**Data Augmentation in NLP**

**Introduction to Text Augmentation**

More data we have, better performance we can achieve. However, it is very too luxury to annotate large amount of training data. Therefore, proper data augmentation is useful to boost up your model performance. Augmentation is very popular in computer vision area. Image can be augmented easily by flipping, adding salt etc via image augmentation library such as [imgaug](https://github.com/aleju/imgaug). It is proved that augmentation is one of the anchor to success of computer vision model.

In natural language processing (NLP) field, it is hard to augmenting text due to high complexity of language. Not every word we can replace it by others such as a, an, the. Also, not every word has synonym. Even changing a word, the context will be totally difference. On the other hand, generating augmented image in computer vision area is relative easier. Even introducing noise or cropping out portion of image, model can still classify the image.

Given that we do not have unlimited resource to build training data by human, authors tried different methods to achieve same goals which is generating more data for model training. In this story, we explore different authors how they leverage augmentation to tickle NLP tasks via generating more text data to boost up the models. The following story will cover:

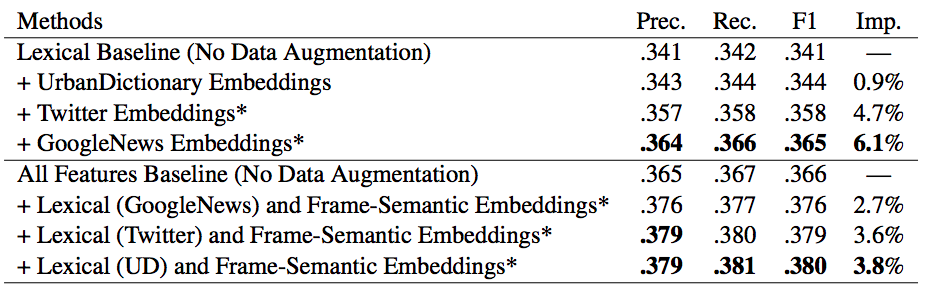
* Thesaurus
* Word Embeddings
* Back Translation
* Contextualized Word Embeddings
* Text Generation

**Thesaurus**

Zhang et al. introduced synonyms [Character-level Convolutional Networks for Text Classification](https://arxiv.org/pdf/1509.01626.pdf). During the experiment, they found that one of the useful way to do text augmentation is replacing words or phrases with their synonyms. Leverage existing thesaurus help to generate lots of data in a short time. Zhang et al. select a word and replace it by synonyms according to geometric distribution.

**Word Embeddings**

Wang and Yang introduced word similar calculation in [That’s So Annoying!!!: A Lexical and Frame-Semantic Embedding Based Data Augmentation Approach to Automatic Categorization of Annoying Behaviors using #petpeeve Tweets](https://aclweb.org/anthology/D15-1306). In the paper, Wang and Yang proposed to use k-nearest-neighbor (KNN) and cosine similarity to find the similar word for replacement.



Imp.: relative improvement to the baseline without data augmentation (Wang and Yang, 2015)

Alternatively, we can leverages pre-trained [classic word embeddings](https://towardsdatascience.com/3-silver-bullets-of-word-embedding-in-nlp-10fa8f50cc5a) such as word2vec, GloVe and fasttext to perform similarity search.

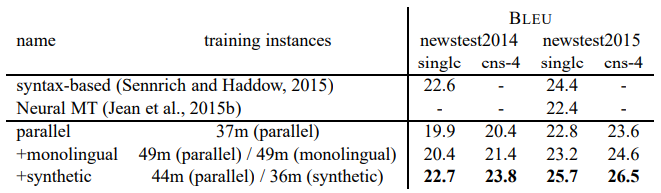


Most similar words of “fox” among classical word embeddings models

**Back Translation**

English is one of the language which having lots of training data for translation while some language may not has enough data for train a machine translation model. Sennrich et al. used back-translation method to generate more training data to improve translation model performance.

Given that we want to train a model for translating English (source language) → Cantonese (target language) and there is not enough training data for Cantonese. Back-translation is translating target language to source language and mixing both original source sentence and back-translated sentence to train a model. So the number of training data from source language to target language can be increased.

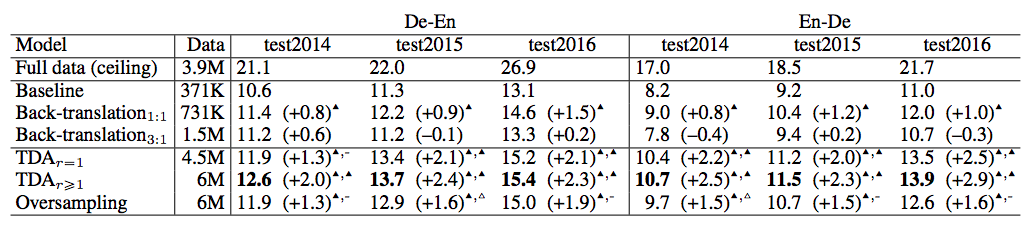


Performance result with and without text augmentation (Sennrich et al., 2016)

**Contextualized Word Embeddings**

Rather than using static word embeddings, Fadaee et al. use contextualized word embeddings to replace target word. They use this text augmentation to validate machine translation model in [Data Augmentation for Low-Resource Neural Machine Translation](https://arxiv.org/pdf/1705.00440.pdf) .

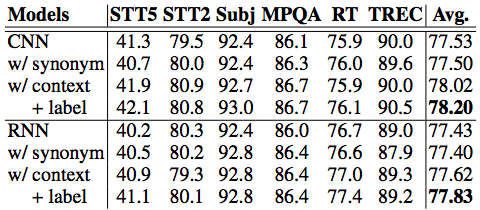
The proposed approach is TDA which stands for Translation Data Augmentation. From the experiment, it showed that machine translation model is improved by leveraging text augmentation.



Performance result with and without text augmentation (Fadaee et al., 2017)

Kobayashi propose to use a bi-directional language model in [Contextual Augmentation: Data Augmentation by Words with Paradigmatic Relation](https://arxiv.org/pdf/1805.06201.pdf). After selected target word, the model will predict possible replacement by giving surrounding words. As the target will exist in any position of sentence, bi-directional architecture is used to learn both rightward and leftward context.

Kobayashi verified the language model approach with CNN and RNN on six dataset and the result is positive. Text augmentation helps to further improve the NLP model result.

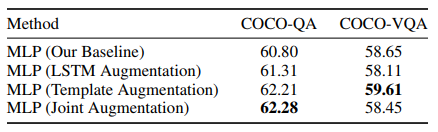


Performance result with and without text augmentation (Kobayashi 2018)

**Text Generation**

Kafle et al. introduced a different approach which generate augmented data by generating it in [Data Augmentation for Visual Question Answering](https://aclweb.org/anthology/W17-3529). Different from previous approach, Kafle et al. approaches do not replace single of few words but generating the whole sentence.

First approach is using template augmentation which is using pre-defined question can using rule-based to generate an answer to pair with the template questions. Second approach leverages LSTM to generate a question by providing image feature.



Performance result with and without text augmentation (Kafle et al. 2017)

**Recommendation**

The above approach is designed to solve problems which authors are facing in their problem. If you understand your data, you should tailor made augmentation approach it. Remember that golden rule in data science is garbage in garbage out.

In general, you can try thesaurus approach without quite understanding of your data. It may not boost up a lot due to the aforementioned thesaurus approach limitation.

**About Me**

I am Data Scientist in Bay Area. Focusing on state-of-the-art in Data Science, Artificial Intelligence , especially in NLP and platform related. Feel free to connect with [me](https://makcedward.github.io/)on [LinkedIn](https://www.linkedin.com/in/edwardma1026) or following me on [Medium](http://medium.com/@makcedward/) or [Github](https://github.com/makcedward).

**Extension Reading**

* Image augmentation library ([imgaug](https://github.com/aleju/imgaug))
* Text augmentation library([nlpaug](https://towardsdatascience.com/data-augmentation-library-for-text-9661736b13ff))
* [Data Augmentation for Text](https://towardsdatascience.com/data-augmentation-library-for-text-9661736b13ff)
* [Data Augmentation for Audio](https://towardsdatascience.com/data-augmentation-for-audio-76912b01fdf6)
* [Data Augmentation for Spectrogram](https://towardsdatascience.com/data-augmentation-for-speech-recognition-e7c607482e78)

**Reference**

* X. Zhang, J. Zhao and Y. LeCun. [Character-level Convolutional Networks for Text Classification](https://arxiv.org/pdf/1509.01626.pdf). 2015
* W. Y. Wang and 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](https://aclweb.org/anthology/D15-1306). 2015
* R. Sennrich, B. Haddow and A Birch. [Improving Neural Machine Translation Models with Monolingual Data](https://arxiv.org/pdf/1511.06709.pdf). 2016
* M. Fadaee, A. Bisazza, C. Monz. [Data Augmentation for Low-Resource Neural Machine Translation](https://arxiv.org/pdf/1705.00440.pdf). 2017
* S. Kobayashi. [Contextual Augmentation: Data Augmentation by Words with Paradigmatic Relation](https://arxiv.org/pdf/1805.06201.pdf). 2018
* K. Kafle, M. Yousefhussien and C. Kanan. [Data Augmentation for Visual Question Answering](https://aclweb.org/anthology/W17-3529). 2017
* C. Coulombe. [Text Data Augmentation Made Simple By Leveraging NLP Cloud APIs](https://arxiv.org/ftp/arxiv/papers/1812/1812.04718.pdf). 2018