing
- ing
- ing
## Fine-tuning

<table>
<thead>
<tr>
<th>Type</th>
<th>PLM</th>
<th>Task</th>
<th>Labels</th>
<th>Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AE</td>
<td>BERT</td>
<td>MLM</td>
<td>prepend</td>
<td>BERT prepend</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>expand</td>
<td>BERT expand</td>
<td></td>
</tr>
</tbody>
</table>
## Fine-tuning

<table>
<thead>
<tr>
<th>Type</th>
<th>PLM</th>
<th>Task</th>
<th>Labels</th>
<th>Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AE</td>
<td>BERT</td>
<td>MLM</td>
<td>prepend</td>
<td>BERT prepend</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>expand</td>
<td>BERT expand</td>
<td></td>
</tr>
<tr>
<td>AR</td>
<td>GPT2</td>
<td>LM (y_1SEP x_1EOS \ldots)</td>
<td>prepend</td>
<td>GPT2 context</td>
<td>(y_iSEPw_1w_2w_3)</td>
</tr>
</tbody>
</table>
# Fine-tuning

<table>
<thead>
<tr>
<th>Type</th>
<th>PLM</th>
<th>Task</th>
<th>Labels</th>
<th>Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AE</td>
<td>BERT</td>
<td>MLM</td>
<td>prepend</td>
<td>BERT prepend</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>expand</td>
<td>BERT expand</td>
<td></td>
</tr>
<tr>
<td>AR</td>
<td>GPT2</td>
<td>LM $y_1SEP x_1EOS \ldots$</td>
<td>prepend</td>
<td>GPT2</td>
<td>$y_iSEP$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>GPT2 context</td>
<td>$y_iSEP w_1 w_2 w_3$</td>
</tr>
<tr>
<td>Seq2Seq</td>
<td>BART</td>
<td>Denoising</td>
<td>prepend</td>
<td>BART word</td>
<td>Replace a token with mask</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>BART span</td>
<td>Replace a continuous chunk words</td>
</tr>
</tbody>
</table>
Algorithm

Algorithm 1: Data Augmentation approach

Input: Training Dataset $D_{train}$
Pretrained model $G \in \{AE, AR, Seq2Seq\}$

1. Fine-tune $G$ using $D_{train}$ to obtain $G_{tuned}$
2. $D_{synthetic} \leftarrow \{\}$
3. foreach $\{x_i, y_i\} \in D_{train}$ do
   4. Synthesize $s$ examples $\{\hat{x}_i, \hat{y}_i\}_p^1$ using $G_{tuned}$
   5. $D_{synthetic} \leftarrow D_{synthetic} \cup \{\hat{x}_i, \hat{y}_i\}_p^1$
4. end
Experiments

- **Baseline**
  - EDA
  - C-BERT

- **Task**
  - Sentiment Classification (SST2)
  - Intent Classification (SNIPS)
  - Question Classification (TREC)

<table>
<thead>
<tr>
<th>Data</th>
<th>Label Names</th>
</tr>
</thead>
<tbody>
<tr>
<td>SST-2</td>
<td>Positive, Negative</td>
</tr>
<tr>
<td>TREC</td>
<td>Description, Entity, Abbreviation, Human, Location, Numeric</td>
</tr>
<tr>
<td>SNIPS</td>
<td>PlayMusic, GetWeather, RateBook, SearchScreeningEvent, SearchCreativeWork, AddTo-Playlist, BookRestaurant</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>SST-2</th>
<th>SNIPS</th>
<th>TREC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train</td>
<td>6,229</td>
<td>13,084</td>
<td>5,406</td>
</tr>
<tr>
<td>Dev</td>
<td>693</td>
<td>700</td>
<td>546</td>
</tr>
<tr>
<td>Test</td>
<td>1,821</td>
<td>700</td>
<td>500</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Model</th>
<th>SST2 (1%)</th>
<th>SNIPS (1%)</th>
<th>TREC (1%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Aug</td>
<td>59.08</td>
<td>57.95</td>
<td>30.65</td>
</tr>
<tr>
<td>EDA</td>
<td>59.09</td>
<td>77.46</td>
<td>29.57</td>
</tr>
<tr>
<td>CBERT</td>
<td>59.85</td>
<td>80.55</td>
<td>29.96</td>
</tr>
<tr>
<td>BERT$_{expand}$</td>
<td>61.24</td>
<td>79.75</td>
<td>31.88</td>
</tr>
<tr>
<td>BERT$_{prepend}$</td>
<td>61.90</td>
<td>81.31</td>
<td>30.28</td>
</tr>
<tr>
<td>GPT2</td>
<td>58.62</td>
<td>68.25</td>
<td>26.24</td>
</tr>
<tr>
<td>GPT2$_{context}$</td>
<td>59.39</td>
<td>77.73</td>
<td>31.54</td>
</tr>
<tr>
<td>BART$_{word}$</td>
<td>62.35</td>
<td>79.98</td>
<td>37.48</td>
</tr>
<tr>
<td>BART$_{span}$</td>
<td>63.00</td>
<td>81.68</td>
<td>37.25</td>
</tr>
</tbody>
</table>
Semantic Fidelity

- Training + Test dataset $\rightarrow$ BERT classifier

<table>
<thead>
<tr>
<th>Model</th>
<th>SST2</th>
<th>SNIPS</th>
<th>TREC</th>
</tr>
</thead>
<tbody>
<tr>
<td>CBERT</td>
<td>96.94</td>
<td>97.32</td>
<td>95.29</td>
</tr>
<tr>
<td>$\text{BERT}_{\text{expand}}$</td>
<td>96.17</td>
<td>96.80</td>
<td>92.68</td>
</tr>
<tr>
<td>$\text{BERT}_{\text{prepend}}$</td>
<td><strong>97.38</strong></td>
<td><strong>97.32</strong></td>
<td><strong>96.08</strong></td>
</tr>
<tr>
<td>GPT2</td>
<td>58.80</td>
<td>42.89</td>
<td>24.44</td>
</tr>
<tr>
<td>$\text{GPT2}_{\text{context}}$</td>
<td>69.84</td>
<td>85.04</td>
<td>73.33</td>
</tr>
<tr>
<td>$\text{BART}_{\text{word}}$</td>
<td>88.99</td>
<td>94.86</td>
<td>87.06</td>
</tr>
<tr>
<td>$\text{BART}_{\text{span}}$</td>
<td>89.39</td>
<td>94.87</td>
<td>86.80</td>
</tr>
</tbody>
</table>
Text Diversity

<table>
<thead>
<tr>
<th>Model</th>
<th>SST2</th>
<th>SNIPS</th>
<th>TREC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>n-gram</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CBERT</td>
<td>0.466</td>
<td>0.906</td>
<td>0.980</td>
</tr>
<tr>
<td>BERT_{expand}</td>
<td>0.490</td>
<td>0.914</td>
<td>0.983</td>
</tr>
<tr>
<td>BERT_{prepend}</td>
<td>0.465</td>
<td>0.907</td>
<td>0.981</td>
</tr>
<tr>
<td>GPT2</td>
<td>0.519</td>
<td>0.929</td>
<td>0.985</td>
</tr>
<tr>
<td>GPT2_{context}</td>
<td>0.524</td>
<td>0.933</td>
<td>0.994</td>
</tr>
<tr>
<td>BART_{word}</td>
<td><strong>0.537</strong></td>
<td>0.941</td>
<td><strong>0.995</strong></td>
</tr>
<tr>
<td>BART_{span}</td>
<td>0.527</td>
<td>0.936</td>
<td><strong>0.995</strong></td>
</tr>
</tbody>
</table>
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!