Unsupervised Machine Translation: The Evolution

Literature Survey

CS 680: Introduction to Machine Learning

Professor: Pascal Poupart

Yixing Sun (y494sun)

David R. Cheriton School of Computer Science

University of Waterloo

- ## Introduction

Motivation

When it comes to the natural languages that human speak, machines always have a hard time to handle them properly, especially for language translation tasks. As it turns out, the raw form of plain text is not suitable for various machine learning algorithms so that the word embedding techniques come into play.

Conceptually, word embedding tries to map words using dictionary to vectors in a way that words are quantified and analyzed based on their contexts. As a result, the string input is not only transformed into its numerical form for easy machine processing, but it also has its semantical meaning preserved within the scope of the training datasets.

The fact is that although different languages have different grammatical structures and alphabets, they share common internal relationships. For instance, the logic between words ‘drive’ and ‘car’ is independent of the languages used. With these similarities, we may expect the relative distances between word embeddings are shared amongst all languages.

Supervised vs Unsupervised?

The machine translation process can be interpreted as finding the transformation matrix that maps the source word embeddings to target word embeddings with minimum loss. In fact, most of the state-of-the-art supervised models are driven by this methodology. However, they are all heavily dependent to the availability, accuracy and scalability of the bilingual dictionaries or parallel corpora.

As matter of fact, these parallel corpora are sometimes very costly to build due to the lack of supervision from experts. In addition, maintaining such existing dictionaries adds more difficulties because of the dynamics in vocabularies over time (There are always new words and phrases created every year, conversely, obsolete or less frequent used expressions may be removed).

On the contrary, monolingual data is much easier to maintain and scale, e.g. for k languages, we will only need k monolingual data instead of k(k-1)/2 bilingual pairs. This project will focus on the unsupervised machine translation models that leverage the monolingual data, and their evolution from word translation to sentence translation.

- **Survey**

The Evolution

Mikolov et al. (2013) first tried to exploit similarities among monolingual embeddings of different languages [1]. The motivation is that the current bilingual dictionaries which are heavily relied on by the supervised machine translation systems are far from perfect, and they are outperformed by human experts most of the time. To mitigate the missing dictionary entries and automate dictionary generation, they proposed using the distributed Skip-gram or Continuous Bag-of-Words (CBOW) for the language representation and use a subset of a bilingual dictionary to learn a linear projection between the latent domains corresponding to each language domain.

Xing et al. (2015) improved this model by arguing the inconsistency among objective functions for embedding learning (MLE on inner products), distance measurement (Cosine Similarity) and linear transformation learning (MSE). [2] They attempted to normalize the word vectors by enforcing them being unit length during training. Furthermore, they proposed the orthogonal constraint on the linear mapping matrix so that the transformed embedding (Wx_{src}) is also normalized, which is reduced to the Procrustes problem where closed form solution can be found from Singular Value Decomposition (SVD).

Recently, several attempts have been made to alleviate the strong reliance on cross-lingual annotated data, but they turned out to be not on par with the state-of-the-art supervised approaches. Conneau et al. (2017) first reversed this situation. They developed a word translation model that is solely dependent on monolingual embeddings without any supervision from cross-lingual dictionary. [3] They leveraged Generative Adversarial Network of Goodfellow et al. (2014) [4] to seek the optimal linear transformation mentioned above. Specifically, they construct a discriminator to distinguish between the transformed source embeddings and the target embeddings, while the linear transformation matrix, or the generator, is trained to fool the discriminator so that it would not be able to make the right decision. To fine tune the shared embedding domain, they introduced so called Cross-Domain Similarity Local Scaling (CSLS). This approach becomes essential to mitigate the hubness problem, as a result, it significantly boosts the absolute performance.

Lample et al. (2018) brought this concept to a higher level. They achieved impressive performance on unsupervised sentence translation. [5] The key component is the sequence-to-sequence model with attention of Bahdanau et al. (2015). [6]

The main goal is to train the encoder/decoder for both source language and target language so that they would eventually share a common latent space, and more importantly, be able to reconstruct either source or target domain from the latent space for bi-directional translation.

- **Analysis**

Part I: Word Translation

Linear Mapping W

Let us first have a close look at Mikolov et al. (2013) approach. [1] Assume we have a bilingual dictionary which contains n pairs of words:

$$\{x_i, y_i\}_{i\in\{1,2,\dots,n\}},$$

$$where\ x_i \in \mathbb{R}^d, y_i \in \mathbb{R}^d$$

$$are\ the\ distributed\ representation\ of\ word\ i\ in\ source\ and\ target\ languages.$$

The objective function:

$$Loss = \min_W \sum_{i=1}^{n} \|Wx_i - y_i\|^2$$

$$\mathrm{W}^* = argmin_W \|WX - Y\|_F$$

As the objective function indicates, we are interested in finding a linear projection $\mathrm{W} \in \mathbb{R}^{d\times d}$ that maps source embeddings to target embedding with minimal loss. Here, the mean square loss function is used, which can be solved by gradient descent.

After we have the optimal W*, we can compute the translation y for any source word x, s.t

$$y^* = argmax_y \cos(W^*x, y)$$

y* is the nearest target word embedding from the transformed source word embedding W*x measured by cosine similarity. (Note: Cosine similarity ranges from 0 (90 degree) to 1 (0 degree). By maximizing the cosine similarity, we effectively find two vectors pointing to the same direction)

Normalized Word Vectors

In the skip-gram model, we intend to make a prediction of the word being near the central word in the defined window. The probability is given by the SoftMax:

$$Pr(w_{i+j}|w_i) = \frac{e^{c^T{}_{w_{i+j}} c_{w_i}}}{\sum_w e^{c^T{}_w c_{w_i}}}, where\ c_w\ denotes\ word\ vector\ of\ w$$

As we can see, the similarity measurement in the training process is given by the inner product $c^T{}_w c_{w'}$. However, when we use those word vectors for translation, we choose cosine similarity:

$\frac{c^T{}_w c_{w\prime}}{|c_w||c_{w\prime}|}$. To mitigate the inconsistency in measurement, Xing et al. (2015) suggested to normalized word vectors so that inner product becomes equivalent to cosine similarity. [2]

Orthogonal Transformation

$$Loss = \min_W \sum_{i=1}^{n} \|Wx_i - y_i\|^2 = \max_w \sum_{i=1}^{n} (Wx_i)^T y_i$$

If we re-examine the above loss function and its equivalence, it is worth noting that, although x_i and y_i are normalized, Wx_i is not guaranteed to be normal. Assuming that source and target word vectors are same in vector space dimensions. The normalization constraint can be satisfied by enforcing orthogonality constraint on W. More importantly, we end up with a closed form solution: $W = UV^T$ for the loss function, and it is well known as the Procrustes problem.

Proof:

$$Rewrite\ the\ above\ loss\ function\ to\ its\ matrix\ form:$$
$$Loss = \min_W \|XW - Y\|_F{}^2$$

$Since\ \|A\|_F{}^2 = tr(AA^T) = tr(A^T A), we\ have,$

$$Loss = \min_W \left(\|XW\|_F{}^2 + \|Y\|_F{}^2 - 2tr(W^T X^T Y) \right)$$

$$Loss = \min_W \left(\|X\|_F{}^2 + \|Y\|_F{}^2 - 2tr(W^T X^T Y) \right) = \max_W (tr(W^T X^T Y))$$

$Apply\ Singular\ Value\ Decomposition, X^T Y = UDV^T$

$$\max_W (tr(W^T X^T Y)) = \max_W (tr(W^T UDV^T))$$

$Since\ trace\ is\ invariant\ under\ cyclic\ reordering,$

$$\max_W (tr(W^T UDV^T)) = \max_W (tr(V^T W^T UD))$$

$Let\ Z = V^T W^T U, since\ V, W\ and\ U\ are\ all\ orthogonal\ matices, Z\ is\ also\ orthognal.$

$$\max_W (tr(V^T W^T UD)) = \max_W (tr(ZD))$$

$Since\ D_{i,i}\ is\ a\ nonnegative\ diagonal\ matrix\ and\ decreasing\ in\ i, the\ maximum\ is\ obtained\ by\ optimize\ w\ so\ that\ Z_{i,i} = 1, in\ other\ word, Z = I.$

$$Consequently, W^T = VU^T, W = UV^T (nice\ closed\ form\ solution).$$

Generative Adversarial Network

Up until now, all the analysis above that seeks to find the linear mapping W is supervised by a complete or subset of cross-lingual dictionary. Conneau et al. (2017) proposed the unsupervised approach which is based on the concept of Generative Adversarial Networks (GANs) of Goodfellow et al. (2014). [3][4]

GANs inexplicitly model the high-dimensional distribution of training data, through training a pair of networks, a generator and a discriminator, against each other. The goal of the generator is to learn the linear mapping that maps the embeddings from source domain to target domain. The job of a discriminator is to determine whether the input embedding comes from the source embeddings or the target embeddings.

Throughout the training process, the discriminator is evolving to be better at classifying the source and the target embeddings, while the generator is constantly trying to fool the discriminator by generating source embeddings closer and closer to the target ones.

The objective function defined by Conneau et al. (2017) is shown below, [3]

$$\mathcal{L}_D(\theta_D|W) = -\frac{1}{n}\sum_{i=1}^{n}\log P_{\theta_D}(source = 1|Wx_i) - \frac{1}{m}\sum_{i=1}^{n}\log P_{\theta_D}(source = 0|y_i)$$

$$\mathcal{L}_W(W|\theta_D) = -\frac{1}{n}\sum_{i=1}^{n}\log P_{\theta_D}(source = 0|Wx_i) - \frac{1}{m}\sum_{i=1}^{n}\log P_{\theta_D}(source = 1|y_i)$$

The first equation demonstrates the loss function for the discriminator, which can be seen as maximizing the log likelihood of successfully classifying Wx_i as source embedding and y_i as target embedding for all training samples i. On the contrary, the generator W tries to mislead the trained discriminator θ_D so that it would wrongly classify embeddings from either language domain.

Cross-Domain Similarity Local Scaling (CSLS)

The parallel vocabulary is generated by the linear mapping learned from the adversarial training. However, this alignment is not perfect. Author observed that rare words tend to have similar embeddings among different languages, so they should be less updated. In addition, compared to frequent words, embeddings for rare words are relatively less expressive, and their qualities are not on par with their frequent counterparts.

In order to improve the performance of the Procrustes optimization, Conneau et al. (2017) attempted to create reliable pairs by considering that, the k-nearest neighbours of the source words from the target domain are very likely to have this particular source word as one of their

k-nearest neighbours. [3] However, it tends to be less promising when it comes to the higher dimensional space. Particularly, some word vectors have higher probability of being the nearest neighbours of others, while some other vectors would have no near neighbour at all. This phenomenon is the result of the asymmetric property, and it is denoted as hubness problem. The solution they proposed is to construct a bi-partite graph where each source embedding is connected by its k-nearest neighbours in the target domain, and vice versa.

The mean similarity of source embedding to its neighbouring target embeddings is given below:

$$r_T(Wx) = \frac{1}{k} \sum_{y \in N_T(Wx)} \cos(Wx, y)$$

By the same token,

$$r_S(y) = \frac{1}{k} \sum_{Wx \in N_T(y)} \cos(y, Wx)$$

The CSLS is defined as,

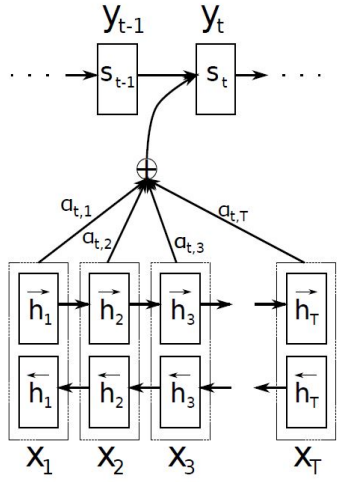
$$CSLS(Wx, y) = 2\cos(Wx, y) - r_T(Wx) - r_S(y)$$

This new similarity measure seeks language pairs (x, y) that x and y shares high similarities, at the same time, they are noticeably different from the k-nearest neighbours on the other domain. This new metric effectively mitigates the hubness problem and revise the similarity measures associated with isolated embeddings.

Part II: Sentense Translation

Sequence-to-Sequence Model with Attention

The main architecture Lample et al. (2018) used for the unsupervised sentence translation is based on sequence-to-sequence model with attention mechanism of Bahdanau et al. (2015). [5][6]



As we can see from the left figure, the encoder is a bi-directional RNN so it could not only summarize the preceding words but also the following words. As a result, each hidden state h_i is concatenated by a forward hidden state $\vec{h}_i$, and a backward hidden state $\overleftarrow{h}_i$.

The state-of-the-art is that, instead of feeding the decoder a single compressed context vector that encodes everything in a sentence or paragraph, Bahdanau et al. (2015) proposed to

compute an adaptive weighted sum of all the intermediate hidden states from encoder as the context vector at each step of decoding, [6]

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j \, , and \; \alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})} , where \; e_{ij} = a(s_{i-1}, h_j)$$

The alignment model for each decoder hidden state s_i given one of the encoder hidden state h_j is learned by a feedforward neural network that takes account of the previous hidden state of the decoder. The weight α_{ij} is a SoftMax of given score for h_j over the sum of scores over all hidden states of the encoder. The benefit of this model is that it learns the local mapping between different domains and mitigate the long-term memory loss.

Objective Functions

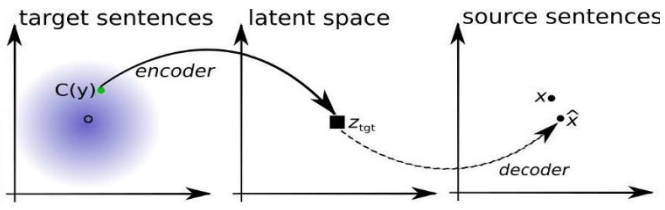
Back to the model Lample et al. (2018) proposed. There are three objective functions required to be optimized. [5]

1) **Denoising Auto-Encoding**: The motivation is that we would like to reconstruct the language domain from the latent space. The caveat is that we cannot simply train the autoencoder by minimizing the loss between original sequence and the output sequence. In that way, the autoencoder would trivially copy each word in the input sequence, and no pattern will be learned. The solution is applying a noise model to the input so that the autoencoder has to reconstruct a corrupted version of the sequence.

$$\mathcal{L}_{auto}(\theta_{enc}, \theta_{dec}, Z, l) = E_{x \sim D_l, \hat{x} \sim d(e(C(x),l),l)}(\triangle(\hat{x}, x))$$

This objective function can be interpreted as follows: Given a sequence x sampled from the language domain D_l, we will first intentionally apply noise model to obtain a corrupted version $C(x)$. After going through encoder e and decoder d, we end up with the output sequence $\hat{x}$. The goal is to learn parameters of this autoencoder (θ_{enc}, θ_{dec}) and the embedding Z, so we could minimize the discrepancy between $\hat{x}$ and x.

2) **Cross Domain Training**: The purpose of the second objective is improving the model capability to map one domain to another, which is the ultimate goal for translation. The key the concept of back translation.



$$\mathcal{L}_{cd}(\theta_{enc}, \theta_{dec}, Z, l_1, l_2) = E_{x \sim D_{l_1}, \hat{x} \sim d(e(C(M(x)), l_2), l_1)}(\triangle(\hat{x}, x))$$

M is the current model of translation. Initially, M is the word-to-word mapping learned from the adversarial training of Conneau et al. (2017). [3] We can then obtain a rough translation $M(x)$.

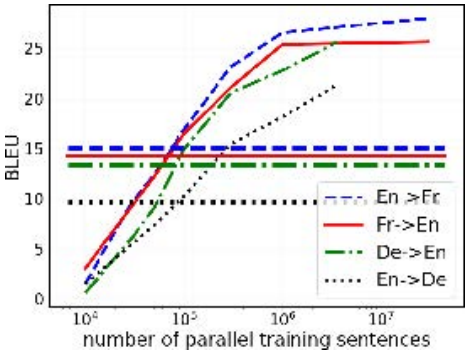
The same trick applied here, the corrupted version of $M(x)$ denoted as $C(M(x))$ gets feed into the target domain encoder, and the latent Z_{tgt} is produced. Lastly, Z_{tgt} get decoded by the source decoder, and the back translated input $\hat{x}$ can be compared with the original.

3) **Adversarial Training**: To ensure decoder cooperates with encoder under different domains, the author again adopted the Generative Adversarial Networks of Goodfellow et al. (2014). The main idea is to train the encoder so that the latent spaces share the same distribution regardless of the input language domain.

$$\mathcal{L}_{adv}(\theta_{enc}, Z|\theta_D) = -E_{(x_i, l_i)}(\log p_d(l_j|e(x_i, l_i)))$$

l_i and l_j represents two different language domains. The encoder is trained to fool the discriminator by classifying the output from one language to another. Eventually, the latent space would not imply any input language domain.

Part III: Open Problem



When comparing the unsupervised model to existing supervised models, we can observe that, under 10^5 sample sizes, unsupervised model outperforms the supervised counterpart. However, as the number of the training samples increases, supervised approaches are still the state-of-the-art. [5]

- **Conclusion**

This literature review provides me an opportunity to have a thorough walkthrough along the evolution of the unsupervised machine translation. Various techniques are closely examined from the basic principles.

The future research may put emphasis on semi-supervised approach which is capable of leveraging a small subset of cross-lingual dictionary while boosting the overall performance much further.

- **Reference**

[1] Tomas Mikolov, Quoc V Le, and Ilya Sutskever. Exploiting similarities among languages for machine translation. arXiv preprint arXiv:1309.4168, 2013.

[2] Chao Xing, Dong Wang, Chao Liu, and Yiye Lin. Normalized word embedding and orthogonal transform for bilingual word translation. Proceedings of NAACL, 2015.

[3] A. Conneau, G. Lample, M. Ranzato, L. Denoyer, and H. Jegou. Word translation without parallel data. arXiv:1710.04087, 2017

[4] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. Advances in neural information processing systems, pp. 2672–2680, 2014.

[5] Guillaume Lample, Alexis Conneau, Ludovic Denoyer, Marc'Aurelio Ranzato. Unsupervised machine translation using monolingual corpora only. arXiv:1711.00043v2, 2018

[6] Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. arXiv:1409.0473v7, 2015