Intralingual Translation without Reference
via Gradient Ascent and Adversarial Training

Divya Gorantla, Tiffany Jaya, Daniel Olmstead

ABSTRACT

With the advent of big data, we have achieved impressive feats in interlingual translation from one language to another. In this paper, we wanted to explore the possibility of *intralingual* 'translation' of sentences between topics in the same language, using data collected from the social media platform Twitter. Specifically, we will automatically identify and rewrite sentences from one geographical region of North America to another. We will do so without using reference data, and generate the translator model in three different ways: enhancing a particular regional pattern in a LSTM and CNN encoder-decoder models and training a LSTM encoder-decoder model adversarially. Translations will be evaluated both by regional accuracy and by retranslation.

1. INTRODUCTION

Automatically rewriting a piece of text to reflect a different topic, style or sentiment while still maintaining the other characteristics of the original presents an interesting challenge, and one we were unable to find addressed in current research. While there is ample work in machine translation that has achieved amazing results (Wu et al. 2016), and in machine paraphrasing (Barzilay and Lee 2003) or changing writing styles (Xu et al. 2012), nearly all of these solutions involve leveraging parallel corpora or, if monolingual, limit themselves to word or phrase level (Conneau et al. 2017). Others have used variational autoencoders to generate text that interpolates between the latent space of two previously known sentences (Kingma and Welling 2013). Our problem is differentiated because rather than trying to say the *same* thing in a different way or language, we're trying to say a *different* thing in the same language.

Two works serve as inspiration for our approach on this challenge. The first is Google Deep Dream, which was designed as a way to use gradient ascent to generate images out of random noise to gain a better understanding of what exactly the model was learning ("DeepDream - a Code Example for Visualizing Neural Networks" n.d.). The second is based on Facebook's work on monolingual unsupervised machine translation (Lample et al. 2017). In both cases, the authors leverage their model's latent space to extract meaning.

As our vehicle for exploring this problem, we chose geographically-coded Twitter data, with the aim of relocating tweets that contain geographic information to a different region in North America. While highly-regional words like sports teams are an obvious target, the model should also pick up more subtle nuances of vocabulary, writing style and regional dialects. As a framework, the approach should work on any corpus of differentially-labelled data: for example, rewriting restaurant reviews from sushi to pizza, or product reviews from negative to positive.

To determine the feasibility of this task, we will approach this problem in three ways across two approaches: 1) using gradient ascent to enhance the regional aspects of latent states in a LSTM or CNN encoder-decoder model such as those developed for language translation (Sutskever, Vinyals, and Le 2014), and 2) training a LSTM encoder-decoder model adversarially. Since multiple characteristics can define the regional information within a tweet, the encoder is used to compress the high dimensionality so that the weights of the hidden layers contain enough information to represent the output in a lower density format. The decoder then steps in to reconstruct the compressed input in a hopefully meaningful way.

Another similarity to note is that, at their core, all three approaches face a classification problem. Although our goal is to translate tweets from one region to another, in order to accomplish this, the model needs to learn during training the origin of a particular tweet. Prior research on identifying geolocation and regional dialects from Twitter data was based on probabilistic models (Eisenstein, n.d.) and content based approaches (Cheng, Caverlee, and Lee 2010). However, we chose to employ deep neural networks to solve this particular problem. The model predicts the

regional origin of the dialect and if it's not accurate, the error that was computed with respect to the cost function is propagated backwards until the weights change to reflect the accuracy of the output. The first category of approach takes the output of the encoding and, utilizing the differentiated gradients learned during training, emphasizes the features that fall in line with the targeted region more and more until it appears to originate from that region.

In contrast, the second approach we took develops encoders for the source and targeted dialects and swaps the output of each encoder as input into the opposite decoder. The decoder for targeted dialect will try to recreate sentences using the source encoding while the decoder for the source dialect will try to recreate sentences using the targeted encoding. The switch occurs via a discriminator which gets fooled in classifying the regional dialect. As a result, the source dialect is forced to translate into the target dialect.

2. METHODOLOGY

2.1. DATA

Data consists of ~3 million geocoded original tweets (no quote-tweets or retweets) collected from Twitter's Real-Time API during November 2018. Every tweet was assigned to one of 23 regions in North America according to proximity to a particular city. The dataset was then balanced across the 13 most populated regions, resulting in a final dataset of ~1.7m tweets, with 135k tweets per region. Only the full text of the tweet and its corresponding label were used as features in the models. URLs and user tags were removed during preprocessing, however since language used in Twitter is often symbolic or strange, we kept everything else within the 10k most frequent tokens, including emojis and unusual spellings that might convey regional signal.

2.2. EVALUATION METRICS

Models were evaluated on a standardized test bed of 130 tweets, 10 from each region and at 10 levels of predictive probability according to Naive Bayes. They are then scored on two main criteria: BLEU scores on sentences that have been translated into a target category and then back into the source category (Rapp 2009), and percentage of times the model's encoder successfully predicts the translation as belonging to the target region. Combined, these measure the model's ability to accurately target the regional identifiers in a given sentence and adjust them with minimal warping to the sentences' coherence. We also measure 'Fidelity,' which is simply a measure of the model's ability to reconstruct an input sentence without any alteration of the latent states.

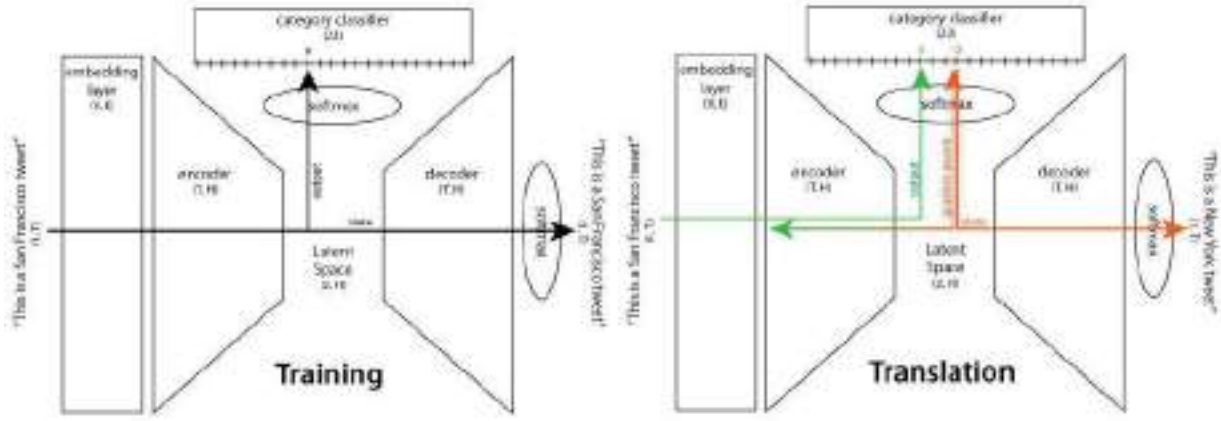
2.3 CLASSIFIERS

As regional classification is a key mechanism in our model, we set out to determine the best classifier for identifying the source of a tweet's content. The most successful classifier was a two-layer word-level LSTM, consisting of 200-vector GloVe embeddings, a single bidirectional layer followed by a regular layer. The CNN classifier that gave us comparable results to LSTM was a 2-layer CNN with max pooling and 2 dense layers with Twitter 200 dimensional GloVe word embeddings. After hyperparameter tuning we found larger word embeddings and more hidden features in each layer improved the classification.

	Accuracy		Accuracy
Random choice	7.8%	Neural Bag-of-Words	9.5%
Human	13.5%	LSTM (Character Level)	10.03%
Unigram Naive Bayes	18.51%	**LSTM (Word Level)**	**21.3%**
Bigram Naive Bayes	13.05%	CNN (Character Level)	19.4%
Logistic Regression	18.62%	CNN (Word Level)	21.2%

2.4. MODELS

2.4.1. ENCODER-DECODER MODEL



Our data has no reference sentences to train the decoder, but as we are staying within the same language, we can leverage the intelligence of the model to alter the sentence. The encoder/decoder is trained to simply reconstruct its own input, so the decoder learns how to build an English sentence from latent space vectors. Rather than being discarded as usual, the output of the encoder is passed through a softmax layer to a classifier, thereby training the encoder to differentiate sentence content based on labels.

During translation, a sentence is passed through the encoder to the classifier, and the gradients of the input layer are calculated with respect to the loss of a target region. Those gradients are iteratively added to the input and the loss recalculated, until the probability that the hidden states now match the target region is maximized. The altered states are then fed into the decoder, which makes its best effort at reconstructing an English sentence from them.

Although language translators of this sort are often done at the character level, our preliminary results indicated a word-level approach might be more productive.

2.4.1.1. LSTM

In all, seven LSTM models were trained at varying levels of complexity, dataset size, regularization, and training length. The most useful results came from two models: a 300k tweet set with a relatively simple Bidirectional encoder, single LSTM decoder and zero regulation, and a more complex 500k tweet set

2.4.1.1.1 SUBJECTIVE RESULTS

Like Deep Dream, the results that come out of the model are often somewhat hallucinatory, but there are signs that it is following some sort of internal logic. There is a lot of noise in tweets, and as a result the models can have a difficult time accurately targeting the translation--or at least sufficiently altering the text to classify as originating from the correct region. Despite this, regional patterns are clearly reflected in the results, as is apparent in the tweet *'nobody has better food than waffle house'* translated into every other region:

Target Region	Translated Tweet	Predicted Region	Probability
Chicago	nobody has better meat food house 🤦🏾♂	Nashville	12.81%
Cincinnati	nobody has better food than waffle house	Charlotte	23.44%
Houston	nobody has every better waffle juice house bro	Charlotte	24.90%
Los Angeles	nobody has better food than james house	Charlotte	19.30%
Nashville	nobody has good food than waffle house	Charlotte	20.98%

New York	nobody has rich food for illegal	Charlotte	12.38%
Oklahoma City	nobody has better food than pizza house	OK City	14.37%
San Francisco	nobody has better taste two bucks house	Cincinnati	29.38%
Seattle	nobody has better food than james house	Charlotte	19.30%
Tampa	nobody has better food than waffle house bill	Charlotte	25.06%
Toronto	nobody has food better meat at against	Cincinnati	13.45%
Washington	nobody has better food for diet	Houston	12.35%

Although the model failed to meaningfully relocate the tweet, it did recognize that, by and large, Waffle House is a southeastern US phenomenon and is not reflected in northern or western regions (it turns out there are Waffle Houses in Cincinnati). There is a chain called 'Pizza House' in Oklahoma City. New York is curious.

2.4.1.1.2. GRADIENT ASCENT

Because Gradient Ascent is iterative, and we compute the loss at each step, we can set the threshold for how far to translate a given sentence into its target category, and choose how much regional 'flavor' to imbue the result. Here are the interim steps of translating a job search from Indiana to New York:

Step	Decoded Latent State
0	**(source)** interested in a #job in #indianapolis, in? this could be a great fit: #construction #careerarc
1	interested in a #job in #winchester, va? this could be a great fit: #construction #hiring #careerarc
2	interested in a #job in #winchester, va? this could be a great fit: #physician #nyc #careerarc
3	interested in a #job in ct? this could be a great film. @ new #facilitiesmgmt #hiring #careerarc
4	interested in a #job in #winchester, ct? this could be a great fit: nj #hiring #careerarc

Iterations after 100% loss quickly lose any relation to the source sentence or the target region, but there is a nebulous 'sweet spot' around between loss of 85% and a couple steps past 100% where coherence and translation are maximized. Small modifications to the normalization that is applied to the gradients at each step can have a profound effect on the results. We got the best results from simple x/max(x), although this could be scaled.

2.4.1.2. CNN

The encoder–decoder model with CNN consists of a two-convolution-layer encoder and a two-convolution-layer decoder with batch normalization and dropout. The latent space consists of three dense layers and a softmax layer to classify the regions used for applying gradient ascent for the translation.
The common implementation of the encoder-decoder (autoencoder) model with CNN found in literature typically consists of the max pooling layer on the encoder and up-sampling layer on the decoder, along with convolution layers. But in our case, we found having just the convolution layers had better accuracy in replicating the input tweet. We used 100-dimensional Twitter GloVe word embedding trained on 2 billion tweets. This model is mainly for inspecting how it translates tweets in-place, that is, keeping the length of sequences constant is it able to change words according to the target region.

2.4.1.2.1 TRANSLATIONS

The in-place translations from the CNN model are not coherent, but one interesting observation is that when translating to Houston the model incorporates words like 'texas', 'tx', 'cowboys' and encoder classifier is also highly accurate (0.9) in identifying the translated text as Houston region. This might indicate that the encoder got trained better on this region.

Source	Target	Original Tweet	Translated Tweet	Predicted Region
Cincinnati	Houston	made this for & hope they like it	tx carolina wore ion shoutout cowboys ion love	Houston

2.3.3. GAN ENCODER-DECODER MODEL

Before we break down the basic premise of our third approach, we first had to decide whether or not to use a character or a word model. The beauty of the character model is that it does not have to worry about unknown word tokens, which might be the case with regional dialects. We assume regional dialects are more prone to specialized vocabulary and therefore more unknown word tokens, but it has drawbacks that make us lean towards using the word model. With a large dataset, we do not have the monetary resources to computationally train an expensive character model. In addition, a character model cannot capture how the early parts of the sentence affects the later part of the sentence. For this reason, we began building the third approach with a word model in mind.

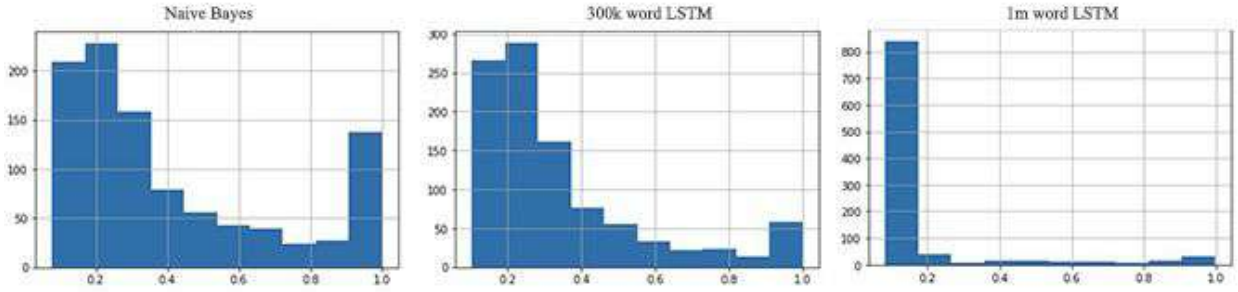
Another decision that we made early on is to build the sequence-to-sequence model with attention. Without an attention mechanism, the decoder can only use the last hidden state of the encoder. Since we have reasoned to use a word model, having an attention mechanism gives the decoder all hidden states of the encoder by computing the weighted sum of the encoder states. We also chose to create word embeddings for each particular region (additional information about word embeddings is enclosed in the appendix).

Once we have made these early decisions, we began to develop two encoder-decoder models in parallel: one for the source regional dialect and another for the target regional dialect. The encoder for each region translates each tweet into the latent space. To prevent the model from overfitting, we added noise by dropping every word in the input sentence with a probability of 10% and shuffling the input words randomly no more than three words away. Once it passes through the encoders, the discriminator, which is another classifier network, takes the result and determines whether the latent vectors correspond to the actual dialect before passing it into the decoder which will translate the latent vectors into the translated tweets. The discriminator would like to minimize its error rate in order to classify the latent vectors correctly, but the encoders beg to differ. In this case, the encoders play a role as the generators. We would like the source encoder to change its latent vectors into the targeted dialect's latent vectors. By applying gradient ascent on the discriminator's error rate, the encoder would not only learn how poorly it performs in convincing the discriminator (that it is the target dialect) but it also learns how to tweak its weights so that it can convince the discriminator that its resulting latent vector belongs to the targeted dialect. In performing the weight adjustment, the source encoder shifts its latent vector closer in space to the targeted dialect's latent vector. When the source encoder is able to shift its latent vector so closely to the targeted dialect's latent vector, only then can it fool the discriminator in passing its latent vector to the targeted decoder. The loss function that we decided to use on the discriminator is the cross-entropy error because the weight changes do not decrease over time so training is not likely to stall. And we chose LSTM encoder-decoder model simply because as classifier at the word-level, LSTM has performed better than the other classifiers. After receiving the latent vector from the discriminator, the decoder iteratively takes input from the previously generated word and predicts the highest probability of being the next one, producing the translated tweets as a result.

3. RESULTS AND DISCUSSIONS

Model	Type	Words	Epochs	Fidelity	Trans. Accuracy	Retrans. Accuracy	BLEU for translation	BLEU for retranslation
LSTM	1L x 1L	300k	12	.83	56.92	39.2	.21	.18
LSTM	1Bi x 1L	300k	8	.902	50.83	31.5	.34	.29
LSTM	1Bi x 2L	300k	8	.96	10.77	2.3	.45	.37
LSTM	1Bi x 1L	500k	5	.97	38.33	15.4	.42	.34
LSTM	1Bi x 1L	500k	7	.974	25.83	3.1	.47	.4
LSTM	1Bi x 2L	500k	5	.97	14.62	2.3	.47	.39
LSTM	1Bi x 2L	1m	1	.964	44.62	19.2	.3	.26

Surprisingly, we found that in some ways for this particular problem, less is more. Generally speaking, the more we trained the models--whether on more complex models, on larger datasets, or for more epochs, the more they "hardened" and became reluctant to reconstruct the altered hidden states into something else. On the other hand, simpler models may be more willing to change the text, but the outputs are generally more chaotic, so there is a trade-off to consider. According to the metrics we've chosen, the ideal model would have high translation accuracy (ie, it actually shifts the text to the target region), a lower BLEU score for the translation and a higher BLEU score for the retranslation, indicating a return of some of the original text (the perfect model would recreate the source sentence completely).



Part of the highly-trained models' rigidity is in their deep uncertainty over what differentiates the classes. These histograms represent probability levels for predicted tweets, and it is clear that the 300k model is much more similar to Naive Bayes in its level of confidence than the 1m-tweet model. Although their respective accuracies are not far off (~20%), when you ask the highly-trained LSTM to alter and amplify the regional elements of a tweet, it is unclear what to do. The result is a potpourri of words in the higher levels of probability that might have topical relevance, but are literally all over the map regionally (ie, attempting to translate a tweet containing the word *Packers* will retrieve *Jets*, *Giants*, *Bears*, *Packers*, *Steelers*, and even *Yankees* and *Warriors* as candidates), and the model isn't sure which of these to promote to the top. Because of this, the more foolhardy 1Bix1L 300k model often provides the most interesting results, even if its BLEU scores are not as high.

Model	Type	Words	Epochs	Fidelity	Trans. Accuracy	Retrans. Accuracy	BLEU for translation	BLEU for retranslation
CNN	2Conv x 2Conv	100k	4	0.941	17.69	19.23	0.081	0.102
CNN	2Conv x 2Conv	300k	1	0.942	16.92	20	0.042	0.060

The CNN model's ability to perform vanilla autoencoding is promising, but the inplace translation needs more fine tuning, the encoder loss has been a difficult metric to tame to a good level. We expect that improving this loss can show noticeable improvements in translation. The CNN model has been computationally expensive, but more epochs and a more complex network might improve the translation.

The third approach, which is to train a LSTM encoder-decoder model adversarially, requires two encoder-decoder models to train in parallel. As a result, we found that it became unstable to train. Our model parameter ended up never converging during training, which is one of the common problems found in general adversarial networks.

4. CONCLUSIONS AND FURTHER WORK

If the chaotic nature of gradient ascent could be tamed, the LSTM model shows the most promise as a means of algorithmically rewriting text to related topics, styles, or sentiment within a labelled corpus. Further research is necessary to determine a way to balance the differing losses between the encoder and decoder during training, and boost the ability of the model to differentiate regional characteristics within a piece of text. Facebook AI's technique of training the model to decode noisy versions of itself could be beneficial, as could trying it on a more easily differentiable corpus (ie, food reviews). Attention could also be a better way of surfacing the elements that carry the highest categorical signal for change. In regard to CNN, we suspect that bigger networks with 3-4

convolution layers and sizable hidden features with larger dataset (around one billion tweets) could have better results.

5. REFERENCES

Barzilay, Regina, and Lillian Lee. 2003. "Learning to Paraphrase: An Unsupervised Approach Using Multiple-Sequence Alignment." In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics. http://aclweb.org/anthology/N/N03/N03-1003.

Cheng, Zhiyuan, James Caverlee, and Kyumin Lee. 2010. "You Are Where You Tweet: A Content-Based Approach to Geo-Locating Twitter Users." In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management - CIKM '10*, 759. Toronto, ON, Canada: ACM Press. https://doi.org/10.1145/1871437.1871535.

Conneau, Alexis, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2017. "Word Translation Without Parallel Data." *ArXiv:1710.04087 [Cs]*, October. http://arxiv.org/abs/1710.04087.

"DeepDream - a Code Example for Visualizing Neural Networks." n.d. *Google AI Blog* (blog). Accessed December 6, 2018. http://ai.googleblog.com/2015/07/deepdream-code-example-for-visualizing.html.

Eisenstein, Jacob. n.d. "A Latent Variable Model for Geographic Lexical Variation," 11.

Kingma, Diederik P., and Max Welling. 2013. "Auto-Encoding Variational Bayes." *ArXiv:1312.6114 [Cs, Stat]*, December. http://arxiv.org/abs/1312.6114.

Lample, Guillaume, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2017. "Unsupervised Machine Translation Using Monolingual Corpora Only." *ArXiv:1711.00043 [Cs]*, October. http://arxiv.org/abs/1711.00043.

Rapp, Reinhard. 2009. "The Backtranslation Score: Automatic MT Evaluation at the Sentence Level without Reference Translations." In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, 133–136. Suntec, Singapore: Association for Computational Linguistics. http://www.aclweb.org/anthology/P/P09/P09-2034.

Sutskever, Ilya, Oriol Vinyals, and Quoc V Le. 2014. "Sequence to Sequence Learning with Neural Networks." In *Advances in Neural Information Processing Systems 27*, edited by Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, 3104–3112. Curran Associates, Inc. http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf.

Wu, Yonghui, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, et al. 2016. "Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation." *ArXiv:1609.08144 [Cs]*, September. http://arxiv.org/abs/1609.08144.

Xu, Wei, Alan Ritter, Bill Dolan, Ralph Grishman, and Colin Cherry. 2012. "Paraphrasing for Style." In *Proceedings of COLING 2012*, 2899–2914. Mumbai, India: The COLING 2012 Organizing Committee. http://www.aclweb.org/anthology/C12-1177.

Eisenstein, Jacob, Brendan O'Connor, Noah A Smith, and Eric P Xing. "A Latent Variable Model for Geographic Lexical Variation," n.d., 11. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1277–1287, MIT, Massachusetts, USA, 9-11 October 2010. http://aclweb.org/anthology/D10-1124.

Cheng, Zhiyuan, James Caverlee, and Kyumin Lee. "You Are Where You Tweet: A Content-Based Approach to Geo-Locating Twitter Users." In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management - CIKM '10*, 759. Toronto, ON, Canada: ACM Press, 2010. https://doi.org/10.1145/1871437.1871535.

Rapp, Reinhard. "The Back-Translation Score: Automatic MT Evaluation at the Sentence Level without Reference Translations." In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers on - ACL-IJCNLP '09*, 133. Suntec, Singapore: Association for Computational Linguistics, 2009. https://doi.org/10.3115/1667583.1667625.

6. APPENDIX

6.1 TOP-LEVEL WORDS

The output of the decoder is greedy, returning the highest probability word at each timestep. However, softmax calculates the probability of every word in the vocabulary at each step, so we can retrieve the top n words to see what the decoder is leaving behind. Here are the top-level words returned for a tweet, *'this season the packers fans have zero room to talk shit about football,'* translated to Los Angeles.

300k Words

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	this	playing	the	buckeyes	fans	kickoff	to	be	there	bs	about	shit	football	players.
1	better	season	season	eagles	game	zero	out	have	bs	w	talk	damn	sports	yet
2	playing	football	football	season	season	on	at	cum	no	...	no	hating	espn	🤦‍♂️
3	the	game	eagles	football	kickoff	without	have	any	like	n	shit	dude	vs.	today?
4	season	buckeyes	missed	packers	swing	at	could	10pm	@	....	w	hell	thing	🔥
5	playoffs	running	buckeyes	kickoff	football	booked	any	has	...	shit	than	talk	radio	blue.

500k Words, 5 epochs

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	this	season	the	eagles	fans	have	short	artist	to	talk	shit	about	football
1	the	again	a	packers	supporting	as	preferred	space	see	?	here	at	hunger
2	a	finished	this	bears	voted	be	commercial	studio	out	cry	cold	from	reason
3	an	opening	at	detroit	fan	by	no	room	back	chicago	out	for	what
4	next	starting	notre	fans	showing	create	an	public	i	tv	ass	out	gratitude
5	it	beginning	from	#bears	support	any	painting	interview	going	girls	over	around	every

500k Words, 7 epochs

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	this	season	the	eagles	fans	have	next	drama	to	talk	shit	about	football
1	the	starting	celebrate	fans	❤	has	every	inspiration	then	nobody	"	every	✨
2	all	yankees	congrats	steelers	best	experience	an	sex	can	mind	mood	hours	night.
3	we	past	forever	fallout	appreciation	will	this	emotion	someone	reality	moment	this	🍒
4	our	the	forget	best!	🌟	need	been	rare	words	hell	cold	reading	conversation.
5	let's	kicking	hannah	packers	excited	be	no	song,	i	speak	sleep	at	day!

1m Words

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	this	season	the	jets	ive	have	shown	room	to	talk	shit	about	the
1	the	finished	a	giants	fans	can	sporting	chicago	on	watch	about	than	football.
2	it	the	we're	packers	remember	open	sports	coffee	game	shit	time	besides	bears
3	just	windy	you're	matchup	i've	shown	spot	spot	any	stuff	worth	football	to
4	a	michigan	#sunrise	#bears	#thankful	compete	beer	outside	that	hype	than	what	ucla

While there are a variety of sports teams in the proper areas of the sentence for each model, notably absent is any mention of the Los Angeles Rams. For that matter, there are no teams west of Ohio, suggesting that either Los Angeles was apathetic about their football team this November, or the models were unable to fully absorb the regional characteristics of the various sports teams. However, the models did pick up that slot, #3, as the portion of the sentence carrying the greatest regional signal, and tried to put something else in there in an effort to shift the sentence's origin.

With a language model and/or a POS tagger trained on this corpus, these added layers of words would enable an optimization algorithm to adjust the decoder output towards greater coherence, balanced against maintaining regionality.

6.2 SAMPLE TRANSLATIONS

To further illustrate the differences between the models, here are some example sentences that contain regional information translated to every region, across the 300k 1Bix1L model (quick to translate but chaotic), 500k 1Bix1L 5 epoch model (more restrained but more precise), and the 500k 1Bix2L model (much more rigid, but reluctant):

Source text: *i drank about a gallon of soda during this game*

Target Region	300k 1Bi x 1L	500k 1Bi x 1L	500k 1Bi x 2L
Charlotte	i drank about a couple of mashed potatoes by this year	i drink about a gallon of chicken cereal this week.	i drank at a lot 😐 express time this game
Chicago	i drank about a bottle of 🙃 during this game	i drink about a hot of wood during this game	i drank about a production nursing french moments this game
Cincinnati	i drank about a beer of rose during this game	i drank about a gallon of soda during this in	i drank about a gallon of during mug this game
Houston	i drank about a bottle of 🙃 before this young	i craving about a lb of soda houston in football	i drank about a gallon of 5am during this game
Los Angeles	i drank about a wine of pour during this game	i cursed i'm a type of to myself in this	i cry about a gallon of 5am during this game
Nashville	i drank about a bottle of 🙃 which day training	i drank about a 38 of soda during this game	i am! about a meal 😩 of 2) this game
New York	i drank about a like, waste of service this day 😭😭	i gained about a beauty of wood instead this is	i drank about a gallon of industrial during this game
Oklahoma City	i smoked about a bourbon of coffee. much before this football	i drank about a gallon of couch during this game.	i drank at a gallon of time, just saw an
San Francisco	i drank about a soda of 🙃 for this day	i drank about a gallon of soda ii in this	i drank about a gallon of soda during this game
Seattle	i drank about a glass of billion for this game	i drank about a grey of bed, during this is	i drank about a gallon of soda during this game
Tampa	i drank about a event, of custom last year as	i yelled about a gallon of to coffee this week.	i drank about a bubble of workers at this game
Toronto	i drank about a spectacular course for almond in every overtime	i drank about a huge of wood cooking this game	i drank about a ford of soda during this game
Washington	i drank about a sip of drank this year by	i drank about a problem with pepper 😬 this is	i drank about a gallon of soda during this game

Source text: *andrew gillum let's bring this home! #florida #miami*

Target Region	300k 1Bi x 1L	500k 1Bi x 1L	500k 1Bi x 2L
Charlotte	andrew didn't know! bring this too. 🙏🟫 stfu shits	up, let's making well! y'all ✔️ #picoftheday dam	andrew 🤣🤣🤣🤣🤣 don't bring this // #florida #miami
Chicago	andrew lost. turned up, this bring #florida leadership.	premier cock fires. bring this willie landed #newyork	andrew gators didn't by: this today!!! treasure #florida

Cincinnati	andrew lost. "we bring this song. anymore. #florida	andrew 2: let’s bring this up, thanks, chapel	andrew gillum won’t prepare this tennessee us! #florida
Houston	andrew lost. ima bring this question? speech ❗	andrew owns god’s couldn’t thank in. turkey fence	texas 8. let’s bring this rocket 🙏🟫 amen.
Los Angeles	kevin ♥ die. love singer to die. #dj 😩	andrew #love let’s couldn’t this spot. #happythanksgiving #sunrise	trudeau survived let’s bring this loss. lovers #florida
Nashville	andrew gillum don’t get this courage in whore	jay 🎧 let’s 😂 bring me!!! 🦃 😮	andrew jr.
New York	Andrew bday i bring this sad anymore. #comedy #funny	andrew beyond	we bring this super god! sam
Oklahoma City	andrew gillum lost. get it truth on coffee.	we couldn’t follow this #picoftheday #bbn	andrew gillum forced hope this #happythanksgiving2018 #florida #miami
San Francisco	andrew kit 🙏🟫 take a brave day, #actor ✨	alright. congratulations shit. let’s this barber 🦃 #miami	edwards moore let’s bring this again. enjoy! florida
Seattle	richard ⚡ i’ll bring this question? #live #mood	btw, broken house, bring & i’ll move. landing	#dance soros allowed bring this chapter #florida #florida
Tampa	andrew lost. lost. can make a #florida #sundayfunday	andrew represent now, thanks this 🦃 wall, 🇨🇦	trudeau commission lands easy this weekend! #florida #florida
Toronto	andrew lost. quickly want this beauty #rapper	andrew gillum let’s bring this washed #florida #miami	andrew gillum let’s bring this this #florida #florida
Washington	andrew gillum lol bring this courage now. #actor #florida	andrew 8, two. let’s this washed #florida #vote	ontario carter let’s bring this weekend! a #florida

Source text: *does anyone care that california is burning?*

Target Region	300k 1Bi x 1L	500k 1Bi x 1L	500k 1Bi x 2L
Charlotte	does anyone care that prevent america does.	dare anyone care that by is bitch!	does recommend everybody that is or racist.
Chicago	does anyone spending of that trump is aaron	does anyone any american is by hill,	does anyone care that california is foh
Cincinnati	does guilt adults that is living fault.	does anyone wears that all is cindy	does anyone care that california is trash.
Houston	does anything family and change 🤫	does real 🤦🏾‍♀️ that r business equality	does care once that here is plate
Los Angeles	does anyone care that is california closing	does anyone feel that 🤓 music #maga	does anyone care that california is #ncte18
Nashville	does anyone care that california is what happened.	does anyone exclusive that for @ isis	does anyone care that is california pathetic.
New York	does anyone care at what is lit	does anyone care that as is 😻	does anyone care that is purchased #dogsofinstagram
Oklahoma City	does anyone care is what we’re on hahaha	does so... cares that god for arkansas	does anyone care that california is pathetic.
San Francisco	does anyone care that california is perfect.	does anyone care that california is 🙏	does anyone care that california is #hiring
Seattle	does anyone care that is cleaning	does anyone else. that california are #seattle	does anyone care that california is water.
Tampa	does harm care that governor is hilarious.	anyone does care no that is #goodmorning	does anyone care that california is #disney
Toronto	does anyone care that comes now	does anyone any that says is blues	does anyone care that california is trash.
Washington	does anyone care that is here?	mr problem offer that god is corporations	does care whatever that california is involved.

6.3. WORD EMBEDDINGS

For the adversarial approach, we started off with word embeddings before developing our translation model because back-propagation is possible over continuous representations (as in the case of CBOW) than discrete ones (versus BOW).

Word embeddings overcome the many limitations that Bag of Words produced, in particular large sparse vectors that do not describe the meaning of the words. It provides a projection in the vector space where words with similar meanings cluster together. There are two main algorithms used for training the word vectors: Continuous Bag of Words (CBOW), which predicts the word given its context, and Skip-gram, which predicts the context given the word. We will employ the latter than the former because we have limited training data. Even though CBOW trains several times faster than Skip-gram and is slightly better in terms of accuracy for the frequency of words, it also requires a larger text corpus ranging from 1-100B words. Currently our training data is limited to 767,446 words.

```
Word2Vec(vocab=767446, size=100, alpha=0.025)
```

GloVe is another successful word embedding algorithms with generally better word embeddings because it combines both the global statistics of matrix factorization techniques like LSA with the local context-based learning in Word2Vec. Since we would like the word embeddings to learn from live Twitter feeds used, we chose Word2Vec.

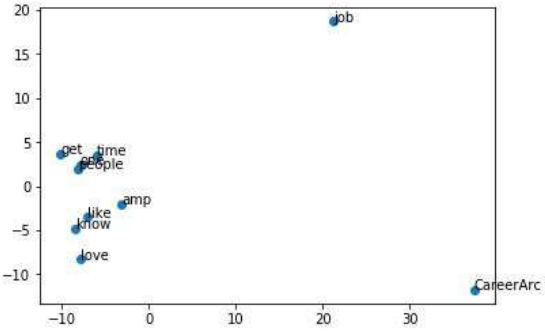
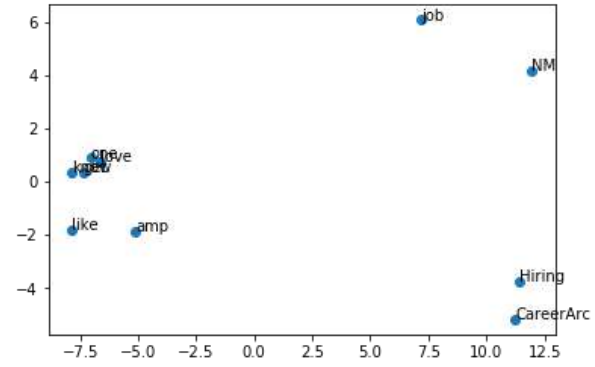
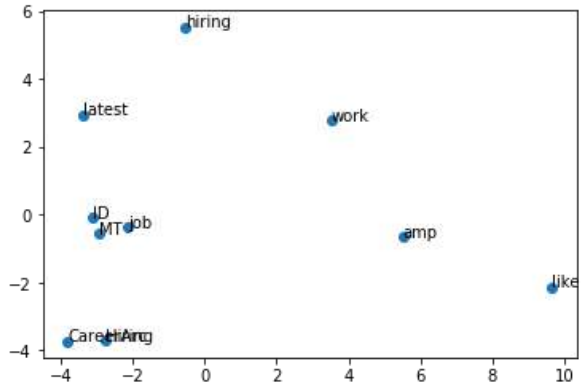
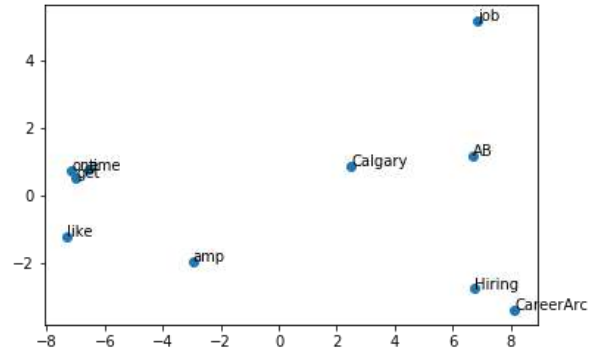


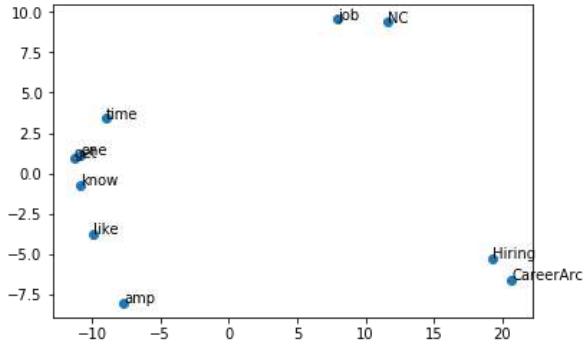
Fig 1. Word embeddings of the top 10 most frequent word used in Twitter in all regions

Albuquerque	Billings
	

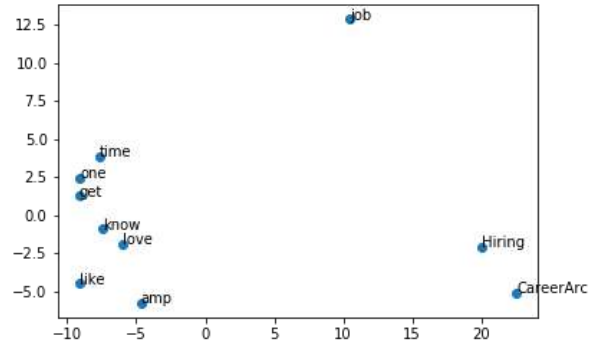
Calgary



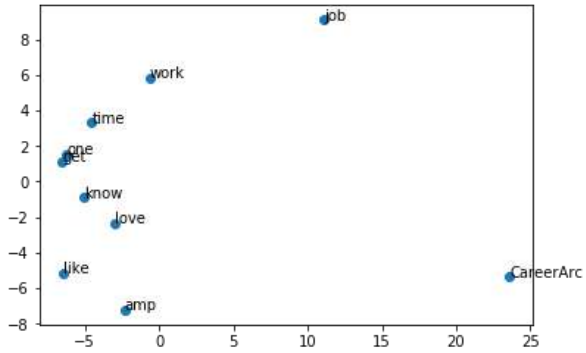
Charlotte



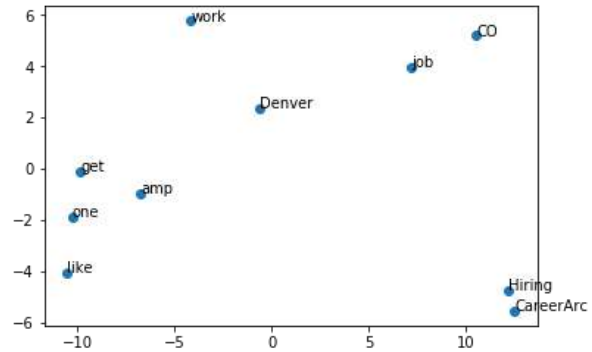
Chicago



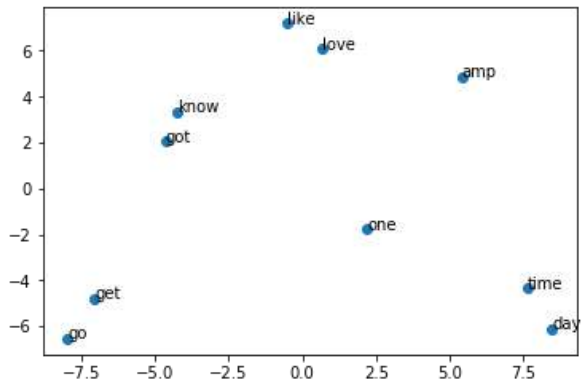
Cincinnati



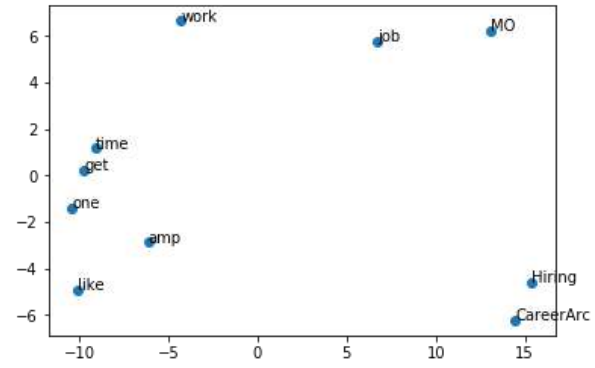
Denver



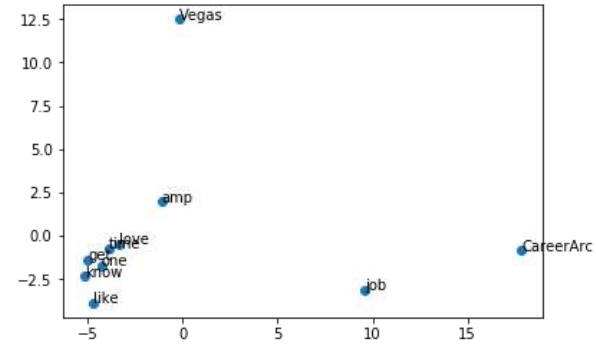
Houston



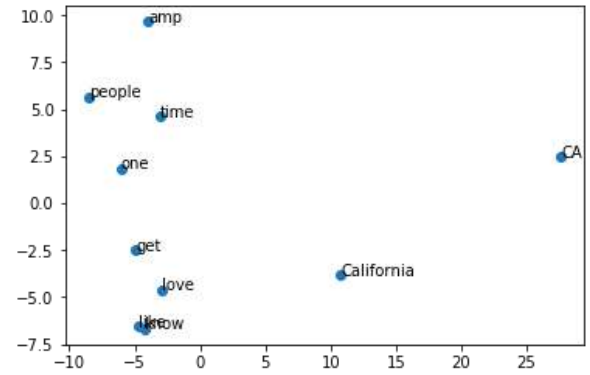
Kansas City



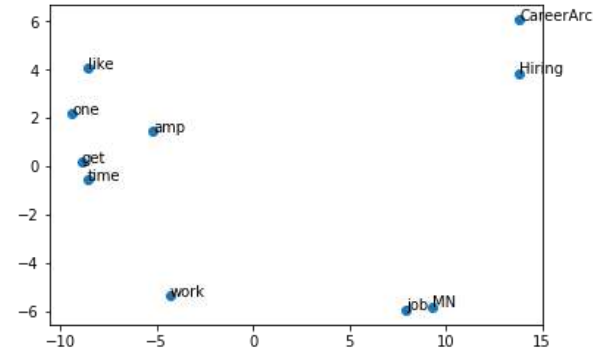
Las Vegas



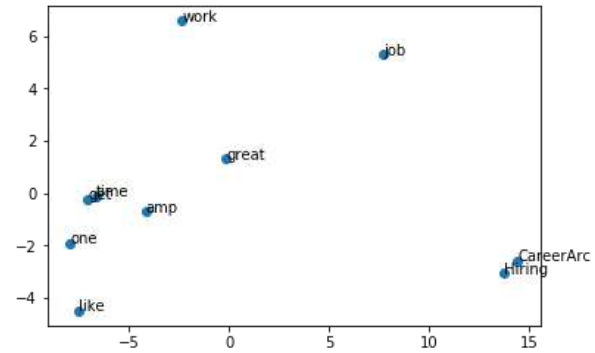
Los Angeles



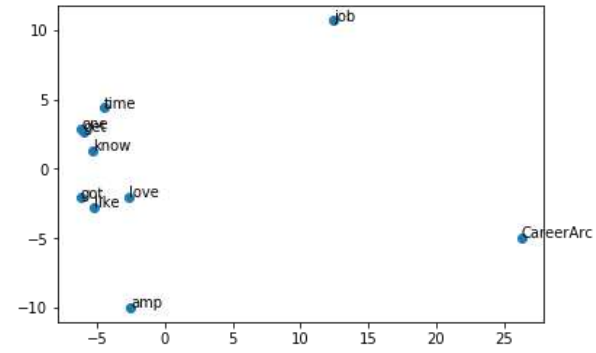
Minneapolis



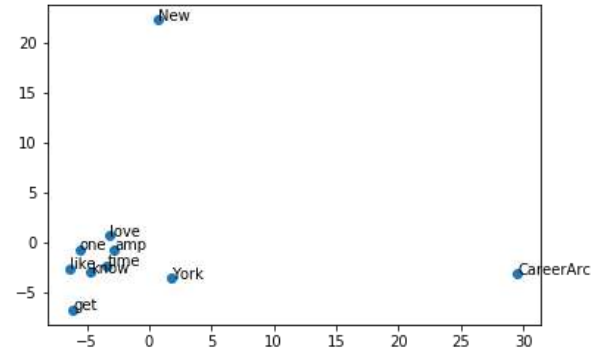
Montreal



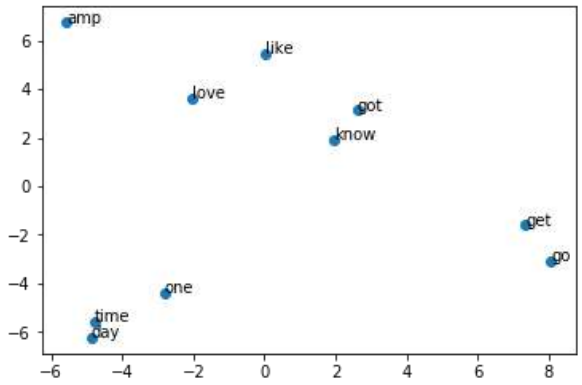
Nashville



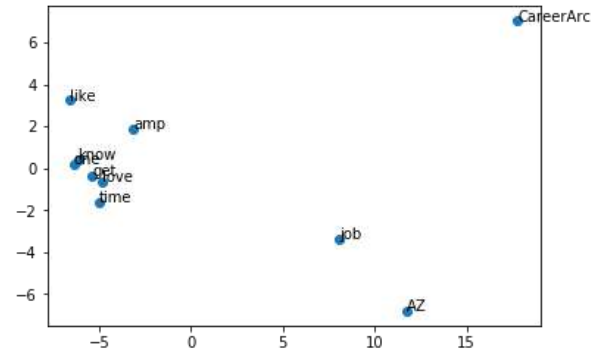
New York



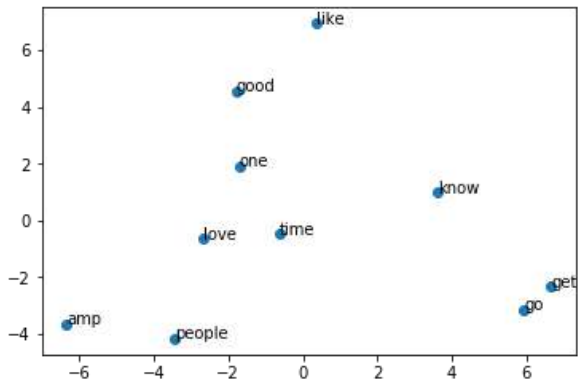
Oklahoma City



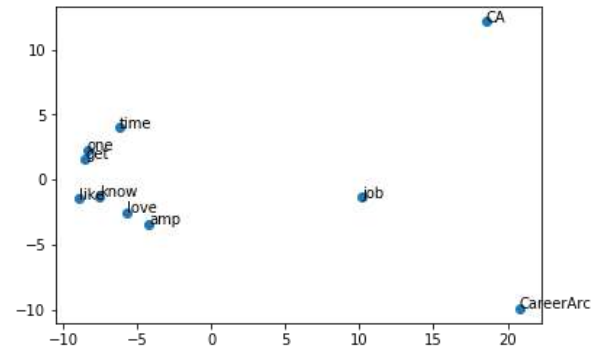
Phoenix



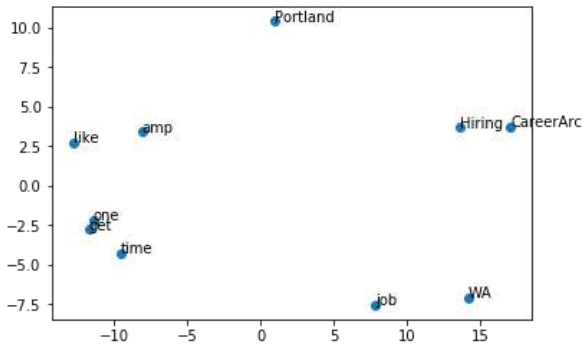
Pittsburg



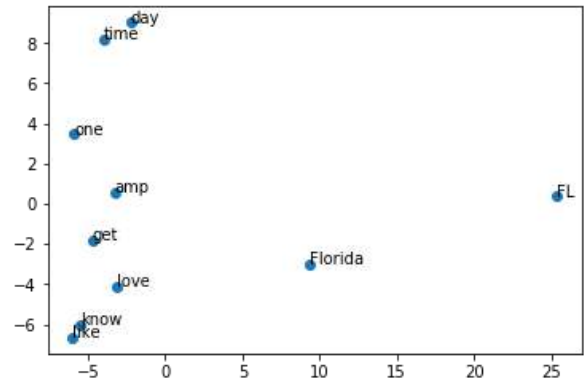
San Francisco



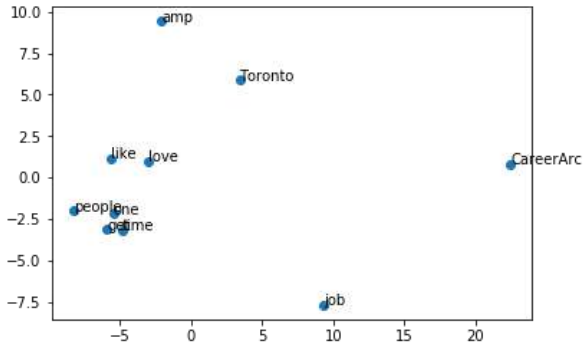
Seattle



Tampa



Toronto



Washington

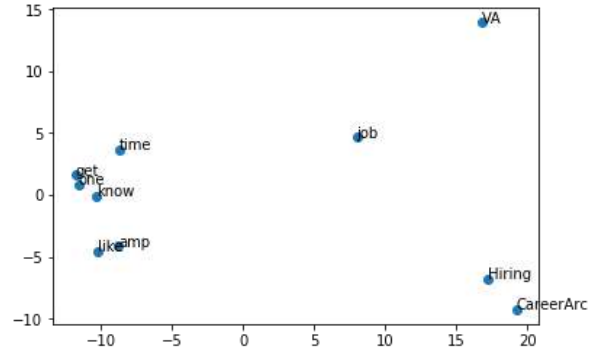


Fig 2. Word embeddings of the top 10 most frequent word used in Twitter by regions

6.4. CHALLENGES OF GENERAL ADVERSARIAL NETWORK

Taken in quote from Jonathan Hui's Medium article "Why it is so hard to train Generative Adversarial Network!"

- Non-convergence: the model parameters oscillate, destabilize and never converge,
- Mode collapse: the generator collapses which produces limited varieties of samples,
- Diminished gradient: the discriminator gets too successful that the generator gradient vanishes and learns nothing,
- Unbalance between the generator and discriminator causing overfitting, and
- Highly sensitive to the hyperparameter selections.