A Survey of Data Augmentation Approaches for NLP (Steven Y. Feng, Varun Gangal, Jason Wei, et al.)

Plichta Weronika, Taczała Michał

Warsaw University of Technology

January 15, 2025

Data Augmentation that we know of

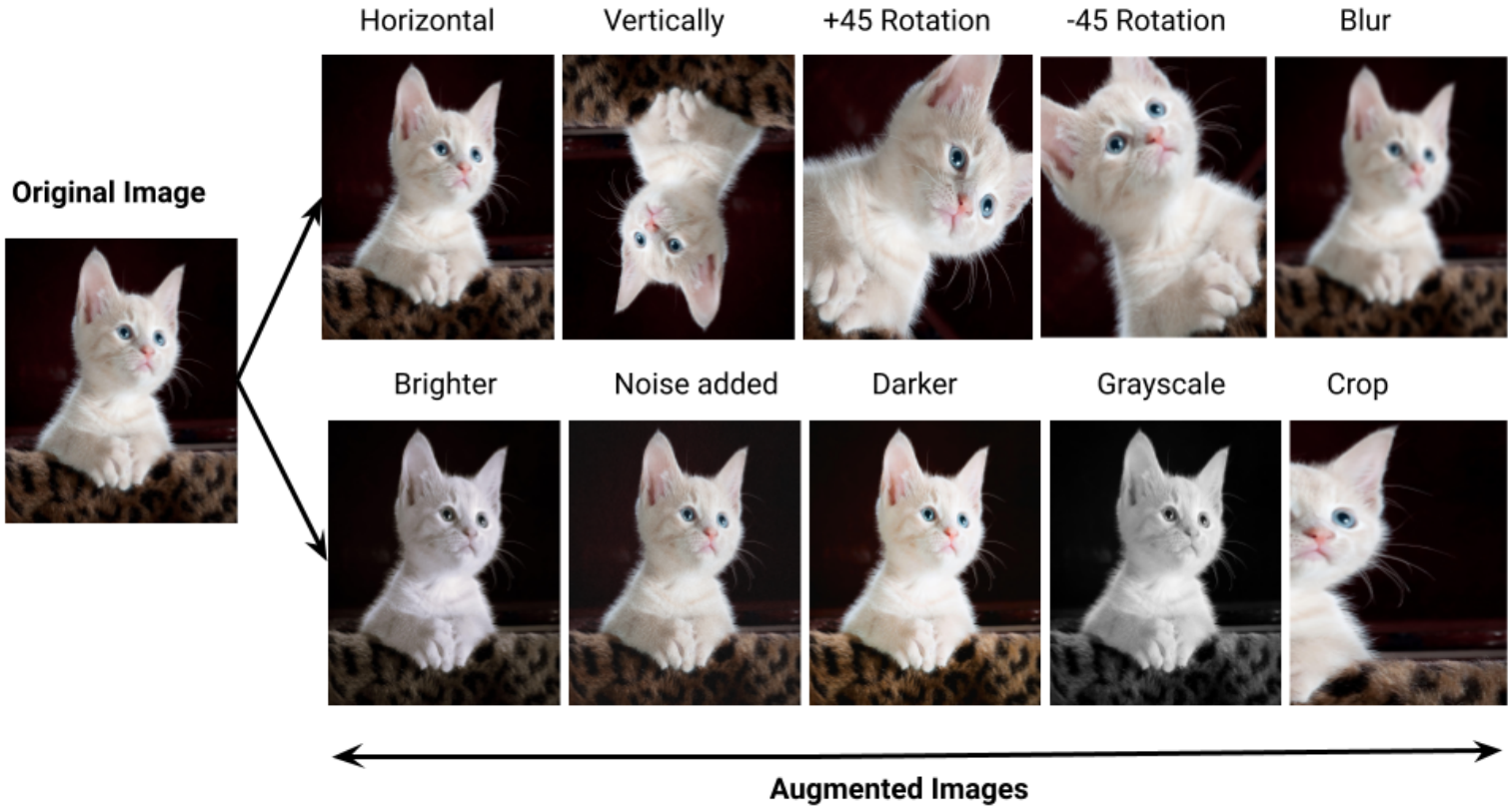


Figure: Image augmentation example

Why do we need DA in NLP?

- Limited training data
- Overfitting
- Imbalance of the classes
- Different languages and areas of interest

Introduction to Data Augmentation

- DA increases training data diversity without explicit collection of data
- Techniques borrowed from Computer Vision(Random cropping/Text truncation, Adding noise)
- Key applications: low-resource domains, bias mitigation, and few-shot learning

Trade-offs? Why can't we have everything that we want...

- The more complex the augmentation, the better the result(very often that's the case)
- Diverse data prevent overfitting. Diverse data also might not maintain similarity and provide some incorrect information
- Some augmentation might not be very useful for big pre-train models but might be helpful for smaller models with more specific area of interest

Challenges in NLP Data Augmentation

- Language is discrete
- Balancing data diversity and distributional similarity
- Lack of theoretical understanding of DA effectiveness(LIMITED RESEARCH!)
- Task-specific augmentations

Rule-based DA Techniques

- **Easy Data Augmentation (EDA)**: Token-level operations(f.e. synonyms)
- Original: The cat sat on the mat
- Synonym Replacement: The feline sat on the mat
- **Dependency Tree Morphing**: Structural transformations
- Original: The cat sat on the mat.
- Augmented: On the mat, the cat sat.
- **Back translation**: Translating to another language and back
- Original: The cat sat on the mat.
- Translated: The cat was sitting on the carpet.
- **Random Noise Injection**: Deleting single letters/changing letter order etc.
- Original: The cat sat on the mat.
- Augmented: The ct sat on teh mat.

Interpolation-based DA Techniques

- **Embedding-level MIXUP**: Interpolates inputs and labels
- Blends the embedding of both sentences
- $s_{\text{mix}} = \lambda s_1 + (1 - \lambda) s_2$
- **SEQ2MIXUP**: Extends MIXUP for sequence tasks
- Sentence 1: The cat sat on the mat.
- Sentence 2: The dog barked loudly.
- Augmented: The cat barked loudly on the mat.
- **SWITCHOUT**: Switch random word with one from the vocabulary set
- Original: The cat sat on the mat
- Augmented: The dog sat on the chair

Model-based DA Techniques

- Backtranslation: Generates paraphrases via translation with the use of a model.
- Contextual Augmentation: Uses pre-trained models to modify text.
- Label-conditioned generation: Generates data based on class labels.

Applications: Low-Resource Languages

- Using high-resource languages with similar properties(pretrain on high-resource, fine-tune on low-resource)
- Back translation
- Rare word augmentation

Applications: Mitigating Bias

- Gender bias mitigation via counterfactual data augmentation, F.e. Gender swapping entities
- Oversampling
- Cultural and Ethnic Bias Mitigation(change names to match different countries, genders etc)

Applications: Fixing Class Imbalance

- Oversampling minority classes with Synthetic Minority Oversampling (SMOTE).
- Few-shot learning benefits from analogy-based augmentations.
- Techniques: Random sampling, SMOTE, Easy Data Analysis.
- Most of the standard techniques aren't very useful - they don't provide meaningful information

Applications: Few-Shot Learning

- Approach where a model learns to understand new tasks or classify new data points based on just a few example
- It contains support set small set of labeled examples, which contains input-output pairs
- New, unseen examples to be classified
- Model applies learned patterns
- Example: Curriculum learning with augmented data.
- $+$ Improved generalization with limited data.

Applications: Summarization and Question Answering

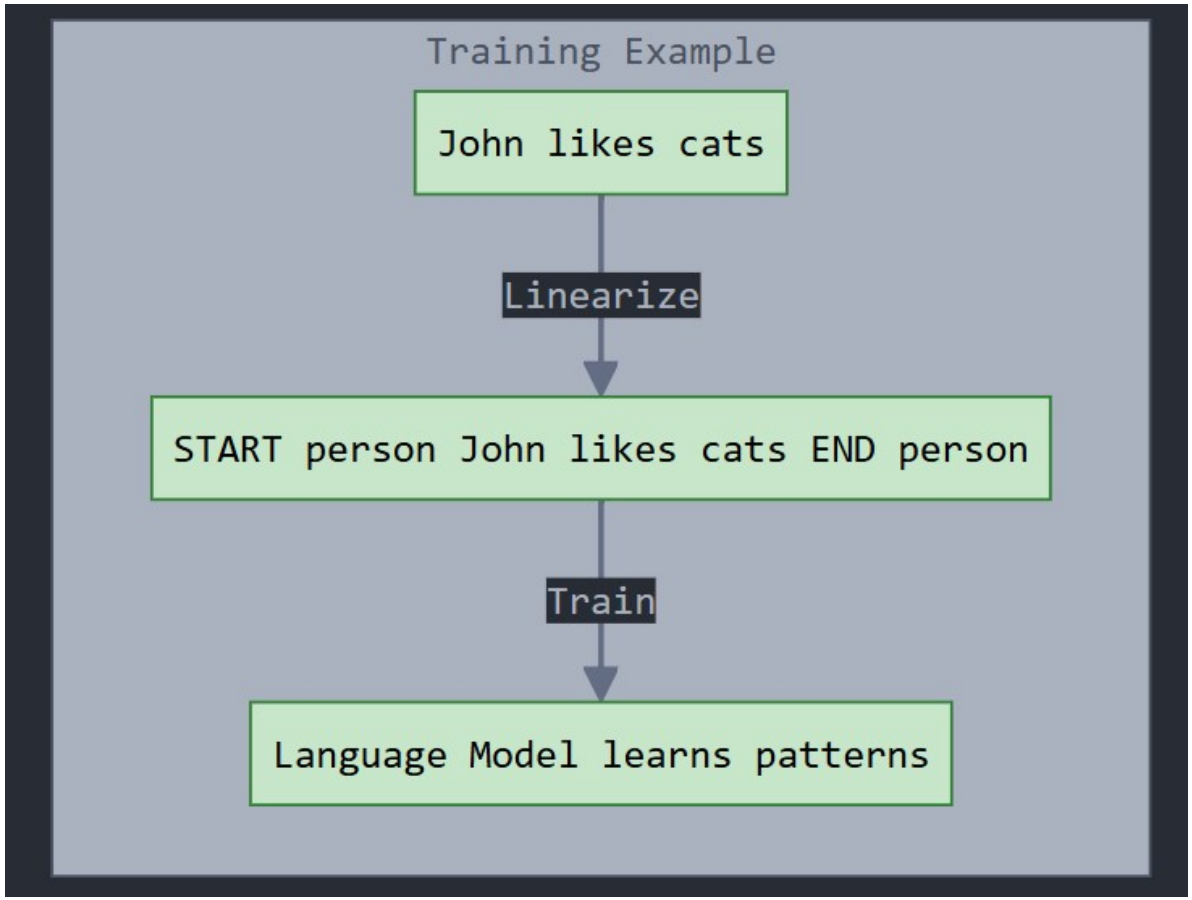
- Techniques: Backtranslation, synthetic and real data mixing.
- English $\rightarrow$ Polish, Polish $\rightarrow$ English. Translated text can give us new variants of text.
- Use case: Low-resource summarization tasks.
- Cross-lingual augmentation for multilingual QA.

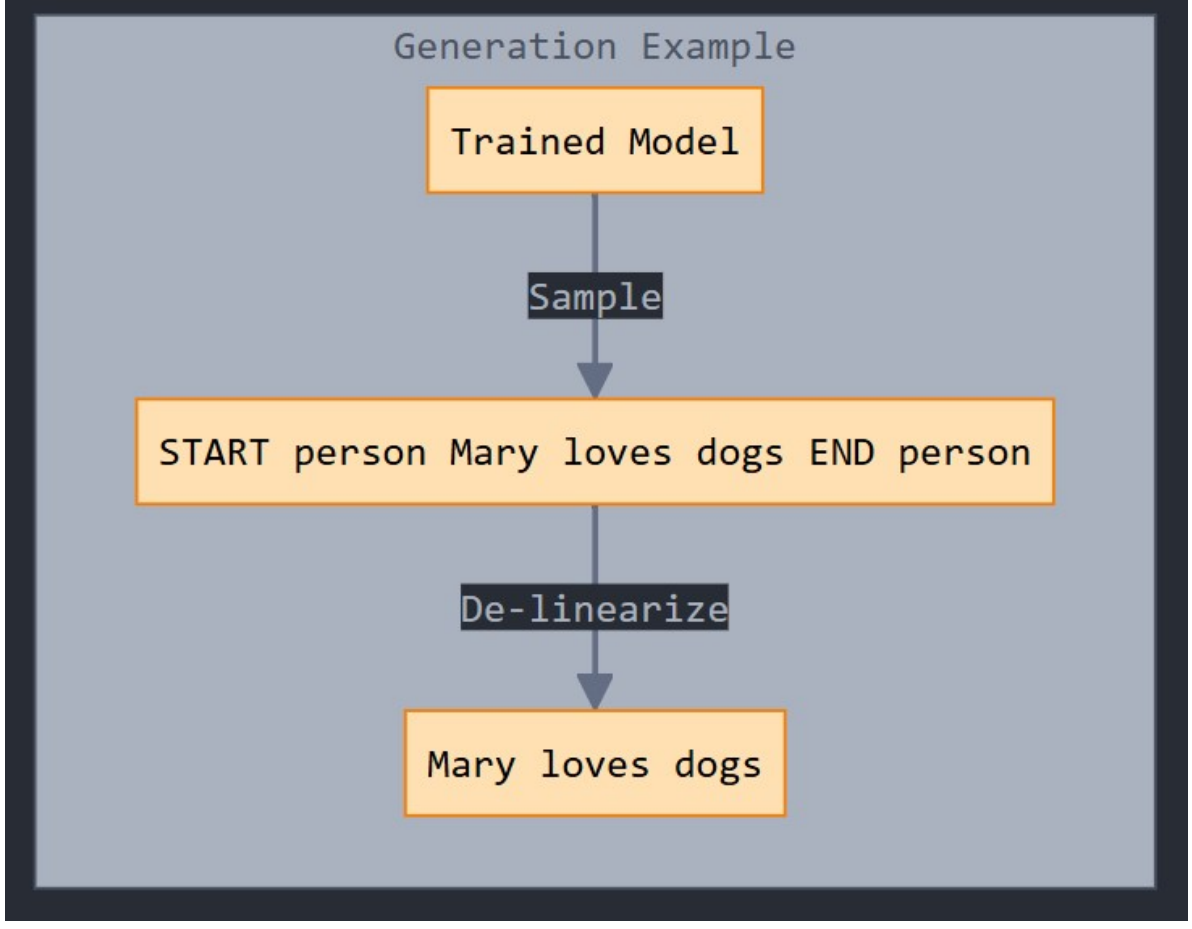
Applications: Sequence Tagging

- It aims to classify each token (word) in a class space C. This classification approach can be independent (each word is treated independently) or dependent (each word is dependent on other words).
- Technique: DAGA approach

DAGA approach

1. A language model over sequences of tags and words linearized as per a certain scheme is learned.
2. Sequences are sampled from this language model and de-linearized to generate new examples.




Generation Example
Trained Model
Sample
START person Mary loves dogs END person
De-linearize
Mary loves dogs

Parsing task

- DATA RECOMBINATION for injecting task-specific priors to neural semantic parsers. A synchronous context-free grammar (SCFG) is induced from training data, and new "recombinant" examples are sampled. (Jia and Liang (2016))
- Compositionality to construct synthetic examples for downstream tasks like semantic parsing. Fragments of original examples are replaced with fragments from other examples in similar contexts. (Andreas 2020)

Grammatical Error Correction (GEC)

- Using revision history of Wikipedia German edits and use those relating to GEC as augmented training data. (Boyd 2018)
- Adding synthetic errors to noise the text Wang et al. (2019a) There were two apporaches: 1) token-level perturbations and 2) training error generation models with a filtering strategy to keep generations with sufficient errors.

Applications: Neural Machine Translation

- Techniques: Backtranslation, SWITCHOUT - randomly replaces words in both source and target sentences with other random words from their corresponding vocabularies
- Data diversification with multiple forward and backward models.

Applications: Data-to-Text Generation

- refers to tasks which require generating natural language descriptions of structured or semi-structured data inputs, e.g. game score tables
- Augmentation techniques: WebNLG (Web Natural Language Generation), E2E-NLG (End-to-End Natural Language Generation)

WebNLG

Purpose

Convert RDF triples into natural language text

Input Format: (Alan Bean, occupation, Test pilot), (Alan Bean, birthDate, 1932-03-15)

Output:

"Alan Bean was a test pilot who was born on March 15, 1932."

Key Features:

- DBpedia-based data
- Multiple domains
- Tests:
  - Information aggregation
  - Entity relationships
  - Grammar accuracy
  - Factual correctness

Applications: Dialogue Systems

1. Slot Substitution (SLOT-SUB)

- Replaces slot values while preserving semantics
- Uses existing training data values
- Example: "cheapest" $\rightarrow$ "least expensive" (COST_RELATIVE)
- Most effective method in experiments

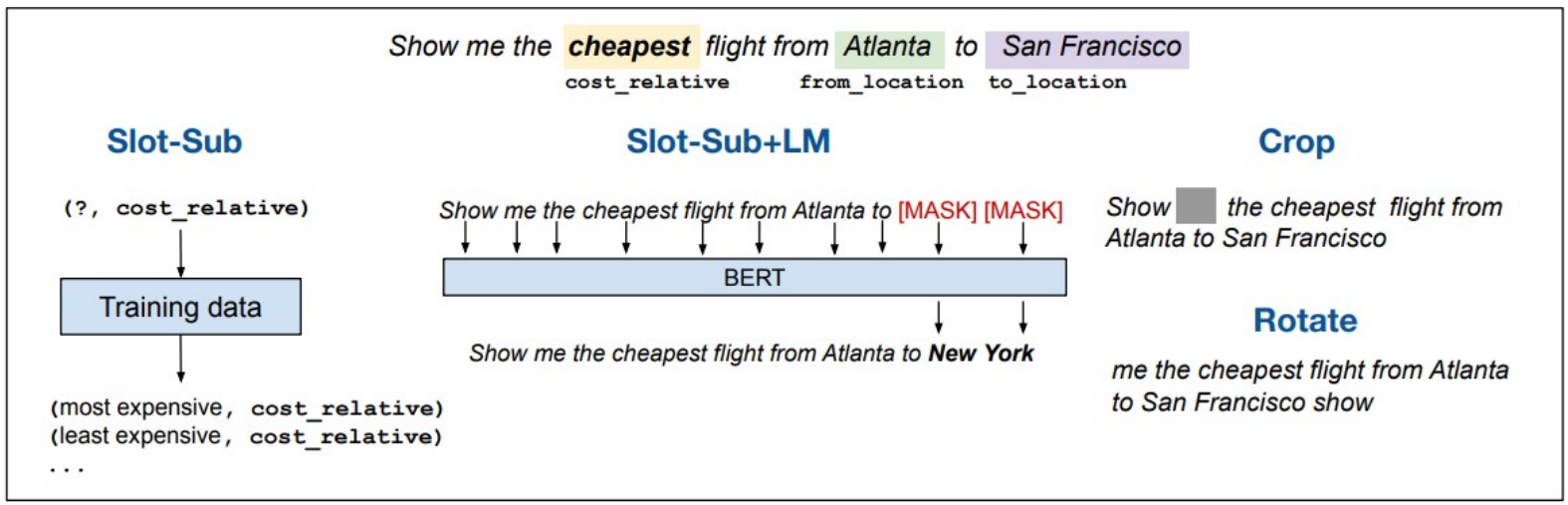
2. Slot Substitution with LM (SLOT-SUB-LM)

- Uses BERT to generate substitutions
- Includes semantic compatibility filter
- Example: Masks slots with [MASK] tokens

3. Structural Operations

- CROP: Removes sentence fragments
- ROTATE: Changes sentence structure
- Based on dependency parsing
- Example: “show me the flight” $\rightarrow$ “me the flight show”

Simple is Better! Lightweight Data Augmentation for Low Resource Slot Filling and Intent Classification



Challenges and Future Directions

- **Theoretical gaps in understanding DA effectiveness**. - Most studies might show empirically that a DA technique works and provide some intuition, but it is currently challenging to measure the goodness of a technique without resorting to a full-scale experiment.
- DA - Rather than being universally applied, it should be used strategically only when dealing with limited or unique data scenarios that aren't well-represented in pre-training.

Pedro Domingos
@pmddomingos

Data augmentation is one of the ugliest hacks in ML. If you know what the invariances are, encode them into the architecture. Don't blow up the size of you dataset in order to approximate them.

4:01 PM · May 10, 2021 · Twitter Web App

33 Retweets **9** Quote Tweets **362** Likes

Challenges

- Working in specialized domains and low resourceful languages such as those with domain-specific vocabulary and jargon (e.g. medicine) can present challenges. Many pretrained models and external knowledge (e.g. WordNet) cannot be effectively used. Studies have shown that DA becomes less beneficial when applied to out-of-domain data, likely because the **distribution of augmented data** can substantially differ from the original data
- A common practice for DA in NLP is to generate augmented data offline and store it as additional data to be loaded during training. Future work on **a lightweight module for online DA** in NLP could be fruitful

Summary of Findings

- DA enhances NLP models in low-resource and specialized tasks.
- Techniques vary from simple rule-based to complex model-based methods.
- Significant room for research in theoretical and practical aspects of DA.

References and Acknowledgments

- Full paper: arXiv:2105.03075
- GitHub repository: DataAug4NLP
- Authors: Steven Y. Feng, Varun Gangal, Jason Wei, et al.
- Samuel Louvan and Bernardo Magnini. 2020. Simple is better! lightweight data augmentation for low resource slot filling and intent classification
- Sebastien Montella, Betty Fabre, Tanguy Urvoy, Johannes Heinecke, and Lina Rojas-Barahona. 2020. Denoising pre-training and data augmentation strategies for enhanced RDF verbalization with transformers