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. 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:

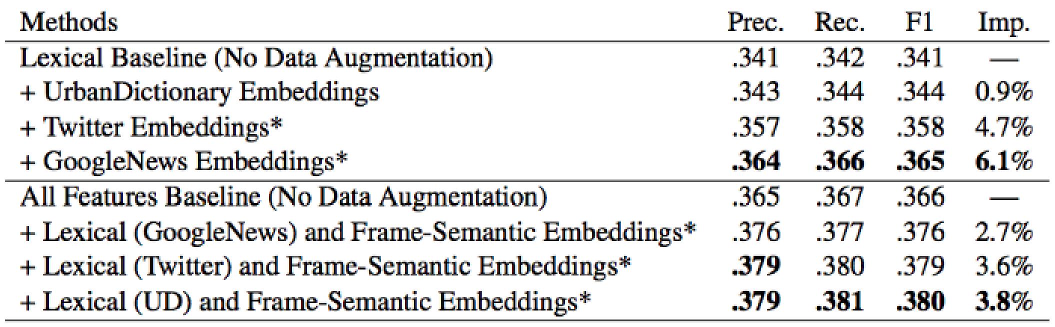
- Thesaurus
- Word Embeddings
- Back Translation
- Contextualized Word Embeddings
- Text Generation

Thesaurus

Zhang et al. introduced synonyms Character-level Convolutional Networks for Text Classification. 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

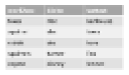
Wang and Yang introduced word similar calculation in That's So Annoying!!!: A Lexical and Frame-Semantic Embedding Based Data Augmentation Approach to Automatic Categorization of Annoying Behaviors using #petpeeve Tweets. In the paper, Wang and Yang proposed to use k-nearest-neighbor (KNN) and cosine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

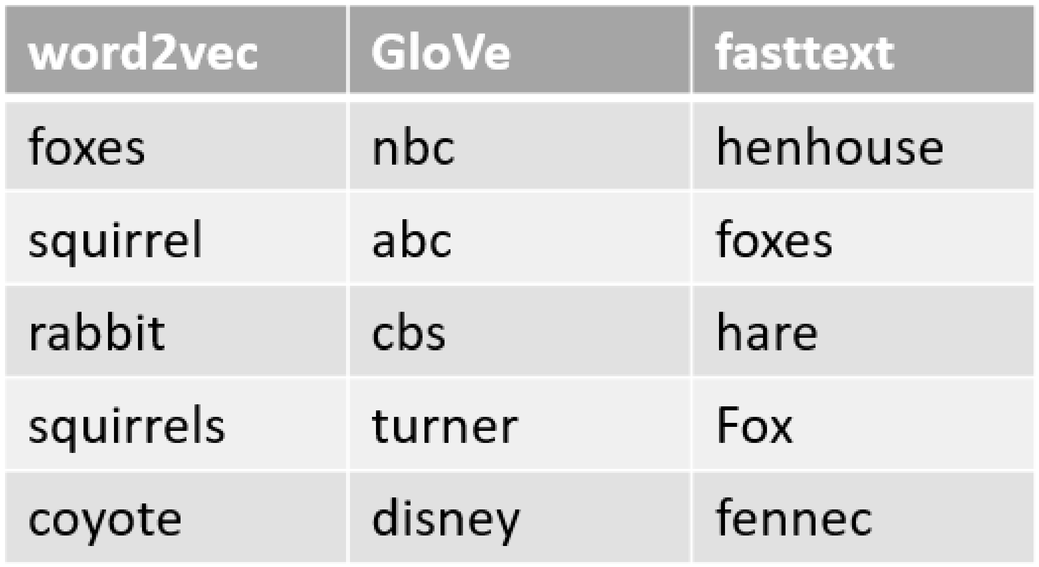


Methods	Prec.	Rec.	F1	Imp.
Lexical Baseline (No Data Augmentation)	.341	.342	.341	—
+ UrbanDictionary Embeddings	.343	.344	.344	0.9%
+ Twitter Embeddings*	.357	.358	.358	4.7%
+ GoogleNews Embeddings*	**.364**	**.366**	**.365**	**6.1%**
All Features Baseline (No Data Augmentation)	.365	.367	.366	—
+ Lexical (GoogleNews) and Frame-Semantic Embeddings*	.376	.377	.376	2.7%
+ Lexical (Twitter) and Frame-Semantic Embeddings*	**.379**	.380	.379	3.6%
+ Lexical (UD) and Frame-Semantic Embeddings*	**.379**	**.381**	**.380**	**3.8%**

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

Alternatively, we can leverages pre-trained classic word embeddings such as word2vec, GloVe and fasttext to perform similarity search.





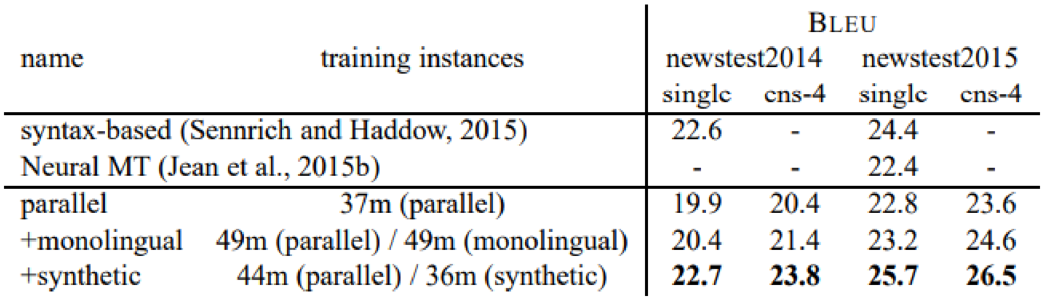
word2vec	GloVe	fasttext
foxes	nbc	henhouse
squirrel	abc	foxes
rabbit	cbs	hare
squirrels	turner	Fox
coyote	disney	fennec

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.



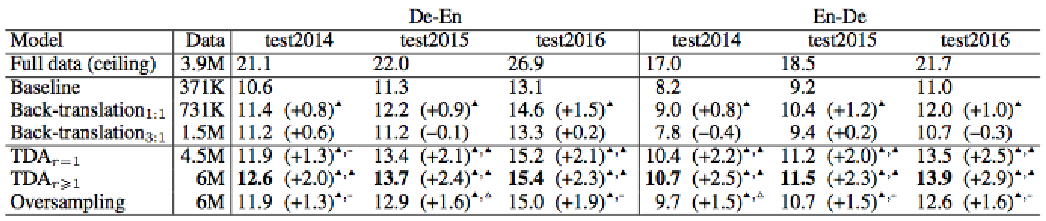
name	training instances	BLEU newstest2014 single	BLEU newstest2014 ens-4	BLEU newstest2015 single	BLEU newstest2015 ens-4
syntax-based (Sennrich and Haddow, 2015)		22.6	-	24.4	-
Neural MT (Jean et al., 2015b)		-	-	22.4	-
parallel	37m (parallel)	19.9	20.4	22.8	23.6
+monolingual	49m (parallel) / 49m (monolingual)	20.4	21.4	23.2	24.6
+synthetic	44m (parallel) / 36m (synthetic)	**22.7**	**23.8**	**25.7**	**26.5**

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 .

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.

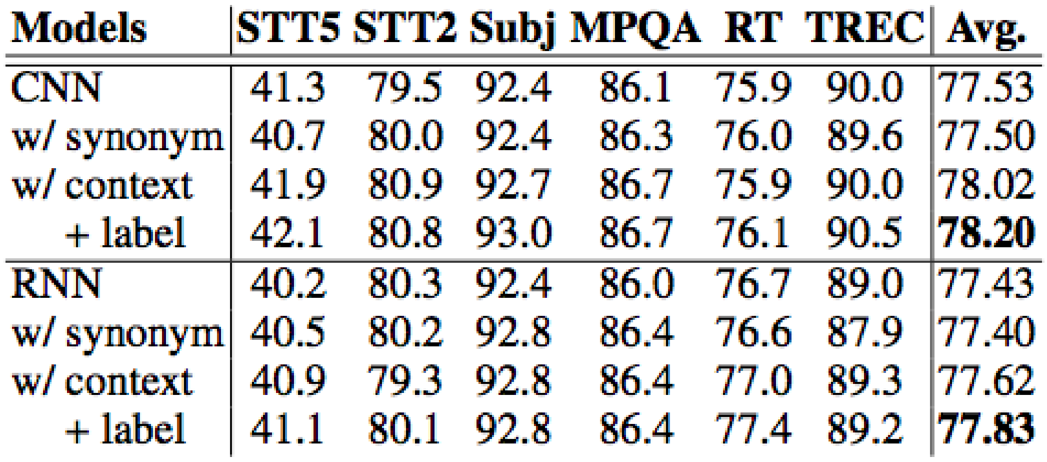


		De-En			En-De		
Model	Data	test2014	test2015	test2016	test2014	test2015	test2016
Full data (ceiling)	3.9M	21.1	22.0	26.9	17.0	18.5	21.7
Baseline	371K	10.6	11.3	13.1	8.2	9.2	11.0
Back-translation$_{1:1}$	731K	11.4 (+0.8)$^{\blacktriangle}$	12.2 (+0.9)$^{\blacktriangle}$	14.6 (+1.5)$^{\blacktriangle}$	9.0 (+0.8)$^{\blacktriangle}$	10.4 (+1.2)$^{\blacktriangle}$	12.0 (+1.0)$^{\blacktriangle}$
Back-translation$_{3:1}$	1.5M	11.2 (+0.6)	11.2 (–0.1)	13.3 (+0.2)	7.8 (–0.4)	9.4 (+0.2)	10.7 (–0.3)
TDA$_{r=1}$	4.5M	11.9 (+1.3)$^{\blacktriangle,-}$	13.4 (+2.1)$^{\blacktriangle,\blacktriangle}$	15.2 (+2.1)$^{\blacktriangle,\blacktriangle}$	10.4 (+2.2)$^{\blacktriangle,\blacktriangle}$	11.2 (+2.0)$^{\blacktriangle,\blacktriangle}$	13.5 (+2.5)$^{\blacktriangle,\blacktriangle}$
TDA$_{r\geqslant1}$	6M	**12.6** (+2.0)$^{\blacktriangle,\blacktriangle}$	**13.7** (+2.4)$^{\blacktriangle,\blacktriangle}$	**15.4** (+2.3)$^{\blacktriangle,\blacktriangle}$	**10.7** (+2.5)$^{\blacktriangle,\blacktriangle}$	**11.5** (+2.3)$^{\blacktriangle,\blacktriangle}$	**13.9** (+2.9)$^{\blacktriangle,\blacktriangle}$
Oversampling	6M	11.9 (+1.3)$^{\blacktriangle,-}$	12.9 (+1.6)$^{\blacktriangle,\vartriangle}$	15.0 (+1.9)$^{\blacktriangle,-}$	9.7 (+1.5)$^{\blacktriangle,\vartriangle}$	10.7 (+1.5)$^{\blacktriangle,-}$	12.6 (+1.6)$^{\blacktriangle,-}$

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. 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.



Models	STT5	STT2	Subj	MPQA	RT	TREC	Avg.
CNN	41.3	79.5	92.4	86.1	75.9	90.0	77.53
w/ synonym	40.7	80.0	92.4	86.3	76.0	89.6	77.50
w/ context	41.9	80.9	92.7	86.7	75.9	90.0	78.02
+ label	42.1	80.8	93.0	86.7	76.1	90.5	**78.20**
RNN	40.2	80.3	92.4	86.0	76.7	89.0	77.43
w/ synonym	40.5	80.2	92.8	86.4	76.6	87.9	77.40
w/ context	40.9	79.3	92.8	86.4	77.0	89.3	77.62
+ label	41.1	80.1	92.8	86.4	77.4	89.2	**77.83**

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. 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.

[illegible]

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 on LinkedIn or following me on Medium or Github.

Extension Reading

Reference