Modeling the Graphotactics of Low-Resource Languages Using Sequential GANs

Isaac Wasserman
Department of Computer Science
Haverford College
Haverford, PA 19041
iwasserman@haverford.edu

Abstract

Generative Adversarial Networks (GANs) have been shown to aid in the creation of artificial data in situations where large amounts of real data are difficult to come by. This issue is especially salient in the computational linguistics space, where researchers are often tasked with modeling the complex morphologic and grammatical processes of low-resource languages. This paper will discuss the implementation and testing of a GAN that attempts to model and reproduce the graphotactics of a language using only 100 example strings. These artificial, yet graphotactically compliant, strings are meant to aid in modeling the morphological inflection of low-resource languages.

1 Introduction

1.1 Task

In 2019, Anastasopoulos and Neubig made waves with their multilingual morphological inflection model for low resource languages [1] that they submitted to the SIGMORPHON 2019 shared task [10]. All models submitted were pretrained on high resource languages of similar ancestry to the target language, allowing many models to greatly exceed the performance of previous attempts at low-resource morphological inflection. However, what allowed Anastasopoulos and Neubig's model to outperform other submissions was its use of data "hallucination".

To perform this hallucination, they aligned the lemma with its inflected form, extracted the stem, and generated new artificial examples by replacing this stem with randomly generated strings (in the language's alphabet) of equal length.[1] Though a seemingly haphazard method of data augmentation, this approach allowed for an additional 10% accuracy, on average, when tested against versions of the model that only used cross-lingual transfer.

Surely, a more well informed approach to stem generation would further improve the accuracy of the inflectional model. Given the demonstrated ability of GANs to produce photorealistic, yet completely contrived images, they are potentially ideal for such a task. The experiments detailed in this paper attempt to produce a technique for generating fake word stems that provide more relevant information to the inflectional model from Anastasopoulos and Neubig, 2019 [1], thereby increasing the accuracy of its inflections. By modeling the graphotactics of the target language using a GAN, it should be possible to produce strings that more accurately depict possible character sequences.

[1] The alignment process assumes that the lemma and inflected form share a common substring.

1.2 Generative Adversarial Networks

Generative adversarial networks are a class of unsupervised machine learning architectures, most commonly used for image generation. These networks consist of a generator and a discriminator that are trained simultaneously on a set of data representing a class or domain; this domain could be anything from photos of human faces to time series of hourly temperatures.[2] The generator is tasked with producing "fake" examples that are within this domain without ever seeing any real examples from the training set. Meanwhile, the discriminator is fed a combination of fake examples (from the generator) and real examples and is tasked with classifying them as real or fake. The respective goals of the generator and discriminator constitute a zero-sum game, in which the generator is constantly trying to outsmart the discriminator, while the discriminator hones its ability to distinguish between in-domain and out-of-domain examples.

Though GANs are most often applied to image data (as in the popular CycleGAN [15], StyleGAN [8], and DiscoGAN [9]), the same logic is also applicable to other types of data. For example, in 2017, Esteban et al. developed a pair of recurrent GANs which they applied to medical time series data [4], and in 2016, Yu et al. developed a GAN architecture made specifically for sequences and language generation, utilizing techniques from reinforcement learning [14].

1.3 GANs for Data Augmentation

Although GANs are most popularly used for domain transfer, the same basic architecture is also a good candidate for a data augmentation strategy called hallucination in which new (fake) training examples based on a small number of existing (real) examples.

Though theoretically, the addition of these fake examples should not improve the performance of a discriminative model (since they can be no more representative of the true distribution than the examples they are based on), an empirical study by Antoniou et al. observed accuracy improvements on multi-class classification of up to 13% on benchmark low-resource datasets such as Omniglot [2]. These performance gains are supported by a fairly extensive body of similar evidence-based studies on GAN-based data augmentation in low-resource settings. These studies are, more often than not, concerned with medical imaging, in which segmentation is a more salient issue than classification.

Shin et al., 2018 [12] leveraged the popular Pix2Pix [7] architecture to augment a brain-tumor segmentation dataset. This task was considerably more complex as it involved the hallucination of image pairs. However, training their segmentation model on a combination of real and fake data, they observed performance improvements of up to 16% over unaugmented data (minimum improvement of 1%). They also tested the effects of an entirely synthetic dataset; however, this resulted in a significant loss of performance compared to the baseline. Despite the improvements realized when compared to their baseline segmentor, even their best model was unable to outperform the best-in-class reference model [13].

Sandfort et al., 2019 [11] carried out a similar study using the also popular CycleGAN architecture [15] to segment anomolous CT scans of kidneys, livers, and spleens. On average, this study showed no appreciable difference between the performance gains afforded by traditional augmentation and GAN-based augmentation. However, when the resulting models were tested on images that were out-of-distribution, they found that while the baseline model scored 0.101, the traditionally augmented and GAN augmented models received scores of 0.535 and 0.747 respectively.[3] This increased flexibility is interesting as it suggests that the augmented examples constitute a wider domain than the examples they are based on. From a theoretical standpoint, this would have to be possible if GAN augmented datasets were to outperform others.

While all of the other related works cited have used GANs to create synthetic image data, Gupta, 2019 [5] applies GAN-based augmentation to language data for the purpose of sentiment analysis. Unlike the other applications, labeled language data for sentiment analysis is not scarce. The datasets used each contain between 2000 and 4000 real examples, and the classifier was pre-trained on a dataset of over 1.6 million examples. Real and fake examples were combined in a somewhat nontraditional

[2] Technically speaking, the generator and discriminator are most often trained one after another on a repeated basis.

[3] While the training set was based on CTs with contrast, these out-of-distribution images came from scans performed without contrast.

way; instead of concatenating the real and synthetic datasets, separate classifiers were trained for each and their outputs were combined via bagging. In these experiments, accuracy was only improved by up to 1.2% when fake examples were added. Though these gains are less impressive than others cited above, these results support the idea that GAN-based augmentation can be applied to domains outside of image data.

2 Method

2.1 Data Pipeline

All language data used was sourced from the SIGMORPHON 2019 repository and was initially formatted according the the Unimorph schema. The data for each language was gathered from their respective Wikipedia. Only the "train-low" sets were used for training. These datasets has only 100 examples, with each containing a lemma (base form), an inflected form, and a list of the morphological categories associated with the inflected form.[4]

To produce training data for the GAN, base forms were aligned with inflected forms using a technique developed by Mans Hulden called Chinese restaurant process alignment [6]. Based on the alignment, a series of likely stems are generated using code from Anastasopoulos and Neubig, 2019 [1]; as most of the languages tested use primarily suffixational inflection, the first of these "likely" stems was assumed to be the true stem and was included in the language's training set as an example of graphotactic compliance.

Non-alphabetic symbols, such as punctuation, numerals, and spaces were removed from each stem. Although the very small training sets provided by SIGMORPHON were likely hand gathered, the data was not free of anomolous examples. While attempting to gauge the model's behavior on a familiar language, it was found that the English dataset contained extremely uncommon characters such as "æ", "é", "ë", "ö", and "œ"; however, it was decided that was not necessary to exclude these characters, as the model should recognize their infrequency, even in a dataset of only 100 examples.

Finally, the stems were padded to reach a length of 10 and encoded as $10 \times len(\text{alphabet})$ one-hot sequences.

2.2 Stem GAN

Although proven methods exist for building GANs for sequence data such as SeqGAN [14], it was decided that these models would likely be too complex and therefore prone to overfitting and vanishing gradients when trained on such a small dataset. Instead, a minimalally complex GAN was created (Fig. 1). The generator took noise input of shape 10 (representing the maximum expected stem length) $\times$ $len(\text{alphabet})$ and consisted of two 100 unit LSTMs with $tanh$ activations, separated by a 20% dropout layer, and concluded with a single $len(\text{alphabet})$ unit dense layer with softmax activation, producing an output of equal shape to the input. The discriminator receives this output concatenated with one real example from the training set. Due to signs of mode collapse in early tests with two layers, the discriminator was given just one 100 unit LSTM. The discriminator terminates with a sigmoid activated single unit dense layer, that outputs a predicted probability that each example is real.

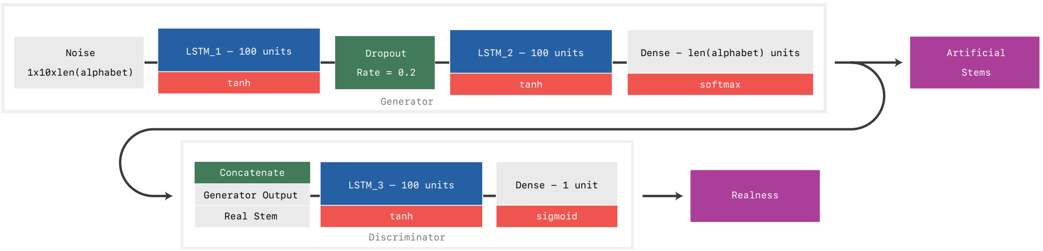


Figure 1: Architecture of stem the generation model

[4] Although the "train-low" sets do occasionally have 1000 examples, those corresponding to the languages used in this experiment have only 100.

The model was trained with a batch size of 1 for 1500 epochs on each low-resource language. After early tests resulted in mode mode collapse, the loss functions of both the discriminator and GAN as a whole were switched from binary cross entropy to Wasserstein loss [3].

2.3 Augmentation and Output Cleaning

The resultant models were each used to produce 10,000 artificial training examples. This process involved a modified version of the augmentation script from Anastasopoulos and Neubig, 2019 [1].

The strings produced by the resultant models were, unfortunately, not always of the correct shape. Some had zeroes (representing null characters) in the middle, others filled the entire 10 available characters (which most real stems do not). To combat this, model output strings were "cleaned". Strings were cut off at the first zero from the left (all languages were left-to-right), if a character was repeated more than twice consecutively, it was collapsed to just two, and if the length of the stem was longer than the real stem it was replacing, it was shortened to a matching length.

2.4 Testing

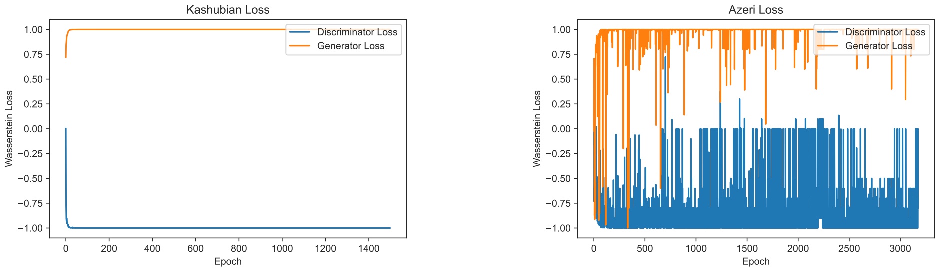
The inflectional model from Anastasopoulos and Neubig, 2019 [1] was used as a benchmark. For each language, three sets of "hallucinated" data were generated. The first used the random character selection method from Anastasopoulos and Neubig, 2019 [1], the second was the GAN produced set, and the third was produced by a simple trigram model trained on same 100 example as the GAN.

The model was run without cross-lingual transfer. All other parameters were left at their defaults. The number of epochs used by the training routine varied programmatically according to perceived accuracy.

3 Results

3.1 GAN Training

Models generally fell into two camps when training. With some, the generator loss quickly rose to 1 (the maximum) while the discriminator loss simultaneously fell to -1 (the minimum) (Fig. 2a). With the others, generator loss fluctuated rapidly between 0 and 1, while the discriminator loss fluctuated rapidly between -1 and 0 (Fig. 2b).



(a) Loss pattern 1

(b) Loss pattern 2

Figure 2: Examples of each loss pattern

In the first case, it seems that the model is learning for the first few epochs but quickly exhausts its potential. This is evident in the sample strings generated as well. As an example, for the first 100 or so epochs, the Kashubian model generated strings that were mostly a single character: "şşşşşşşşşş". But after about 300 epochs, string began to resemble stems with a reasonable number of zeroes (null characters) at the end: "dümç000000".

In the second case, the model is able to return to these more realistically structured strings occcasionally (denoted by the fluctuations), but it is not able to produce these strings with any consistency.

3.2 Performance

Neither the GAN nor the trigram model were able to consistently produce data that was more informative than the random strings (Fig. 3). For all languages except North Frisian and Occitan, training on the randomly augmented set resulted in the highest accuracy. The models trained on the trigram augmented set often performed similarly to those trained on the random set, having identical accuracy in Occitan and slighly higher accuracy in North Frisian.

No model trained on GAN augmented data achieved higher than 18% accuracy, and in 10 of 13 cases, these models were less than 10% accurate. In all cases, training on the 100 example "train-low" set alone resulted in a more accurate model than training on both the "train-low" and GAN augmented sets; on average, adding GAN augmented data resulted in a model that was only one fifth as accurate as models trained on unuagmented data.

Language	Unaugmented	Random	Trigram	GAN
Azeri	0.10	**0.31**	0.25	0.00
Bengali	0.25	**0.56**	0.55	0.03
Crimean Tatar	0.21	**0.63**	0.62	0.11
Karelian	0.34	**0.54**	0.38	0.02
Kashubian	0.46	**0.56**	0.30	0.14
Livonian	0.12	**0.32**	0.26	0.07
Maltese	0.24	**0.32**	0.19	0.00
Middle High German	0.62	**0.68**	0.66	0.18
North Frisian	0.15	0.16	**0.19**	0.00
Occitan	0.26	**0.78**	**0.78**	0.01
Old Church Slavonic	0.24	**0.44**	0.21	0.08
Pashto	0.10	**0.40**	0.36	0.03
Tatar	0.27	**0.71**	0.61	0.03
Average	0.26	**0.49**	0.41	0.05

Figure 3: Accuracy of resultant inflectional models using various data augmentation methods

On average Levenshtein distance calculations follow a similar pattern to that of model accuracy; however, this metric occasionally favors the models trained on trigram augmented data (Fig. 4). Additionally, in the case of North Frisian, a particularly difficult language to inflect (based on the accuracy results), the model trained on just the 100 original examples had a slightly lower average Levenshtein distnace.

Language	Unaugmented	Random	Trigram	GAN
Azeri	3.13	**1.76**	2.02	5.42
Bengali	2.66	**0.81**	0.95	4.59
Crimean Tatar	2.34	0.54	**0.50**	3.18
Karelian	1.74	1.94	**1.46**	4.10
Kashubian	1.44	**1.22**	**1.22**	2.78
Livonian	3.91	**2.23**	2.55	4.78
Maltese	2.31	**1.51**	1.91	5.29
Middle High German	0.80	**0.46**	0.60	1.92
North Frisian	**3.12**	3.21	3.54	7.88
Occitan	2.19	**0.53**	0.57	5.78
Old Church Slavonic	2.34	**1.09**	1.94	3.00
Pashto	2.53	**1.25**	1.34	4.79
Tatar	2.19	**0.50**	0.62	5.52
Average	2.36	**1.31**	1.48	4.54

Figure 4: Average Levenshtein distance (minimum number of elementary string operations to go from prediction to label) of resultant inflectional models using various data augmentation methods

4 Discussion

4.1 Analysis

GAN based augmentation, using the techniques described, is not a good option for creating artificial training data for the inflection model. It seems as though 100 examples is simply insufficient for the GAN to learn properly. Though there was initially concern that GAN based augmentation could only produce redundant examples, providing no additional information, the fact that it results in worse inflection performance than using unaugmented data means that the GAN is providing novel, yet misleading, information.

Observation of the artificial datasets reveals obvious flaws in the distribution of characters. For example, in the case of Azeri, the GAN augmented dataset contains 165 "ü" characters per 100 examples, while the unaugmented, randomly augmented, and trigram augmented datasets contain 44, 35, and 28 "ü" characters respectively. Similar flaws can be observed in the GAN augmented datasets for each language. These human-recognizable flaws could likely be averted using some sort of penalty based on relative character frequency in the training data, but there likely exist equally detrimental flaws that cannot be easily observed and would therefore be more difficult to penalize. The relative success of the trigram model suggests that many of these flaws can be avoided by using a less complex (or statistical) model that requires less data to function properly. However, given that random augmentation outperforms all other methods tested, it seems there is little reason to pursue such methods.

Anastasopoulos and Neubig's method, random augmentation, remains the best method for creating informative yet nondeceptive artificial examples [1]. The randomness indicates to the inflection model that the stem varies freely and has no bearing on the inflection. However, this is, of course, not always the case. Though many languages have "regular" inflectional paradigms (adding "-ed" for past tense in English), there are also many irregular forms, influenced by the stem ("run" [+past] = "ran"). In these cases, the stem must be taken into consideration to predict the correct inflectional morpheme. This means that an ideal informed (not entirely random) model should be able to produce more accurate and informative data than the random augmentor.

4.2 Future Work

Based on the conclusions above, there are two avenues of research that are begging to be pursued. The first involves modifying the task and model of Anastasopoulos and Neubig, 2019 [1] to work only with the non-stem characters in each example (those which are not shared between the lemma and inflected form). Working on the assumptions made by random augmentation (that the stem has no bearing on inflection), such a model should perform similarly to one using random augmentation if it is trained for an equal number of iterations.

The second avenue for future research involves rejecting the assumption made by random augmentation and instead assuming a finite number of inflectional paradigms, so that inflection may be turned into a classification task. Lemmas in the training set would first be categorized into paradigms; then, a classification model would be trained using only the lemmas as X and the paradigm's ID as y. Each inflectional paradigm corresponds to a function $f(T)$ where T is the set of morphological tags for the inflected form requested.

4.3 Conclusion

The emergence of new machine learning technologies are an opportunity to revisit attempts at low-resource language preservation and support. As language speakers and language learners, we know that a dataset of thousands of examples is not a necessary prerequistite for learning the inflectional paradigms of natural language. There is no doubt that small datasets like those used in the experiments above will one day be sufficient for inflection automation with human-like accuracy.

Code and Data

Written

Referenced

1. Code from Anastasopoulos and Neubig, 2019: `https://github.com/antonisa/inflection`
2. Stem alignment code from Mans Hulden: `https://github.com/mhulden/crpalign`
3. GAN training routine from Géron: `https://github.com/ageron/handson-ml2/blob/master/17_autoencoders_and_gans.ipynb`
4. Data for the SIGMORPHON 2019 shared task: `https://github.com/sigmorphon/2019`
5. Wasserstein loss implementation: `https://machinelearningmastery.com/how-to-implement-wasserstein-loss-for-generative-adversarial-networks`

References

[1] Antonios Anastasopoulos and Graham Neubig. Pushing the limits of low-resource morphological inflection. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 984–996, Hong Kong, China, November 2019. Association for Computational Linguistics.

[2] Antreas Antoniou, Amos Storkey, and Harrison Edwards. Data augmentation generative adversarial networks, 2017.

[3] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan, 2017.

[4] Cristóbal Esteban, Stephanie L. Hyland, and Gunnar Rätsch. Real-valued (medical) time series generation with recurrent conditional gans, 2017.

[5] Rahul Gupta. Data augmentation for low resource sentiment analysis using generative adversarial networks. *CoRR*, abs/1902.06818, 2019.

[6] Mans Hulden. crpalign. `https://github.com/mhulden/crpalign`, 2022.

[7] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. *CoRR*, abs/1611.07004, 2016.

[8] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *CoRR*, abs/1812.04948, 2018.

[9] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. Learning to discover cross-domain relations with generative adversarial networks. *CoRR*, abs/1703.05192, 2017.

[10] Arya D. McCarthy, Ekaterina Vylomova, Shijie Wu, Chaitanya Malaviya, Lawrence Wolf-Sonkin, Garrett Nicolai, Christo Kirov, Miikka Silfverberg, Sabrina J. Mielke, Jeffrey Heinz, Ryan Cotterell, and Mans Hulden. The SIGMORPHON 2019 shared task: Morphological analysis in context and cross-lingual transfer for inflection. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 229–244, Florence, Italy, August 2019. Association for Computational Linguistics.

[11] Veit Sandfort, Ke Yan, Perry J. Pickhardt, and Ronald M. Summers. Data augmentation using generative adversarial networks (cyclegan) to improve generalizability in ct segmentation tasks. *Scientific Reports*, 9(1):16884, Nov 2019.

[12] Hoo-Chang Shin, Neil A Tenenholtz, Jameson K Rogers, Christopher G Schwarz, Matthew L Senjem, Jeffrey L Gunter, Katherine P Andriole, and Mark Michalski. Medical image synthesis for data augmentation and anonymization using generative adversarial networks. In *Simulation and Synthesis in Medical Imaging*, Lecture Notes in Computer Science, pages 1–11. Springer International Publishing, Cham, 2018.

[13] Guotai Wang, Wenqi Li, Sébastien Ourselin, and Tom Vercauteren. Automatic brain tumor segmentation using cascaded anisotropic convolutional neural networks. *CoRR*, abs/1709.00382, 2017.

[14] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: Sequence generative adversarial nets with policy gradient. *CoRR*, abs/1609.05473, 2016.

[15] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *CoRR*, abs/1703.10593, 2017.