Domain Adaptation of Cloud NLP Services through Word Substitutions

V. Sreeramdass [1] **V. Piratla** [1] **S. Sarawagi** [1] **S. Chakrabarti** [1]

Abstract

Several cloud services are available which perform natural language tasks like sentiment classification, named entity recognition, dependency parsing, fine type tagging, etc. While these services are trained on a large diverse dataset encompassing a large number of domains, the performance of the service on a specific narrow domain relevant to the application may be sub par. For the tasks of sentiment classification and NER, this work attempts to adapt sentences in the target domain to the domain of the service model, to improve the performance of the service model, through word substitutions.

1. Introduction

Cloud Services are quite useful for quick integration of NLP tasks into an application. While the performance of the services are commendable in general, the same in a narrow target domain is questionable. This is because models that cloud services use aim for broad applicability at the cost of performance in any particular domain.

While cloud services are certainly useful for a particular application in a particular domain, the services by themselves cannot provide reliable performance for deployment in the real world.

It was observed that certain NER services did not perform adequately when subjected to English and Hindi code mixed sentences. This is reasonable as many transliterated words are outside the vocabulary of the service models. Even if the words did not belong to the entity spans, due to the lack of these context signals, entities in the sentence were not detected. For instance, if a cloud service like Google[1] is subjected to the sentence "John is walking on Ambedkar Marg", the service fails to detect "Ambedkar Marg" as a location, while in the case of "John is walking on Ambedkar Street" it succeeds in doing so.

[1]Computer Science and Engineering, Indian Institute of Technology Bombay. Correspondence to: V. Sreeramdass <vsreeramdass@gmail.com>.

[1]https://cloud.google.com/natural-language/

Another example is sentiment classification. With a model that is trained on the movie reviews, the phrase, "I walked out" is labeled negative, but the phrase "I returned it" is labeled positive, if not negative with low confidence.

This establishes the need to strategically alter the input to a service to obtain correct labels.

It is certainly possible to use sequence-to-sequence models to perform these required modifications. Because we wish to treat the labels produced for the modified sentences as those for the original sentence while also preserving the semantics of the sentence, the sequence-to-sequence models would have to be subjected to a number of constraints. In case of NER, the rephrasal system would have to learn to preserve the entities in the sentence, learn a mapping of entities in the original sentence and the target sentence. In case of sentiment classification, the rephrasal system would have to preserve the information in the sentence.

It is important to first establish that simple word substitutions are sufficient for the two tasks at hand. This being one of the preliminary works in the area, we attempt to provide a strong working baseline that can perform reliably.

2. Related Work

Thread of work on style transfer [2] of text without parallel aligned text pairs is relevant to this work. (Shen et al., 2017) (Yang et al., 2018) learn a style transformer by encoding a sentence to a style agnostic content representation and decoding conditional on the content and the target style. This simple auto-encoder like transformer is shown to perform well on tasks such as translating between two related languages, transforming reviews from positive to negative sentiment, deciphering word substitution dictionary. (Prabhumoye et al., 2018; Fu et al., 2018) are also relevant to our work.

(Lample et al., 2019) extends the earlier work on style transfer of single attribute such as sentiment to transferring multiple attributes. They propose to use a simple de-noising auto-encoder with back-translation loss and argue that the discriminator adversarial loss is ineffective and unnecessary.

[2]https://github.com/fuzhenxin/Style-Transfer-in-Text

(Li et al., 2018) exploits the fact that the edits required to transform sentence are sparse, they propose to identify and remove phrases that are highly correlated to the source label and replace them with its appropriate target substitution. They propose four different variants for learning to replace including retrieval based, template matching and RNN based decoder.

(Gong et al., 2019) identifies one of the drawbacks in the existing work on style supervision, which is that they do not explicitly attempt to retain the content of the text being transformed. They propose three components that reward each of (1) success of style transfer (2) fluency (3) word-embedding distance between the original sentence and its transformation. The input sentence is encoded using a GRU and a generator GRU with attention on the source's hidden states is trained using REINFORCE. Since rewards are only generated at sentence level, in order to obtain reward for $P_G(y_t/s_t)$, the sentence is unrolled to the full length using previously sampled word at every time step and reward for the unrolled sentences are then used to update generator parameters.

3. Problem Statement

3.1. Sequence Tagging

Given a sentence $\bar{x_i}$ with tokens $x_{i1}, x_{i2}, x_{i3}, ..., x_{in}$ and sequence tags $y_{i1}, y_{i2}, y_{i3}, ..., y_{in}$ where x_{ij} belongs to the target vocabulary V^T and y_{ij} belongs to the tag space $\mathcal{Y}$, our objective is to learn a transformation $T : X \rightarrow \hat{X}$ where X is the target domain and $\hat{X}$ is the source domain, so that the tags $\hat{y_i}$ predicted by the service model M_S on the input sequence $T(\bar{x_i})$ match the ground truth tags y_i.

3.2. Sequence Classification

Similar to the above case, for a given sentence $\bar{x_i}$ and label y_i in label space $\mathcal{Y}$, we wish we wish to learn a transformation T such that the label $\hat{y_i}$ predicted by the service model M_S on the input sequence $T(\bar{x_i})$ matches the ground truth label y_i.

3.3. Word Substitution Model

The the transformation function $T : X \rightarrow \hat{X}$ is limited to take the form

$$T(\bar{x}) = (T_s(1, \bar{x}), T_s(2, \bar{x}), T_s(3, \bar{x}), ..., T_s(n, \bar{x}))$$

where $T_s : X \rightarrow Vs$, Vs being the source vocabulary.

4. Methods

4.1. Reinforcement Learning

A network is designed to model the probability distribution over the source vocabulary at a position, given the input sentence. By design, $T_s = argmaxP(x|\bar{x})$ for the ith position.

REINFORCE (Sutton et al., 2000) is used to train this network based on the rewards at each position. In case of classification, the sentence wide reward is used at each position.

5. Aspects of the approach

5.1. Language Models

One way to ensure the sentence retains its meaning in a broad sense, the number of tokens replaced in a sentence is to be minimized. To do this and introduce an initial bias towards a natural replacement, we use language models, specifically ELMO (Peters et al., 2018). We did not use BERT (Devlin et al., 2018) as there were several issues with BERTs word-piece tokenizer. Word piece tokenizer splits entities into several sub-tokens and replacing sub-words amount to deleting or inserting or replacing sub-words. Aligning expected and predicted tags under such transformations that do not retain token-level alignment proved to be cumbersome.

5.2. Rephrasal

The language model ELMO provides the probability distributions $P(x_i|x_{1:i-1})$ and $P(x_i|x_{i+1:n})$. We design the model as

$$P_R(\cdot|x_{1:n}) = softmax(W_e f_{1:i-1} + W_e b_{i+1:n} + W_e e_i)$$

where $f_{i:j}$ and $b_{i:j}$ are the forward and backward LSTM states obtained from reading tokens $x_{i:j}$ accordingly. e_i is the token embedding, obtained by an LSTM from character embeddings of the token x_i. W_e is initialized with ELMO token embeddings over the source vocabulary V_s.

5.3. Masker

We also study a masker network to model whether a particular token is to be replaced by another token or left as is.

$$P_M(replace, i|x_{1:n}) = sigm(W_{m_1} relu(W_{m_2} h))$$

$$h = f_{1:i-1} \cdot b_{i+1:n} \cdot e_i$$

If the probability is greater than 0.5, the token is replaced by a token as per the rephrasal network predicted probability distribution, else left to be the input token.

The probability of an action in case of the use of a masker is

$$P_T(\cdot|x_{1:n}) =$$
$$P_M(replace, \cdot|x_{1:n}) * P_R(\cdot|x_{1:n}) +$$
$$(1 - P_M(replace, \cdot|x_{1:n})) * I(\cdot, i)$$

5.4. Training

During training, a set of N_s sentences are constructed for each x_i in a batch. Samples from this sampling distribution P_S are derived from P_T by doing the following. Inferring the tokens to be replaced from P_M, and only sample replacement tokens at positions where mask is one. With probability $\frac{\epsilon}{|V_S|}$, sample from a uniform distribution and with probability $1 - \epsilon$ sample from P_R.

For these samples, the service provides tags/labels and rewards. The objective function for one particular sentence $\hat{x}$ is as follows:

$$OBJ(\Theta) = -RLT + \lambda_M RGT$$
$$RLT = \Sigma r_j * log(P_T(\hat{x_j}|x_{1:n}))$$
$$RGT = \Sigma_j |P_M(replace, j|x_{1:n})|$$

RLT is the REINFORCE term, where as RGT is a regularization term. λ_M is a hyper parameter to be tuned. Its value is representative of the number of tokens that are allowed to be replaced in a sentence.

5.5. Service Model

The service model is a black box which only provides tags/labels for the input sentence. However, because we want to analyze the behaviour of the service model, understand how the rephrasal model works and what class of words are replaced with which others, we train our own service models.

For the NER task, our service model is a BiLSTM-CRF model built on top of ELMO embeddings. We also use an BiLSTM-CRF model built with glove embeddings. For the Sentiment Classification task, our service model is a classifier built on top of BERT. The models are trained/fine-tuned on the source domain datasets.

5.6. Reward Structure

For named entity recognition, the reward is positive if the predicted tags match the ground truth tags and negative otherwise. They are +10 if the matched tags are entity tags and +1 for "O" tags. If an entity word's tag is predicted "O", it receives a -10 reward and -1 otherwise.

For sentiment classification, the reward is +1 if the predicted label matches the ground truth label, and -1 otherwise. Each position is associated with the same reward.

To decrease variance in the policy function updates, the reward r_j considered is the reward obtained less the baseline. In a standard MDP, baseline is the value of the value function at the particular state. But since no such estimate is available, the baseline used is the reward obtained on the original sentence $x_{1:n}$.

6. Datasets

6.1. Named Entity Recognition

The CoNLL dataset (Sang & De Meulder, 2003) comprises of news articles from the Reuters Corpus with entities annotated to be one of the four, person, location, organization and miscellaneous.

We use a fabricated dataset CoNLL-Spanish derived from CoNLL. For constructing this dataset, we first consult a translation dictionary [3] mapping common words from English to Spanish. One-fifth the tokens in a sentence are replaced by their Spanish counterparts randomly, with those words more likely to be replaced that occur frequently around entities. The tags are retained.

Twitter NER Dataset (Ritter et al., 2011) comprises of tweets tagged with entity labels. Sentences in this dataset are unlike the sentences in the CoNLL in terms of style, formality and adherence to syntax.

WNUT NER Dataset [4] comprises of sentences harvested from various noisy internet sources. The dataset is particularly challenging to learn from and suits our task of domain adaptation.

In all datasets, the tags are appropriately mapped to tags appearing in CoNLL.

6.2. Sentiment Classification

Multi-Domain Sentiment Dataset (Blitzer et al., 2007) is a popular dataset of reviews from amazon in four domains: books, dvd, electronics and kitchen. One of the domains is held out as target domain while a union of the rest is used to train the service model.

Amazon Product Reviews (He & McAuley, 2016) is a bigger dataset with reviews spanning a lot more domains. A few of the domains are used.

IMDB Sentiment Dataset (Maas et al., 2011) comprises of movie reviews and is useful for sentiment classification.

In all the datasets, reviews with a rating above three are assigned a positive sentiment label, while those below 3 are assigned a negative one. The ones with a rating three are discarded. The datasets are balanced so that the num-

[3] http://www.june29.com/idp/

[4] https://noisy-text.github.io/2017/emerging-rare-entities.html

ber of reviews with positive and negative labels are equal. Train/test splits if not already provided are in the ratio 4:1.

7. Experiments

7.1. NER

One of the service models used for NER is a BiLSTM+CRF model using glove token embeddings. It is tested in two settings with restrictions on the input vocabulary, one with a large vocab (LV) of 768k tokens from ELMO and a small vocab (SV) 10k most frequent tokens in CoNLL. No masking is used in the rephrasal model. The probability distribution at each position is obtained by multiplication of probability distributions obtained from the left, right contexts and token embeddings.

Test	LV	SV
CoNLL	84.9	61.7
CoNLL-Spa	76.6	53.1
Twitter	31.9	31.4
WNUT	25.6	19.12

Table 1. Performance of the BiLSTM+CRF service model trained on CoNLL data in terms of token level F1 scores for NER tags.

Table 2 shows performance on the respective test set when trained with the replacement vocabulary V_S restricted to SV. While the rephrasal system certainly succeeded in replacing unknown tokens with tokens in-vocabulary, the service model is an under-performer. In almost all the cases, it was observed that the performance did not increase beyond the reported numbers upon further training.

The increase of F1 score on CoNLL-Spanish from 53.1 to 59.6 in the case of an untrained rephrasal model was observed to be because the chosen language model was a character level model. Out of vocabulary spanish words were replaced with english words with similar spellings, which in many cases turned out to be the correct in-vocabulary replacements for the same.

Dataset	F1	E
CoNLL-Spa	53.1→59.6	0
CoNLL-Spa	53.1→73.2	1
Twitter	31.4→33.05	3
WNUT	19.12→27.2	3

Table 2. Performance of the BiLSTM+CRF service model trained on CoNLL data in terms of token level F1 scores for NER tags when trained and tested on the respective set with and without word substitutions. E denotes the number of epochs the rephrasal system was trained for.

The other service model considered is a BiLSTM model that uses ELMO embeddings whose performance on the source domain is closer to state of the art. V_S is the union of SV and LV. While the service model has an F1-score of 36.28 on WNUT, even after 3 epochs of training, the score only improved to 37.01 points. It was observed that the words were replaced by themselves and not those that were more relevant to the source domain.

7.2. Sentiment Classification

Source \ Target	IMDB	a-clothes
IMDB	72.81	71.64

Source \ Target	md-bde	md-kitchen
md-bde	73.14	73.98

Table 3. Service model, a classifier built on top of BERT, trained and tested on the indicated source and target domains. The performance metric is F1-score. V_S is the union of SV and LV.
md-bde is the union of reviews from the domains books, dvds and electronics; a-clothes is a dataset of Amazon reviews on Apparel/Clothing

When source domain is IMDB and the target is amazon-clothes, the service has a performance score of 84.47. A service trained on IMDB is capable of performing as high as 89.88 on IMDB. Given that the margin of improvement is substantial, it is theoretically possible to improve to 89.88 at least.

E \ Mask	✓	✗
0	71.63	71.83
2	71.62	70.89
4	71.62	70.41
6	71.64	-
8	71.64	-

Table 4. Performance of the model with and without a masker as training proceeds. E denotes the epoch number

In the case of absence of a masker, the performance of the model deteriorates as training proceeds. This is presumably because the negative reward is not attributed to the appropriate token. This could be leading to reduction in confidence of all the tokens and the unlearning of the language model.

In the presence of masker however, all the tokens that were marked for replacement were replaced by the same token. It could be the case that the bias of the language model was not eliminated in favour of favourable replacements. There was no observable pattern in the tokens selected for replacement either.

8. Future Work and Conclusion

It is clear that there are a number of issues with the current approaches.

The choice of replacement vocabulary is non trivial. While V_S should be sufficiently large so as to not restrict the information contained in the input sentence, it should be small enough for the algorithm to be able to explore the space systematically.

Random exploration in the vocabulary space is fruitless as the space is quite large. In the presence of unlabeled data, the rephrasal models could be pretrained on the source domain. This would provide a better initial bias to the exploration that is relevant to the source domain. For the task of sentiment classification, exploratory policies guided by sentiment sensitive embeddings could be explored.

There is a dire need to investigate how to assign sentence level rewards to individual tokens. For this cause, systematic sampling techniques could be employed, where sample sentences with single tokens / bigrams replaced are generated. A systematic way to do this could be to use CRFs to model bigram distributions on the source domain and explore based on context signals, if not decoder LSTMs and use Monte Carlo sampling to guide explorations.

While this particular application does not suffer from temporal reward assignment problem, it does in fact suffer in the battle between exploration and exploitation. There is a need to actively remember what explorations lead to good rewards and introduce noise in the training of the network. This also addresses the concern regarding the network not realizing a reward sufficiently quick.

The task is, strictly speaking, not an Reinforcement Learning problem, but a Multi-Arm Bandit problem. Use of MAB methods like LUCB, Thomson Sampling should provide not only good baselines, but justifications to modeling replacement distributions as a function of context.

Going beyond reinforcement learning, there are other methods to be explored like max-margin training, imitation learning among others which perform a "hard" assignment of rewards.

References

Blitzer, J., Dredze, M., and Pereira, F. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th annual meeting of the association of computational linguistics*, pp. 440–447, 2007.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Fu, Z., Tan, X., Peng, N., Zhao, D., and Yan, R. Style transfer in text: Exploration and evaluation. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Gong, H., Bhat, S., Wu, L., Xiong, J., and Hwu, W.-m. Reinforcement learning based text style transfer without parallel training corpus. *arXiv preprint arXiv:1903.10671*, 2019.

He, R. and McAuley, J. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*, pp. 507–517. International World Wide Web Conferences Steering Committee, 2016.

Lample, G., Subramanian, S., Smith, E., Denoyer, L., Ranzato, M., and Boureau, Y.-L. Multiple-attribute text rewriting. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=H1g2NhC5KQ.

Li, J., Jia, R., He, H., and Liang, P. Delete, retrieve, generate: A simple approach to sentiment and style transfer. *arXiv preprint arXiv:1804.06437*, 2018.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*, pp. 142–150. Association for Computational Linguistics, 2011.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

Prabhumoye, S., Tsvetkov, Y., Salakhutdinov, R., and Black, A. W. Style transfer through back-translation. *arXiv preprint arXiv:1804.09000*, 2018.

Ritter, A., Clark, S., Etzioni, O., et al. Named entity recognition in tweets: an experimental study. In *Proceedings of the conference on empirical methods in natural language processing*, pp. 1524–1534. Association for Computational Linguistics, 2011.

Sang, E. F. and De Meulder, F. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*, 2003.

Shen, T., Lei, T., Barzilay, R., and Jaakkola, T. Style transfer from non-parallel text by cross-alignment. In *Advances in neural information processing systems*, pp. 6830–6841, 2017.

Sutton, R. S., McAllester, D. A., Singh, S. P., and Mansour, Y. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pp. 1057–1063, 2000.

Yang, Z., Hu, Z., Dyer, C., Xing, E. P., and Berg-Kirkpatrick, T. Unsupervised text style transfer using language models as discriminators. In *Advances in Neural Information Processing Systems*, pp. 7287–7298, 2018.