The Journal of Open Source Software

Augmenty: A Python Library for Structured Text Augmentation

Kenneth Enevoldsen [1]

1 Center for Humanities Computing, Aarhus University, Aarhus, Denmark

DOI: 10.21105/joss.06370

Software

- Review
- Repository
- Archive

Editor: Arfon Smith

Reviewers:

- @sap218
- @wdduncan

Submitted: 09 December 2023
Published: 27 April 2024

Summary

Text augmentation - the process of generating new text samples by applying transformations to existing samples - is a useful tool for training (Wei & Zou, 2019) and evaluating (Ribeiro et al., 2020) natural language processing (NLP) models and systems. Despite its utility, existing libraries are often limited in terms of functionality and flexibility. They are confined to basic tasks such as text classification or catering to specific downstream use cases such as estimating robustness (Goel et al., 2021). Recognizing these constraints, `Augmenty` is a tool for structured augmentation of text along with its annotations. `Augmenty` integrates seamlessly with the popular NLP library `spaCy` (Honnibal et al., 2020) and seeks to be compatible with all models and tasks supported by `spaCy`. Augmenty provides a wide range of augmenters that can be combined flexibly to create complex augmentation pipelines. It also includes a set of primitives that can be used to create custom augmenters such as word replacement augmenters. This functionality allows for augmentations within a range of applications such as named entity recognition (NER), part-of-speech tagging, and dependency parsing.

Statement of need

Augmentation has proven to be a powerful tool within disciplines such as computer vision (Wang et al., 2017) and speech recognition (Park et al., 2019) where it is used for both training more robust models and for evaluating the ability of the models to handle perturbations. Within NLP augmentation has seen some uses as a tool for generating additional training data (Wei & Zou, 2019), but has shined as a tool for model evaluation, such as estimating robustness (Goel et al., 2021) and bias (Lassen et al., 2023), or for creating novel datasets (Nielsen, 2023).

Despite its utility, existing libraries for text augmentation often exhibit limitations in terms of functionality and flexibility. Commonly they only provide pure string augmentation which typically leads to the annotations becoming misaligned with the text. This has limited the use of augmentation to tasks such as text classification while neglecting structured prediction tasks such as named entity recognition (NER) or coreference resolution. This has limited the use of augmentation to a wide range of tasks for training and evaluation.

Existing tools such as `textgenie` (Pandya, 2023), and `textaugment` (Marivate & Sefara, 2020) implement powerful techniques such as backtranslation and paraprashing, which are useful augmentations for text-classification tasks. However, these tools neglect a category of tasks that require that the annotations be aligned with the augmentation of the text. For instance, even simple augmentations such as replacing the named entity "Jane Doe" with "John" will lead to a misalignment of the NER annotation, part-of-speech tags, etc., which if not properly handled will lead to a misinterpretation of the model performance or generation of incorrect training samples.

Other tools for data augmentation focus on specific downstream applications such as textattack (Morris et al., 2020) which is useful for adversarial attacks of classification systems or robustnessgym (Goel et al., 2021) which is useful for evaluating the robustness of classification systems.

Augmenty introduces a flexible and easy-to-use interface for structured text augmentation, seeking to augment the annotations along with the text. Augmenty is built to integrate well with spaCy (Honnibal et al., 2020) and seeks to be compatible with the broad set of tasks supported by spaCy. Augmenty provides augmenters that take a spaCy Doc-object (but works just as well with string-objects) and return a new Doc-object with the augmentations applied. This allows for augmentations of both the text and the annotations present in the Doc-object. Augmenty does not seek to replace tools such as textattack, but seeks to provide a general-purpose tool for augmentation of both the text and its annotations. This allows for augmentations within a range of applications such as named entity recognition, part-of-speech tagging, and dependency parsing.

Features & Functionality

Augmenty is a Python library that implements augmentations based on spaCy's Doc object. spaCy's Doc object is a container for a text and its annotations. This makes it easy to augment text and annotations simultaneously. The Doc object can easily be extended to include custom augmentation not available in spaCy by adding custom attributes to the Doc object. While Augmenty is built to augment Docs the object is easily converted into strings, lists or other formats. The annotations within a Doc can be provided either by human annotations or using a trained model.

Augmenty implements a series of augmenters for token-, span- and sentence-level augmentation. These augmenters range from primitive augmentations such as word replacement to language-specific augmenters such as keystroke error augmentations based on a French keyboard layout. Augmenty also integrates with other libraries such as NLTK (Bird et al., 2009) to allow for augmentations based on WordNet (Miller, 1994) and allows for specification of static word vectors (Pennington et al., 2014) to allow for augmentations based on word similarity. Lastly, augmenty provides a set of utility functions for repeating augmentations, combining augmenters, or adjusting the percentage of documents that should be augmented. This allows for the flexible construction of augmentation pipelines specific to the task at hand.

Example Use Cases

Augmenty has already been used in a number of projects. The code base was initially developed for evaluating the robustness and bias of DaCy (Enevoldsen et al., 2021), a state-of-the-art Danish NLP pipeline. It is also continually used to evaluate Danish NER systems for biases and robustness on the DaCy website. Augmenty has also been used to detect intersectional biases (Lassen et al., 2023) and used within benchmarks of Danish language models (Sloth & Rybner, 2023).

Besides its existing use cases Augmenty could for example also be used to a) upsample minority classes without duplicating samples, b) train less biased models by e.g. replacing names with names of minority groups c) train more robust models e.g. by augmenting with typos or d) generate pseudo-historical data by augmenting with known spelling variations of words.

Target Audience

The package mainly targets NLP researchers and practitioners who wish to augment their data for training or evaluation. The package is also targeted at researchers who wish to evaluate their models with augmentations or want to generate new datasets.

Acknowledgements

The authors thank the contributors of the package notably Lasse Hansen which provided meaningful feedback on the design of the package at early stages of development.

References

Bird, S., Loper, E., & Klein, E. (2009). *Natural language processing with python. O'reilly media inc., sebastopol, USA*.

Enevoldsen, K., Hansen, L., & Nielbo, K. L. (2021). *DaCy: A Unified Framework for Danish NLP*. https://ceur-ws.org/Vol-2989/short_paper24.pdf

Goel, K., Rajani, N. F., Vig, J., Taschdjian, Z., Bansal, M., & Ré, C. (2021). Robustness gym: Unifying the NLP evaluation landscape. In A. Sil & X. V. Lin (Eds.), *Proceedings of the 2021 conference of the north american chapter of the association for computational linguistics: Human language technologies: demonstrations* (pp. 42–55). Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.naacl-demos.6

Honnibal, M., Montani, I., Van Landeghem, S., & Boyd, A. (2020). *spaCy: Industrial-strength natural language processing in python*. https://doi.org/10.5281/zenodo.1212303

Lassen, I. M. S., Almasi, M., Enevoldsen, K., & Kristensen-McLachlan, R. D. (2023). Detecting intersectionality in NER models: A data-driven approach. In S. Degaetano-Ortlieb, A. Kazantseva, N. Reiter, & S. Szpakowicz (Eds.), *Proceedings of the 7th joint SIGHUM workshop on computational linguistics for cultural heritage, social sciences, humanities and literature* (pp. 116–127). Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.latechclfl-1.13

Marivate, V., & Sefara, T. (2020). Improving short text classification through global augmentation methods. *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*, 385–399.

Miller, G. A. (1994). WordNet: A lexical database for English. *Human Language Technology: Proceedings of a Workshop Held at Plainsboro, New Jersey, March 8-11, 1994*. https://aclanthology.org/H94-1111

Morris, J., Lifland, E., Yoo, J. Y., Grigsby, J., Jin, D., & Qi, Y. (2020). TextAttack: A framework for adversarial attacks, data augmentation, and adversarial training in NLP. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 119–126.

Nielsen, D. (2023). ScandEval: A benchmark for Scandinavian natural language processing. In T. Alumäe & M. Fishel (Eds.), *Proceedings of the 24th nordic conference on computational linguistics (NoDaLiDa)* (pp. 185–201). University of Tartu Library. https://aclanthology.org/2023.nodalida-1.20

Pandya, H. (2023). *Hetpandya/textgenie*. https://github.com/hetpandya/textgenie

Park, D. S., Chan, W., Zhang, Y., Chiu, C.-C., Zoph, B., Cubuk, E. D., & Le, Q. V. (2019). SpecAugment: A simple data augmentation method for automatic speech recognition. *Interspeech*. https://api.semanticscholar.org/CorpusID:121321299

Pennington, J., Socher, R., & Manning, C. (2014). GloVe global vectors for word representation. In A. Moschitti, B. Pang, & W. Daelemans (Eds.), *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532–1543). Association for Computational Linguistics. https://doi.org/10.3115/v1/D14-1162

Ribeiro, M. T., Wu, T., Guestrin, C., & Singh, S. (2020). Beyond accuracy: Behavioral testing of NLP models with CheckList. In D. Jurafsky, J. Chai, N. Schluter, & J. Tetreault (Eds.), *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 4902–4912). Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.442

Sloth, T. R., & Rybner, A. S. (2023). *DaDebias/genda-lens*. DaDebias. https://github.com/DaDebias/genda-lens

Wang, J., Perez, L., & others. (2017). The effectiveness of data augmentation in image classification using deep learning. *Convolutional Neural Networks Vis. Recognit*, *11*(2017), 1–8.

Wei, J., & Zou, K. (2019). EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In K. Inui, J. Jiang, V. Ng, & X. Wan (Eds.), *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)* (pp. 6382–6388). Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1670